IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 August 2022, accepted 4 September 2022, date of publication 12 September 2022, date of current version 27 September 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3205720*

RESEARCH ARTICLE

# Optimization of Apparel Supply Chain Using Deep Reinforcement Learning

**JI WON CHONG[1], WOOJU KIM[1], AND JUNESEOK HONG[2]**
[1]Department of Industrial Engineering, Yonsei University, Seodaemun-gu, Seoul 03722, South Korea
[2]Department of Management Information Systems, Kyonggi University, Yeongtong-gu, Suwon-si, Gyeonggi-do 16227, South Korea

Corresponding author: Wooju Kim (wkim@yonsei.ac.kr)

This work was supported by the Jungseok Logistics Foundation.

**ABSTRACT** An effective supply chain management system is indispensable for an enterprise with a supply chain network in several aspects. Especially, organized control over the production and transportation of its products is a key success factor for the enterprise to stay active without damaging its reputation. This case is also highly relevant to garment industries. In this study, an extensive Deep Reinforcement Learning study for apparel supply chain optimization is proposed and undertaken, with focus given to Soft Actor-Critic. Six models are experimented with in this study and are compared with respect to the sell-through rate, service level, and inventory-to-sales ratio. Soft Actor-Critic outperformed several other state-of-the-art Actor Critic models in managing inventories and fulfilling demands. Furthermore, explicit indicators are calculated to assess the performance of the models in the experiment. Soft Actor-Critic achieved a better balance between service level and sell-through rate by ensuring higher availability of the stocks to sell without overstocking. From numerical experiments, it has been shown that S-policy, Trust Region Policy Optimization, and Twin Delayed Deep Deterministic Policy Gradient have a good balance between service level and sell-through rate. Additionally, Soft Actor-Critic achieved a 7%, 41.6%, and 42.8% lower inventory sales ratio than the S-policy, Twin Delayed Deep Deterministic Policy Gradient, and Trust Region Policy Optimization models, indicating its superior ability in making the inventory stocks available to make sales and profit from them.

**INDEX TERMS** Deep reinforcement learning, inventory management, markov decision process, supply chain management, soft actor critic.

## I. INTRODUCTION

The introduction section contains two subsections: 1) motivation of the study and 2) contribution of the study. The illustration of this section is given as follows:

### A. MOTIVATION

Supply Chain Management (SCM) is practically the backbone of the success of enterprises, especially with growing global trade competition staying intact. In fact, enterprises need to continuously ensure their SCM operations are efficient so they gain the competitive advantage in the marketplace [1], and inventory management (IM) is an essential determinant for a successful SCM operation. Inventory management refers to a task through which an enterprise must make ordering decisions periodically to meet the demands of its product. A competitive inventory management is also crucial in the apparel industry. Various approaches for this task were oftentimes made with mathematical approaches such as, linear programming, dynamic programming, and heuristics models. In case of heuristics policy, it was recently applied to cash-constrained inventory management [2]. In this study, a cash-constrained small retailer intermittently purchases a product and offers it to clients with non-stationary demand on its way. Heuristics method was also applied to an intermittent review multi-item stock framework with exogenous lot sizes and backordering [3]. These approaches can optimize small, discrete-based settings. Unfortunately, they suffer from considerable drawbacks. Heuristics policies, for instance, are not scalable enough to large problem instances. In addