MDPI

Article

Leveraging Multi-agent Reinforcement Learning for Digital Transformation in Supply Chain Inventory Optimization

Bo Zhang [1]*, Wen Jun Tan [1], Wentong Cai [1], and Allan N Zhang [2,3]

[1] College of Computing and Data Science, Nanyang Technological University, 50 Nanyang Ave, Singapore 639798, Republic of Singapore
[2] Singapore Institute of Manufacturing Technology (SIMTech) Agency for Science, Technology and Research (A*STAR), 5 CleantTech Loop, Singapore 636732, Republic of Singapore
[3] School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore 639798, Republic of Singapore
* Correspondence: bo003@ntu.edu.sg

Abstract: In today's volatile supply chain (SC) environment, competition has shifted beyond individual companies to the entire SC ecosystem. Reducing overall SC costs is crucial for success and benefits all participants. One effective approach to achieve this is through digital transformation, enhancing SC coordination via information sharing, and establishing decision policies among entities. However, the risk of unauthorized leakage of sensitive information poses a significant challenge. We propose a Privacy-preserving Multi-agent Reinforcement Learning (PMaRL) approach to enhance SC visibility, coordination, and performance during inventory management while effectively mitigating the risk of information leakage by leveraging machine learning techniques. The SC inventory policies are optimized using multi-agent reinforcement learning and SC connectivity information. Moreover, the simulation-based evaluation illustrates that the PMaRL method surpasses traditional optimization approaches in achieving cost performance comparable to full visibility methods, all while preserving privacy. This research addresses the dual objectives of information security and cost reduction in SC inventory management, aligning with the broader trend of digital transformation.

Keywords: Digital Transformation; Data-driven Decision Making; Supply Chain Inventory Management; Multi-agent Reinforcement Learning; Privacy Preserving

Citation: Zhang, B.; Tan, W.J.; Cai, W.; Zhang A. Leveraging Multi-agent Reinforcement Learning for Digital Transformation in Supply Chain Inventory Optimization. *Sustainability* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. INTRODUCTION

In recent years, there have been increasing numbers of research regarding supply chain (SC) coordination and information sharing that aimed at improving overall SC operation through digital process [1–3]. Digital transformation empowers SC to utilize data-driven decision-making in order to increase coordination and visibility, which have proven beneficial in enhancing operational profitability, service quality, and mitigating demand uncertainties for individual firms as well as the SC ecosystem [3–5]. Especially during the COVID-19 pandemic, SC uncertainties were also rising, making the SC even more fragile [6]. Furthermore, with the fierce and various global competition and the increasing SC networks complexity, coordination and visibility play an increasingly crucial role in reducing the overall cost and uncertainties to all entities in the SC [7–9]. However, achieving effective coordination and information sharing is challenging. Firms usually seek to improve their performance by centralized coordination and decision-making within complex SC networks [1].

Data-driven coordination in SCs can be categorized into two main types: full information sharing and partial information sharing [10]. Full information sharing involves gathering data from various firms and centralizing it in a trusted party. A decision model is then developed on a central server using historical data to make decisions for all firms using the real-time data [10]. Since decisions are made for all firms using real-time data, it requires

coordination of information at every step. On the other hand, in partial information sharing, for example, each SC firm receives the demand forecast of its downstream firms and aims to meet this demand as fully as possible, utilizing its inventory policy model to generate its own demand prediction. However, the error in demand prediction inevitably accumulates over time. Moreover, although the above-mentioned approaches to collaborating and sharing information offer advantages to firms, they also expose firms to potential risks that the shared information may be leaked. The information leakage may lead to financial loss, reputational setback, and negative brand image [11,12]. Generally speaking, current approaches improve SC visibility, but they come with heightened concerns regarding information security. As a result, protecting the privacy of shared information becomes a major obstacle to SC coordination and impedes the digital transformation of SC optimization.

The development of multi-agent reinforcement learning (MaRL) presents a promising data-driven decision-making solution for improving SC coordination and mitigating the risks of information leakage. MaRL achieves great success in solving various problems. It involves optimizing the behavior of multiple agents within a shared environment [13]. Data can be gathered from the SC insight to train the MaRL system for decision-making. Each agent in MaRL represents a firm in the SC, allowing each firm to make its own decisions. A key feature of MaRL that helps to preserve data privacy is decentralized decision-making [14]. Once the decision model has been trained, each learning agent can make decisions independently. This eliminates the need to share information at every decision step, thereby reducing the risk of information leakage.

In contrast to the existing SC coordination methods, we present a Privacy-preserving Multi-agent Reinforcement Learning (PMaRL) approach which balances privacy concerns and SC performance. In line with the broader trend of digital transformation, PMaRL trains in a semi-privacy manner, requiring firms to share information only during the training. This includes shared embedding data, connectivity details, and training loss among firms. Based on this shared information, each firm can train its own policy and critic networks without disclosing any raw information. The policy networks are used to make decisions based on the current state and the critic networks assess the value of decisions made by the policy networks and guide its parameters update. Once the policy and critic networks have been trained for all firms, each firm can then make decisions independently using its local state.

In addition, PMaRL incorporates topology information into the models to guide agent training to focus on useful shared information. As a result, PMaRL outperforms the MaRL methods that do not utilize network information in terms of training time, convergence speed, and final optimization objective.

The contributions of our work are as follows:

- We propose a PMaRL method that only requires to share limited information during the training stage, overcoming the limitations of existing approaches that require full information visibility across SC.
- The PMaRL approach also incorporates SC network topology to optimize agent training. By using network topology it classifies SC firms into distinct groups, leveraging their relationship to enhance the training process.
- We conduct a simulation-based evaluation that demonstrates the superior performance of the PMaRL method compared to current optimization approaches.

The remaining sections are organized as follows: Section 2 presents the related works on information sharing, inventory optimization approaches, and MaRL. Section 3 describes the problem statement of SC optimization. Section 4 presents the details of the PMaRL method. Section 5 evaluates the performance of PMaRL against the existing methods using SC simulation. Section 6 concludes our paper and discusses limitations and future work.

2. RELATED WORK

2.1. Information Sharing

Multi-stage inventory optimization is a well-studied problem since it is fundamental to study bullwhip effects, operation cost reduction, and firm cooperation in SC [15,16]. In SC optimization, information sharing has been viewed as a significant strategy to counter the bullwhip effect and reduce inventory cost [17]. For instance, downstream firms share their projected demand quantity to minimize the disturbances in the SC. This collaborative approach allows all firms to benefit from increased revenues, more agile demand planning, and reduced SC risk [18]. Highlighting the importance of sharing forecasting information within the SC, studies have demonstrated that such sharing results in lower demand prediction errors [19]. Furthermore, information sharing also helps to smooth out the demand curve [17,20].

However, sharing information among different firms introduces the risk of sensitive information leakage, which is a significant challenge for SCs [21,22]. To address this issue, our model adopts centralized training combined with distributed decision-making. During the training process, only embedding information, connectivity details, and training loss are shared. Decision-making is then distributed based on the well-trained model and the current states of individual firm, reducing the likelihood of information leakage and mitigating potential consequences.

2.2. Inventory Optimization Approach

Existing approaches to SC inventory optimization are varied, including stochastic programming, heuristic search, reinforcement learning (RL), and others. [23,24]. SC inventory optimization involves setting inventory levels to minimize costs and maximize overall efficiency. SC network can be classified into single-chain and multi-stage. A single chain is a linear network; whereas a multi-stage SC consists of multiple tiers or levels of SC entities. SC inventory optimization is confronted by challenges such as uncertain demand, complex SC networks, and complicated interactions among SC entities.

To tackle the single-chain inventory problem, Barlas and Gunduz [17] proposed both optimal and heuristic approaches. For multi-stage SC optimization problems, Rong et al. [25] utilized recursive optimization and decomposition aggregation heuristics. In practice, the multi-stage SC networks are complex [26], and finding optimal solutions for these complex networks is challenging due to the curse of dimensionality [23]. Given the uncertain demand and complex multi-stage network, only the optimal solution for the single chain network can be computed in polynomial time [15,27]. Achieving optimal policy for multi-stage SC using stochastic programming remains difficult [28]. Existing approaches in such cases are NP-hard, with computational time increasing exponentially with model complexity [27].

2.3. Reinforcement Learning

In an increasingly unstable and complex SC environment, managing SC inventory continues to be challenging in many areas, such as decision-making in continuous and discrete processes, the increasing amount of processing data size, and the intricate external SC environment [7]. Advances in computational resources and machine learning algorithms offer new potential for dynamic, data-driven decision-making. Specifically, RL approaches provide end-to-end data-driven solutions for managing SC inventory.

RL primarily focuses on optimizing the actions of either a single agent or multiple agents, typically within simulated environments. In single-agent RL, an agent learns an action policy by exploring and interacting with the environment. In contrast, MaRL involves agents not only interacting with the environment but also developing strategies for competition or cooperation with other agents. This distinct feature of MaRL sets it apart from single-agent RL, as it necessitates that agents account for the dynamics of the multi-agent system and adapt their policies accordingly.

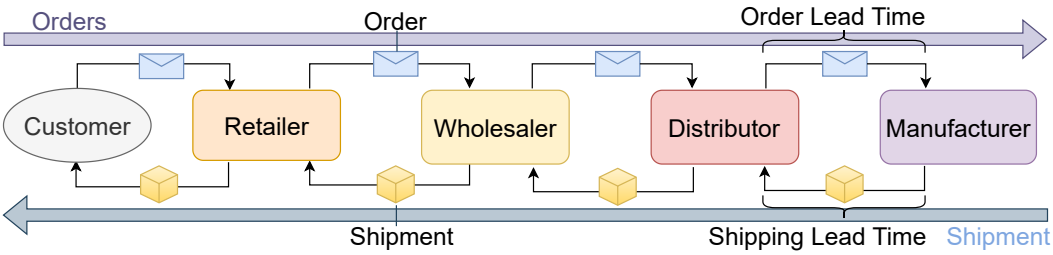


Figure 1. Multi-stage SC.

In single-agent RL, Barat et al. [13] leveraged the actor-critic approach in a SC replenishment scenario, focusing on minimizing wastage and backorder quantities within a firm. Additionally, a deep Q network was introduced to optimize inventory policies, specifically in the context of the beer game [29]. While single-agent RL approaches excel at optimizing the cost for individual entities within the SC, they do so independently. However, when each entity optimizes its operation without cooperation, this can lead to a "prisoner's dilemma" situation [30], potentially degrading overall SC performance. The main drawback of single-agent RL is its limited focus on optimizing the performance of a single entity, often neglecting the broader impact on the entire SC.

In MaRL, studies by Mortazavi et al. [31], Wang and Lin [32], and Nurkasanah [33] applied Q-learning to optimize overall SC performance in inventory management and replenishment strategies. These approaches require raw data, such as inventory level and order quantity, from SC entities to train a shared model, which is then used by each SC entity to make informed decisions. While these approaches help SC entities avoid the "prisoner's dilemma", it necessitates the sharing of original raw data and even decision models amongst entities. The primary goal of MaRL is to find optimized policies for multi-agents within a system, but it does not account for data privacy concerns. Although MaRL demonstrates exceptional performance in SC optimization, the issue of privacy remains a significant concern.

3. PROBLEM DEFINITION

In this section, we describe the problem settings for multi-stage SC optimization. In many instances, SC optimization occurs under the conditions of incomplete information. For example, some firms may not be willing to share their data or business objectives. Additionally, customer demand is often unpredictable and subject to fluctuations. As a result, obtaining complete information during the optimization process is typically impractical.

The objective of SC optimization is to minimize the inventory costs across the entire SC. The multi-stage SC has the following characteristics [30]:

- Cooperation with minor competition
- Incomplete information
- Prisoner's dilemmas

In a multi-stage SC, the optimal strategy for firms is to coordinate with one another to avoid competition. When firms act individually, such as by ordering excessive quantities to reduce short term cost, it ultimately results in higher overall SC costs and increasing long-term costs for individual firms. Moreover, in a multi-stage SC, incomplete information scenarios are common, where firms can only access their own data. The inventory management decisions of each firm also affect others, leading to "prisoner's dilemmas" situations. In these cases, firms must consider the potential actions of others and cooperate accordingly.

Figure 1 shows a special case of a multi-stage SC where each stage has only one entity. The network consists of several entities, namely the *Retailer*, *Wholesaler*, *Distributor*, *Manufacturer*, and *Supplier*. Each of these entities is represented by an agent. In a multi-stage SC, upstream agents (e.g., Wholesaler) receive stochastic orders from downstream agents

(e.g., Retailer). In response, they fulfill orders by sending shipments to the downstream agents.

To simulate real-world scenarios, each agent in the network experiences varying order and shipping lead time. Furthermore, each agent within the SC has its own distinct holding and backorder costs, reflecting the unique cost considerations for each agent.

In the multi-stage SC, C^t is the SC cost at time t:

$$C^t = \sum_{i=1}^{n} c_b^i BO_i^t + c_h^i IL_i^t \tag{1}$$

where n is the total number of agents, c_b^i and c_h^i are the backorder and holding cost coefficients for agent i, respectively. BO_i^t and IL_i^t are the backorder and inventory level for agent i at time t, respectively. Then, the long-run total cost Z is defined as follows:

$$Z = \sum_{t=1}^{T} C^t \tag{2}$$

where T is the total simulation time. The main objective is for each agent to develop an order policy cooperatively in order to minimize the long-run system-wide cost Z.

4. METHOD

In this section, we explore the optimization of the overall inventory cost in a multi-stage SC. To tackle this challenge, we introduce a PMaRL model that incorporates network topology information. The model is trained in a centralized manner to enable effective coordination among agents, but during the decision-making phase it operates in a decentralized manner to ensure privacy. Each agent has its own critic network, which includes encoder, decoder, and the shared attention structure. During training, agents exchange the embedding information and corresponding loss value. While some raw data could potentially be reconstructed through reverse engineering, this approach still offers a viable solution for preserving data privacy.

4.1. Fundamentals

A multi-stage SC can be considered as a special case of Markov Decision Process, defined by a tuple (S, A, P_A, R_A), where S is a set of states; $A = \{A_1, \cdots, A_n\}$, A_i is the action set of agent i; P_A is the probability distribution of all possible next states given state S and action set A: $S \times A_1 \times \cdots \times A_n \rightarrow P_A(S)$; $R_A = \{r_1, \cdots, r_n\}$, r_i is the expected immediate reward of agent i: $S \times A_i \rightarrow r_i$; and n is the total number of agents.

Furthermore, each agent i has its own observation, o_i, representing its observable partial information from the global state, $s \in S$; and its own policy, $\pi_i : o_i \rightarrow P(A_i)$, corresponding to the action probability distribution under agent's observation. Agents optimize policy by maximizing their expected returns,

$$J_i(\pi_i) = E_{a_i \sim \pi_i, \cdots, a_n \sim \pi_n}\big[\sum_{t=0}^{\infty} \gamma^t r_i^t(s^t, a_i^t, \cdots, a_n^t)\big], \tag{3}$$

where $a_i \in A_i$ and $\gamma \in [0, 1]$ is the discount factor that determines the extent to which the policy prioritizes immediate rewards over long-term benefits.

The gradient of agent policy networks can be estimated using the policy gradient technique [34] as follows:

$$\bigtriangledown_\theta J(\pi_\theta) = \bigtriangledown_\theta log(\pi_\theta(a_t|s_t)) \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}(s_{t'}, a_{t'}), \tag{4}$$

where θ represents the learnable parameter. In practice, especially in complex scenarios, it is hard to fully exploit and explore the entire environment to get the precise value of the

term $\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}(s_{t'}, a_{t'})$ as the expected returns in the policy gradient estimator can vary greatly in different training episodes. To address this issue, actor-critic methods [35] are used to augment the original policy gradient approach. It estimates the expected returns by the Q-value function:

$$Q(s_t, a_t) = \mathbb{E}\Big[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}(s_{t'}, a_{t'})\Big]. \tag{5}$$

The Q-value function can be learned by temporal-difference learning technique that minimizes the regression loss of past Q-value and the reward:

$$\mathcal{L}_Q = \mathbb{E}_{s,a,r,s' \sim D}\big[(Q(s,a) - y)^2\big], \tag{6}$$

$$y = r(s,a) + \gamma \mathbb{E}_{a' \sim \pi(s')}\big[Q(s', a')\big], \tag{7}$$

where D is a replay memory storing past experiences.

To encourage the RL agents to explore the environment and avoid premature converging at a local optimal policy, Haarnoja et al. [36] proposes the soft actor critic approach that introduces the entropy term $log(\pi(a|s))$ and incorporates it into the policy gradient:

$$\bigtriangledown_\theta J(\pi_\theta) = \mathbb{E}\big[\bigtriangledown_\theta log(\pi_\theta(a|s))(Q(s,a) - \alpha\, log(\pi_\theta(a|s)) - b(s))\big], \tag{8}$$

where $b(s)$ is the baseline of the Q-value function that measures the value of the current state s, and α is a temperature parameter. Accordingly, the loss function of Q-value is revised by adding the entropy term to Equation 7:

$$y = r(s,a) + \gamma \mathbb{E}_{a' \sim \pi(s')}\big[Q(s', a') - \alpha\, log(\pi(a'|s'))\big]. \tag{9}$$

The actor-critic methods involve updating policy network and critic network using Equations 6 and 8 iteratively. Policy network makes decisions based on the current state and critic network estimates the expected return of the policy network using the Q-value function.

4.2. Policy Networks

In PMaRL method, all agents train in a centralized manner and make decisions based on their own local information. Each agent only observes local information during decision-making and learns to cooperate during the training process. Each agent has its own policy network and the critic network is partially shared amongst all the agents. To learn agents' policies, we propose an approach using MaRL and SC network topology and we provide further details on this approach below.

State variables: In a multi-stage SC, agents are required to make decisions at every time step. The agent i interacts with other agents and the environment by observing state o_i^t, taking action a_i^t, observing new states o_i^{t+1}, and getting reward r_i^t. The local state of agent i, s_i^t, is a tuple $(BO_i^t, IL_i^t, OO_i^t, AS_i^t, AO_i^t)$, where BO_i^t, IL_i^t, OO_i^t, AS_i^t, and AO_i^t are backorder level, inventory level, on order items, arrival shipment, and arrival order at time t of agent i respectively. The observation o_i^t of the agent i at time step t contains its local states during the last m period: $(s_i^{t-m}, s_i^{t-m+1}, ..., s_i^t)$. m is set to 5 in our experiments [14].

Action space: In the actual scenario, each agent can order an arbitrary quantity, $[0, \infty)$, to its upstream agent(s). In MaRL, however, the policy networks output with finite size. Hence, the order quantity needs to be constrained to a finite action space, $\mathcal{A}$, which represents all possible actions an RL agent can take in the environment. To increase the robustness and stability of the policy network, we define each RL agent sends order quantity $d + a_i$ to its upstream agent(s), where d is total demand quantity from its downstream agent(s) [29] and a_i is the output of agent i's policy network. To regulate demand fluctuation of downstream agent(s), the policy network uses a_i to control how much to order based on

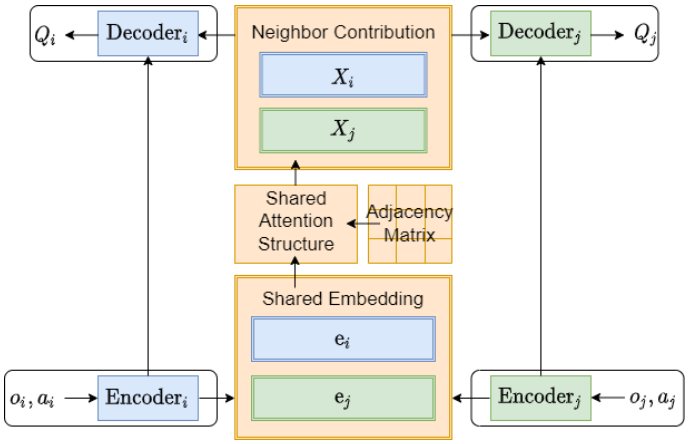


Figure 2. Overview of Critic Models.

the d. After the total order quantity is determined by $d + a_i$, it is then equally divided by the number of upstream agents and sent to them accordingly.

Reward function: At each time step t, agent i observes its own state o_i^t and selects an action a_i^t. The environment then provides a reward r_i^t and generates a new state s_i^{t+1}. To minimize the overall SC cost, r_i^t is measured based on agent's cost at time step t, which includes the sum of inventory holding and backorder costs. However, due to order and shipment lead times in a multi-stage SC, the reward is not immediately observed after taking action a_i^t. Moreover, the reward r_i^t reflects not only a single action but also the combined effect of joint actions from previous periods. As a result, decomposing r_i^t to evaluate the corresponding single action is challenging. So, to estimate action rewards, we define the local reward r_i^t of agent i at time step t as follows:

$$r_i^t = -\frac{1}{\Delta t} \sum_{t'=t+lt_i}^{t'+\Delta t} \gamma^{t'-t} \cdot c_r \cdot C_i^{t'},$$

where Δt represents a time window used to estimate the rewards of a given action within the future, lt_i is the order lead time of the agent i, c_r is the cost coefficient, and C_i^t is defined by Equation 2. Since the action a_i^t impacts the agent's rewards only after the products have arrived and, with the arrival time being at least $t + lt_i$, the local reward r_i^t for the agent i can be considered as the average of future costs within this time window. This allows agent i to approximate the reward for its action.

4.3. Critic Networks

Iqbal and Sha [14] proposed a Multi-Actor-Attention-Critic (MAAC) framework, which demonstrated exceptional performance in general MaRL scenarios. However, in SC networks, agents may be connected to varying numbers of upstream and downstream agents. Instead of adopting a broad approach like MAAC, where attention is evenly distributed among all agents, each agent should focuses more on its immediate neighbors and adopt a decision-making policy that enhances cooperation with adjacent agents.

To address this issue, we incorporate SC network topological information into our proposed PMaRL framework. An overview of the critic models is depicted in Figure 2. Generally, each agent has its own encoder and decoder layers, while the embedding and attention structure are shared across all the agents. The encoder layer generates state embedding by mapping the observation o into a lower-dimensional representation. By sharing these state embeddings, each agent obtains the data encapsulating the current states and actions of the entire SC. Using the topological information, the shared attention structure extracts the hidden relationships among agents from the agents' state embeddings, capturing the contributions of neighboring agents. Finally, the decoder maps these

contributions and state embeddings into the Q-values to assess the performance of the agent's action.

Initially, each agent's observation o_i^t and action a_i^t are fed into the agent-specific encoder f_i, which is a multi-perceptron (MLP) network. This process generates the state embedding e_i^t at each time step t as described by the following equation:

$$e_i^t = f_i(o_i^t, a_i^t).$$

To model the relationship among agents in critic networks, we use an adjacency matrix M, which is an $n \times n$ dimensional matrix representing network topology, where n is the number of agents. The SC networks can be considered as a directed graph. Each element $M_{i,j} \in [-1, 0, 1]$ captures the relation between the agent i and agent j. A value of "-1" indicates that agent i is the predecessor (upstream) of agent j, "0" means agent j is not the neighbor of agent i, and "1" signifies agent i is the successor (downstream) of agent j. This matrix enables each agent to focus on the information from its neighboring agents during training. For each agent i, the list of connected neighbors nb_i can be derived from the adjacency matrix.

To promote cooperation among agents, we use multiple attention heads for each agent in the shared attention structure, shown in Figure 3. Each head has its own set of parameters (W_q, W_k, W_v). It takes embeddings of agent i and its neighbors as input to generate the neighbor contribution X_i for agent i:

$$X_i = \sum_{j \in nb_i} \beta_j W_v e_j, \tag{10}$$

$$\beta_j \propto exp(e_j^T W_k^T W_q e_i) \tag{11}$$

where β_j is the attention weight calculated based on the embedding similarity between agent i and agent j using a bilinear mapping (i.e., scaled dot product operation). Each head captures the latent relationships extracted from both the neighbors' embeddings and the self-embedding. These latent relationships from all heads are then concatenated to form the final neighbor contribution X_i for agent i.

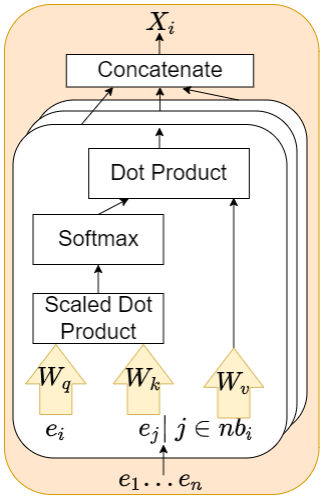


Figure 3. Shared attention structure [37].

Once the attention structure generates X_i, the decoder g, which a MLP networks, converts the X_i to the expected Q-value Q_i:

$$Q_i = g_i(e_i, X_i).$$

In the SC network, agents may differ in the number of input/output edges and, therefore, in their neighbor contributions. To classify the connectivity of agents, we define the number of incoming edges as the total number of downstream agents, and the number of outgoing edges as the total number of upstream agents. Agents with the same number of incoming and outgoing edges are grouped together based on the adjacency matrix, denoted as $G(i)$. $|G(i)|$ is a function used to determine the size of group $G(i)$. Agents within the same group are expected to have similar structures and inventory policies.

To train the critic model, we create a replay memory that stores historical transition data (s_t, a_t, s_{t+1}, r_t) to ensure training stability. At each training epoch, the loss function is defined as follows:

$$L_\theta = \sum_{i=1}^{N} \sigma_{G(i)} \left[(Q_i(o_i, a_i) - y_i)^2 \right],$$

$$\sigma_{G(i)} = \frac{1 + log(|G(i)|)}{|G(i)|}$$

where y_i is defined in Equation 9, $1 + log(|G(i)|)$ represents the log-scaled group weight for the group $G(i)$, and $\sigma_{G(i)}$ denotes the weight assigned to each agent within the group $G(i)$. This formula incorporates the log-scaled group weight into Equation 9 to balance the influence across groups of varying sizes and to encourage agents from different groups to explore.

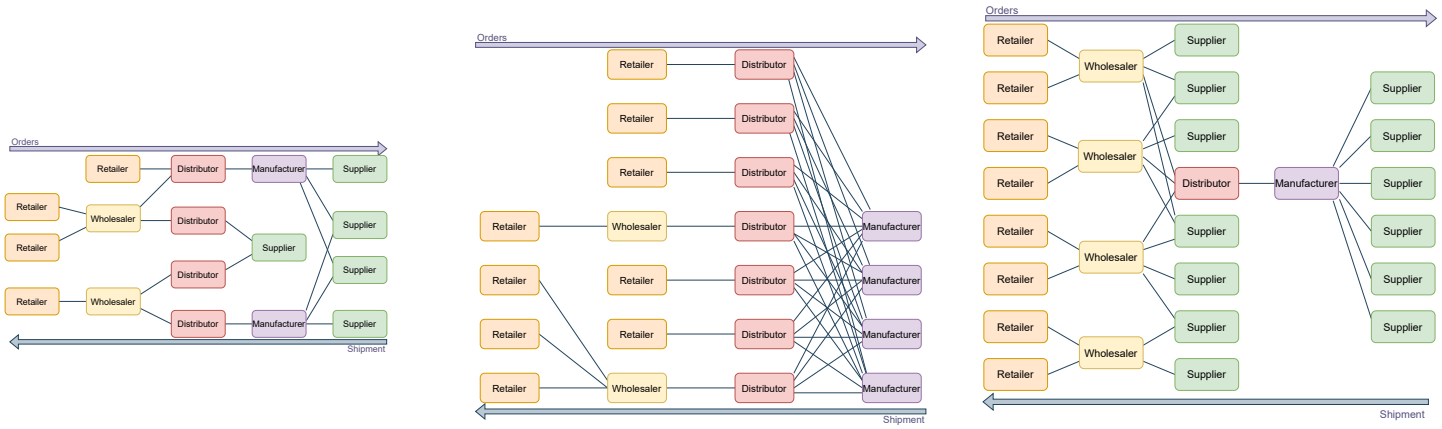
Agents with the same number of incoming and outgoing edges exhibit similar functionalities. To account for this, we assign log-scaled weights based on group sizes. In contrast, treating every agent equally, as done in the MAAC approach [14], can lead to critic model disproportionately emphasizing the loss of agents in larger groups. This can lead to bias toward optimizing the loss of these agents, thereby discouraging exploration by agents in small groups. For example, if an SC network consists of 100 retailers and one wholesaler, the MAAC approach might cause the critic model to focus primarily on the retailers due to their similar structure and functionality. As a result, the wholesaler's critic model may adopt retailers' policy to minimize the cumulative cost, limiting the scope of exploration. In such cases, the MAAC approach often leads to local minimum and inadequate exploration. Unfortunately, real SC networks frequently have a large number of agents within the same group [26]. The MAAC approach may struggle to balance the exploration and exploitation, leading to suboptimal SC performance, as we will demonstrate in Section 5.

In PMaRL, we tackle this issue by incorporating log-scaled group weights, allowing the critic model to fairly assess the actions of agents across different groups. The log-scaled group weight for group $G(i)$ is calculated as $1 + \log(|G(i)|)$. The weights assigned to agents within the same group, $\sigma_{G(i)}$, are distributed equally according to these log-scaled group weights. Compared to MAAC approach, this method ensures that the agents in small groups have a greater impact on the loss function, enabling the critic models to better understand and address the imbalance in SC networks topology.

Subsequently, we apply the loss function to all agents and update their policy networks gradients using Equation 8. The baseline $b(s)$ is set to $\mathbb{E}[Q_i(o_i, (a_i, a_{\setminus i}))]$ to estimate the value of the current state. Each agent samples the actions $a_{\setminus i}$ of other agents from their current policies to estimate its own gradient. Essentially, Equation 8 compares the value of the current action with the average action values of other agents to evaluate whether the current action contributes to the increased rewards.

5. EXPERIMENTS & EVALUATION

In this section, we present a multi-stage SC simulation as an evaluation testbed. The experiments evaluate SC performance in terms of cost for various approaches, as well as the convergence of different MaRL methods. The simulation experiments were conducted on a platform equipped with an 11th Gen Intel Core i9 11900KF CPU, 64 GB memory, and NVIDIA RTX 3090 GPU.



(a) R1. **(b)** R2. **(c)** R3.

Figure 4. Three Real SC Networks [26].

Table 1. SC Network Characteristics.

SC Network	No. Stages	No. Nodes	No. Edges	Network Complexity
Beer Game (BG)	4	4	3	1.5
Real Network 1 (R1)	5	17	18	2.82
Real Network 2 (R2)	4	22	39	1.91
Real Network 3 (R3)	5	27	31	2.27

5.1. Simulation Setup

To evaluate the performance of various inventory management approaches across different SC networks, we developed an SC simulation model as a test bed for assessing total SC costs. In this simulation model, each agent represents an entity within the SC network and is connected according to the network's topology. At every time step of the simulation, each agent follows its respective inventory management policy to determine an order quantity, which is then sent to its upstream partner(s). The upstream agent fulfills the order based on its current inventory levels and initiates a shipment. If the order quantity exceeds the available inventory, the unfulfilled portion is recorded as a backorder, which incurs additional backorder costs. Upon receiving a new shipment, agents prioritize fulfilling any outstanding backorders. The total SC costs are calculated as outlined in Equation 2.

In our experiments, we construct four distinct SC networks based on the beer game (BG) [38] and three real SC networks (R1, R2, and R3) [26], which are illustrated in Figure 4 and Table 1 summarizes their characteristics [39]. The columns labeled *Stages*, *Nodes*, *Edges*, and *Network Complexity* correspond to the number of stages, nodes, and edges in each SC network. *Network Complexity* represents the uncertainty inherent in dynamic logistics within the SC systems [39].

As shown in Table 1, the real SC networks are selected to showcase the SC performance at different levels of network complexity. For instance, R1 and R3 exhibit higher network complexity with numerous linkages (see Figures 4a and 4c). In contrast, R2 shows lower network complexity, featuring a simple and repetitive connection topology among *wholesaler* agents (see Figure 4b). The objective is to evaluate the SC costs of the PMaRL model across different network complexity levels. For each network topology, we adjust the shipment lead time from 1 to 3 days. Increasing the lead time results in higher inventory and backorder costs, thereby allowing us to assess the total cost under various scenarios.

In the simulation, customer demand for agent i at each time step is modeled as $\mathcal{P}(\lambda_i)$, where $\mathcal{P}$ denotes Poisson distribution, and λ_i is the mean of Poisson distribution. The Poisson distribution is chosen because it is commonly used for demand pattern generation [27]. To ensure a fair comparison, the total average demand generated across all retailers in an

Table 2. SC Network Parameters.

SC Network	Retailers	Retailer Demand	Action Space $\mathcal{A}$
BG	1	$d_1^t \sim \mathcal{P}(32)$	{-8, 24}
R1	4	$d_1^t = d_2^t = d_3^t = d_4^t \sim \mathcal{P}(8)$	{-4, 10}
R2	9	$d_1^t = d_2^t = ... = d_9^t \sim \mathcal{P}(32/9)$	{-3, 6}
R3	8	$d_1^t = d_2^t = ... = d_8^t \sim \mathcal{P}(4)$	{-3, 6}

Table 3. Differences between MaRL methods.

Methods	Critic Network	Policy Network
MAAC	without SC network topology	Local Information
PMaRL	with SC network topology	Local Information
FV	with SC network topology	Global Information

SC is fixed at 32, i.e., $\sum \lambda_i = 32$ [40]. The action spaces and the demand values are defined in Table 2:

The number of retailers varies by network: there are 1, 4, 9, and 8 retailers in BG, R1, R2, and R3 respectively based on the SC networks topology in the dataset. The action space $\mathcal{A}$ is influenced by the Poisson distribution, with lower Poisson means leading to smaller action spaces [29].

In the SC simulation, the duration is set to 50 time steps, each representing one day. The first 5 days serve as a warm-up period. The total cost Z (see Equation 2) is calculated by summing the backorders and holding costs of all agents in the SC after the warm-up period. To ensure the robustness of experimental results, the total costs are averaged over 100 simulation runs.

5.2. SC Inventory Management Settings

We compare our proposed PMaRL method with other SC inventory management methods, including both traditional and MaRL approaches, across different SC networks, as shown in Table 1.

For traditional methods, the first comparison is the base-stock (BS) policy, where a firm places a replenishment order whenever inventory falls below a predetermined base-stock level [41]. This base-stock level is calculated using the formula $\overline{D} * LT + Z_{ser} * std(D) * \sqrt{LT}$, where D is the set of retailer demands, LT is the order lead time, Z_{ser} is the standard score for a 95% service level, and $std(D)$ is the standard deviation of demand. The second traditional approach is the decomposition-aggregation (DA) heuristic, which approximates the optimal base-stock level by breaking down the SC network into simpler serial SCs [25]. DA calculates the optimal base-stock levels for these serial SCs and then aggregates the results to derive a heuristic base-stock level. Similarly, the service level in DA is also set to 95%. The key distinction between these methods is that the BS approach uses local information to determine the base-stock level, while DA uses the information from all entities in the network to approximate the optimal level.

For the ML-based approaches, we also compare two methods. The first is the MAAC approach [37], which allows agents to exchange their attention embeddings for better coordination. The second is our proposed PMaRL method. The last method is a variation of the PMaRL approach with full visibility (FV). The differences between the MaRL approach in comparison are in Table 3. In the critic network, both PMaRL and FV models utilize the SC network topology, whereas the MAAC does not. In the policy network, both MAAC and PMaRL reply only on local information as input, while the FV model requires information from all agents.

5.3. Results - Comparison with Traditional Methods

Figure 5 presents a comparison of the normalized costs across different methods for various lead times, with costs normalized against those of the PMaRL approach. In the

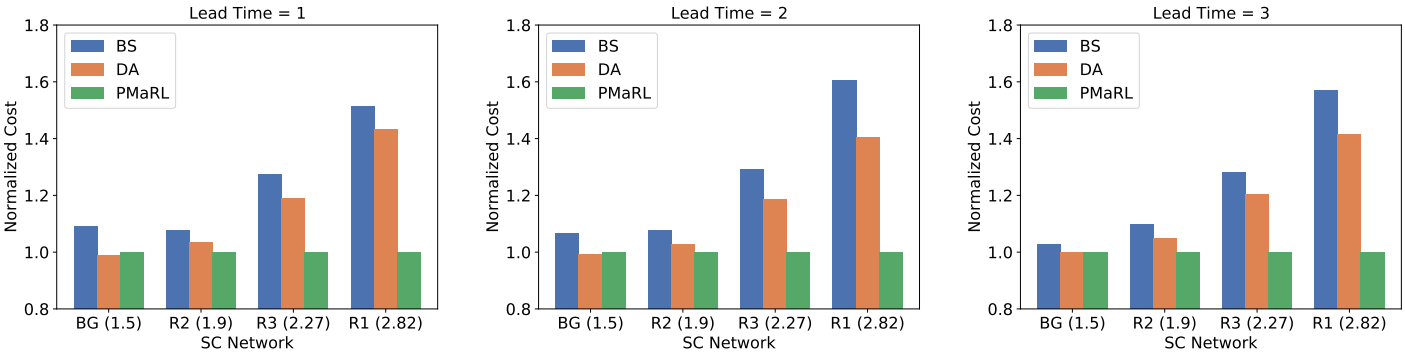


Figure 5. Normalized cost of different optimization methods with increasing network complexity. The value in parentheses denotes the network complexity.

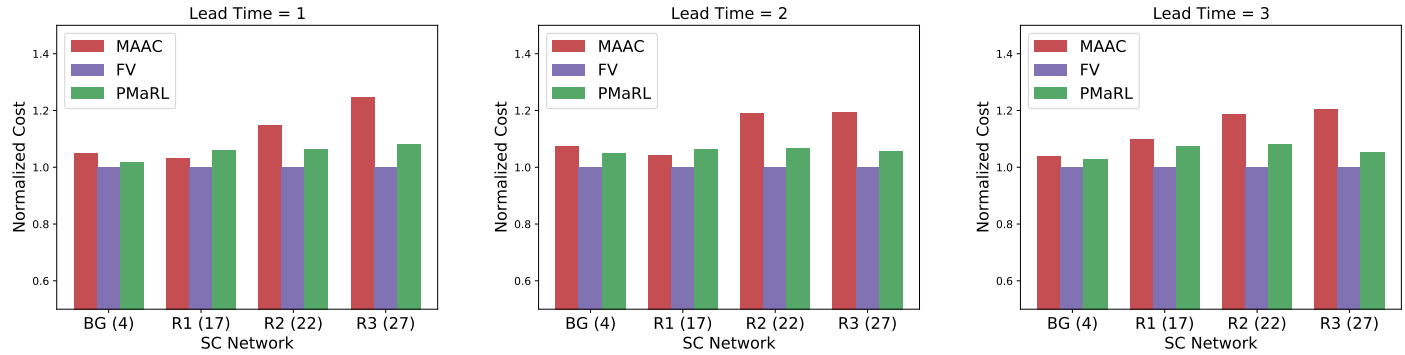


Figure 6. Normalized costs of different MaRL methods with increasing number of nodes. The value in parentheses denotes the number of nodes.

figure, the green, orange, and blue bars represent the normalized cost for the BS, DA, and PMaRL approaches, respectively.

The data in Figure 5 indicates that as the network complexity increases, and the PMaRL approach consistently outperforms other methods, achieving greater cost savings in the SC. Specifically, the PMaRL approach reduces inventory costs by a factor of 1.4 to 1.6 compared to other methods in high network complexity scenarios. The PMaRL approach surpasses the BS method in all cases and shows notable inventory cost savings in high network complexity scenarios. The DA approach outperforms BS due to its increased visibility. The PMaRL approach achieves comparable performance to DA in low network complexity scenarios and significantly better performance in high network complexity scenarios.

Overall, the experiment results consistently demonstrate that the PMaRL approach effectively reduces inventory costs, underscoring its effectiveness and versatility in handling complicated SC systems. Its ability to achieve cost reductions that benefit all entities highlights the practical value and applicability of the PMaRL approach in real-world SC scenarios.

5.4. Results - Comparison amongst Different Multi-agent based RL Methods

Figure 6 demonstrates the normalized cost for different MaRL methods under various lead times, with costs normalized against those of the FV approach. In the figure, the red, purple, and green bars represent the MAAC, FV, and PMaRL methods, respectively. The comparison with the FV approach highlights the differences in performance when full visibility is available versus when it is restricted, whereas the comparison with the MAAC approach emphasizes the advantages of PMaRL method, which incorporates network topology and logarithmic group weight scaling.

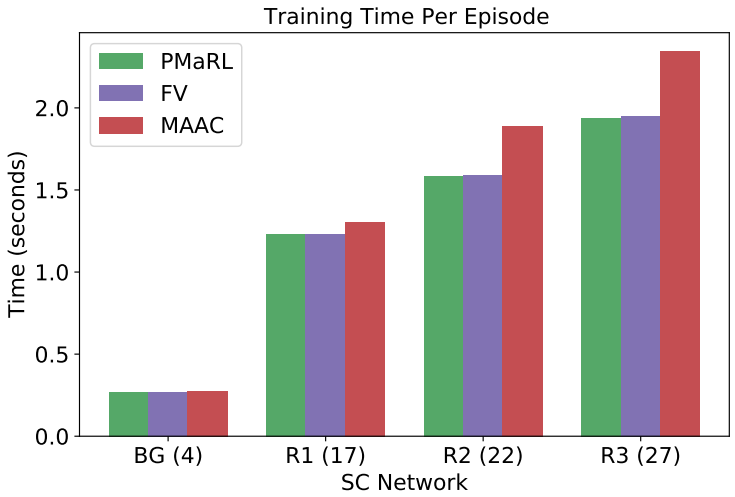


Figure 7. Average training time per episode of different MaRL methods with increasing number of nodes. The value in parentheses denotes the number of nodes.

In general, the PMaRL approach outperforms the MAAC approach as the number of nodes in SC network increases. This is because, in larger networks with numerous agents that have similar structures and functions, the MAAC approach tends to focus on minimizing the losses of these agents, potentially leading to local optima and limiting further exploration. By incorporating grouping based on network topology, the PMaRL approach strikes a more effective balance of exploration and exploitation, preventing premature convergence to suboptimal solutions.

When comparing the PMaRL approach to the FV approach, we observe comparable performance across all scenarios. This demonstrates that the PMaRL approach is capable of effective cooperation even in cases where information sharing is restricted or where privacy concerns exist. This capability is crucial for maintaining data privacy while still achieving performance levels on par with full visibility models.

5.5. Results - Training Time and Convergence

Figure 7 compares the average training time per episode amongst the PMaRL, FV, and MAAC approaches. The variations between different training episodes are negligible. The PMaRL approach is slightly faster than the FV approach. This small difference arises because the PMaRL approach only requires local information as input in its policy network, while the FV approach processes global information. As the number of nodes in the SC network increases, the PMaRL approach demonstrates faster training time per episode, leveraging network topology to focus agents' attention on their neighbors. This helps learning to focus on relevant information, thereby reducing training time. In contrast, the MAAC approach requires agents to compute attention across all other agents, leading to distractions from less relevant information and slower training time.

Figure 8 shows the normalized inventory cost curves during the training process for R3, with the cost normalized against the corresponding BS approach costs. The green, red, and purple lines represent the inventory costs of the PMaRL, MAAC, and FV approaches, respectively. Although the MAAC approach converges faster, the PMaRL approach achieves lower overall inventory costs, further demonstrating its ability to effectively balance exploration and exploitation during training. Compared to the FV approach, the PMaRL approach yields similar inventory costs but converges more quickly.

5.6. Discussion

Overall, the PMaRL approach consistently outperforms traditional methods in complex SC networks. This success is due to the adaptability of the MaRL model, which

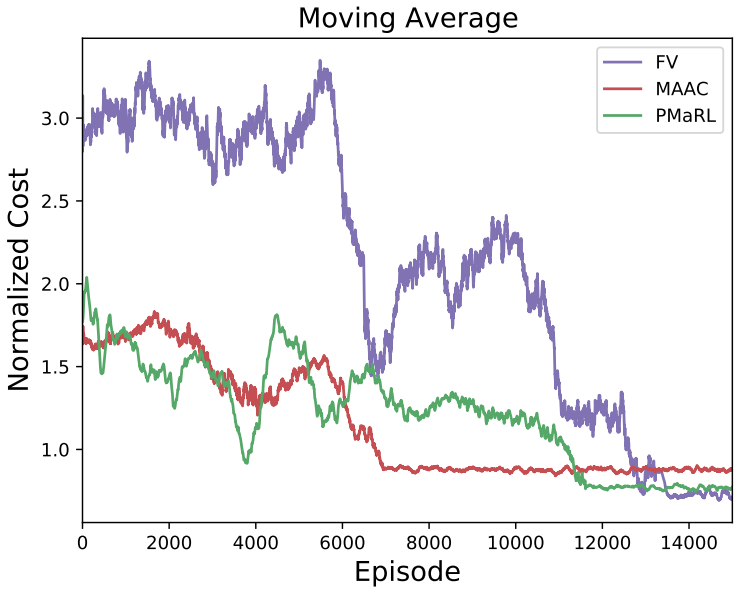


Figure 8. Convergence of different MaRL methods.

generates effective decisions across various scenarios. Even when the demand patterns change, the well-trained PMaRL model can quickly adapt with minimal additional data. This efficiency is because the model has already learned inventory control logic and collaboration strategies, requiring only adjustments to fit the new demand distribution.

Traditional methods typically excel in simple SC networks but struggle with large and complex ones. In contrast, the PMaRL approach utilizes network topology information and explores various inventory policies to establish a robust strategy for the entire SC network. The MaRL approach enables policy models of agents to learn collaboratively during training, enhancing coordination in distributed decision-making.

The PMaRL approach is particularly effective for optimizing complex SC networks because it leverages network topology information and does not require prior knowledge for decision-making. In comparison, the MAAC approach performs well only in simple SC network topologies. The topology of an SC network has a major impact on the agent's actions. As each agent's demand originates exclusively from its downstream, the behaviors of the agent are significantly influenced by its neighbors. Without incorporating network topology information, the MAAC approach fails to fully explore these relationships, leading to higher inventory costs. The PMaRL approach, on the other hand, incorporates a shared attention structure that utilizes network topology and incorporates logarithmic scale group weights in the loss function, making it more adept at handling complex SC network scenarios. Consequently, as the number of nodes in the SC network increases, the PMaRL approach consistently outperforms the MAAC approach.

The PMaRL approach delivers similar performance with faster convergence compared to the FV approach. This indicates that agents can effectively cooperate while making decisions based on local information. The FV approach, however, requires full visibility of information from all agents, which poses privacy concerns in real-world SC scenarios. The PMaRL approach addresses these privacy concerns by sharing only partial information during the training process, allowing agents to make independent decisions during actual operation.

6. CONCLUSION

This paper introduces a novel data-driven decision-making approach using a MaRL model that leverages network topology to optimize inventory costs across the entire SC. By enhancing SC coordination and promoting decentralized decision-making, the model

demonstrates significant improvements. Our experiments in SC inventory cost reduction show that the proposed MaRL model outperforms traditional approaches and the original MaRL model, achieving performance comparable to the FV method. Therefore, our work offers an approach that strikes a balance between model performance and data privacy.

Moving forward, our future research will focus on enhancing privacy protection for firms by integrating Federated Learning (FL) into our model. This approach will allow firms to train their models locally without sharing raw data. By combining FL with MaRL, we aim to reduce the risk of data leakage while simultaneously improving performance through better SC coordination among firms.

Author Contributions: Conceptualization, B.Z., W.J.T., W.C. and A.N.Z; methodology, B.Z.; software, B.Z.; validation, B.Z; formal analysis, B.Z.; investigation, B.Z.; resources, B.Z.; data curation, B.Z.; writing—original draft preparation, B.Z. and W.J.T.; writing—review and editing, W.C. and A.N.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2022-031).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Acknowledgments: This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2022-031). Bo Zhang is supported under NTU PhD scholarship.

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

References

1. Panahifar, F.; Byrne, P.J.; Salam, M.A.; Heavey, C. Supply chain collaboration and firm's performance: the critical role of information sharing and trust. *Journal of Enterprise Information Management* **2018**.
2. Zhang, B.; Tan, W.J.; Cai, W.; Zhang, A.N. Forecasting with Visibility Using Privacy Preserving Federated Learning. In Proceedings of the 2022 Winter Simulation Conference (WSC). IEEE, 2022, pp. 2687–2698.
3. Sánchez-Flores, R.B.; Cruz-Sotelo, S.E.; Ojeda-Benitez, S.; Ramírez-Barreto, M.E. Sustainable supply chain management—A literature review on emerging economies. *Sustainability* **2020**, *12*, 6972.
4. Zhang, A.N.; Goh, M.; Meng, F. Conceptual modelling for supply chain inventory visibility. *International Journal of Production Economics* **2011**, *133*, 578–585.
5. Zavala-Alcívar, A.; Verdecho, M.J.; Alfaro-Saiz, J.J. A conceptual framework to manage resilience and increase sustainability in the supply chain. *Sustainability* **2020**, *12*, 6300.
6. Nikolopoulos, K.; Punia, S.; Schäfers, A.; Tsinopoulos, C.; Vasilakis, C. Forecasting and Planning During a Pandemic: COVID-19 Growth Rates, Supply Chain Disruptions, and Governmental Decisions. *European Journal of Operational Research* **2021**, *290*, 99–115.
7. Rolf, B.; Jackson, I.; Müller, M.; Lang, S.; Reggelin, T.; Ivanov, D. A review on reinforcement learning algorithms and applications in supply chain management. *International Journal of Production Research* **2022**, pp. 1–29.
8. Jackson, I.; Ivanov, D.; Dolgui, A.; Namdar, J. Generative artificial intelligence in supply chain and operations management: a capability-based framework for analysis and implementation. *International Journal of Production Research* **2024**, pp. 1–26.
9. Lazar, S.; Klimecka-Tatar, D.; Obrecht, M. Sustainability orientation and focus in logistics and supply chains. *Sustainability* **2021**, *13*, 3280.
10. Ramanathan, U. Performance of supply chain collaboration–A simulation study. *Expert Systems with Applications* **2014**, *41*, 210–220.
11. Chen, Y.; Özer, Ö. Supply Chain Contracts that Prevent Information Leakage. *Management Science* **2019**, *65*, 5619–5650.
12. Kumar, A.; Shrivastav, S.K.; Shrivastava, A.K.; Panigrahi, R.R.; Mardani, A.; Cavallaro, F. Sustainable supply chain management, performance measurement, and management: a review. *Sustainability* **2023**, *15*, 5290.
13. Barat, S.; Khadilkar, H.; Meisheri, H.; Kulkarni, V.; Baniwal, V.; Kumar, P.; Gajrani, M. Actor based simulation for closed loop control of supply chain using reinforcement learning. In Proceedings of the Proceedings of the 18th international conference on autonomous agents and multiagent systems, 2019, pp. 1802–1804.
14. Iqbal, S.; Sha, F. Actor-attention-critic for multi-agent reinforcement learning. In Proceedings of the International conference on machine learning. PMLR, 2019, pp. 2961–2970.
15. Shang, K.H.; Song, J.S. Newsvendor bounds and heuristic for optimal policies in serial supply chains. *Management Science* **2003**, *49*, 618–638.

16. Chen, L.; Dong, T.; Peng, J.; Ralescu, D. Uncertainty analysis and optimization modeling with application to supply chain management: a systematic review. *Mathematics* **2023**, *11*, 2530.
17. Barlas, Y.; Gunduz, B. Demand Forecasting and Sharing Strategies to Reduce Fluctuations and the Bullwhip Effect in Supply Chains. *Journal of the Operational Research Society* **2011**, *62*, 458–473.
18. Somapa, S.; Cools, M.; Dullaert, W. Characterizing Supply Chain Visibility–a Literature Review. *The International Journal of Logistics Management* **2018**.
19. Yang, D.; Zhang, A.N. Impact of information sharing and forecast combination on fast-moving-consumer-goods demand forecast accuracy. *Information* **2019**, *10*, 260.
20. Feizabadi, J. Machine Learning Demand Forecasting and Supply Chain Performance. *International Journal of Logistics Research and Applications* **2022**, *25*, 119–142.
21. Ried, L.; Eckerd, S.; Kaufmann, L.; Carter, C. Spillover Effects of Information Leakages in Buyer–Supplier–Supplier Triads. *Journal of Operations Management* **2021**, *67*, 280–306.
22. Tan, K.H.; Wong, W.P.; Chung, L. Information and Knowledge Leakage in Supply Chain. *Information Systems Frontiers* **2016**, *18*, 621–638.
23. Saha, E.; Ray, P.K. Modelling and analysis of inventory management systems in healthcare: A review and reflections. *Computers & Industrial Engineering* **2019**, *137*, 106051.
24. Fokouop, R.; Sahin, E.; Jemai, Z.; Dallery, Y. A heuristic approach for multi-echelon inventory optimisation in a closed-loop supply chain. *International Journal of Production Research* **2024**, *62*, 3435–3459.
25. Rong, Y.; Atan, Z.; Snyder, L.V. Heuristics for base-stock levels in multi-echelon distribution networks. *Production and Operations Management* **2017**, *26*, 1760–1777.
26. Willems, S.P. Data set—Real-world multiechelon supply chains used for inventory optimization. *Manufacturing & service operations management* **2008**, *10*, 19–23.
27. Lesnaia, E. Optimizing safety stock placement in general network supply chains. PhD thesis, Massachusetts Institute of Technology, 2004.
28. Ahmadi, E.; Masel, D.T.; Hostetler, S. A robust stochastic decision-making model for inventory allocation of surgical supplies to reduce logistics costs in hospitals: A case study. *Operations Research for Health Care* **2019**, *20*, 33–44.
29. Oroojlooyjadid, A.; Nazari, M.; Snyder, L.V.; Takáč, M. A deep q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manufacturing & Service Operations Management* **2022**, *24*, 285–304.
30. Fuji, T.; Ito, K.; Matsumoto, K.; Yano, K. Deep multi-agent reinforcement learning using dnn-weight evolution to optimize supply chain performance. *Hawaii International Conference on System Sciences* **2018**.
31. Mortazavi, A.; Khamseh, A.A.; Azimi, P. Designing of an intelligent self-adaptive model for supply chain ordering management system. *Engineering Applications of Artificial Intelligence* **2015**, *37*, 207–220.
32. Wang, F.; Lin, L. Spare parts supply chain network modeling based on a novel scale-free network and replenishment path optimization with Q learning. *Computers & Industrial Engineering* **2021**, *157*, 107312.
33. Nurkasanah, I. Reinforcement learning approach for efficient inventory policy in multi-echelon supply chain under various assumptions and constraints. *Journal of Information Systems Engineering and Business Intelligence* **2021**, *7*, 138–148.
34. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems* **1999**, *12*.
35. Konda, V.; Tsitsiklis, J. Actor-critic algorithms. *Advances in neural information processing systems* **1999**, *12*.
36. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International conference on machine learning. PMLR, 2018, pp. 1861–1870.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
38. Chen, F.; Samroengraja, R. The stationary beer game. *Production and Operations Management* **2000**, *9*, 19–30.
39. Cheng, C.Y.; Chen, T.L.; Chen, Y.Y. An analysis of the structural complexity of supply chain networks. *Applied Mathematical Modelling* **2014**, *38*, 2328–2344.
40. Zhao, Y. Evaluation and optimization of installation base-stock policies in supply chains with compound Poisson demand. *Operations Research* **2008**, *56*, 437–452.
41. Graves, S.C.; Willems, S.P. Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management* **2000**, *2*, 68–83.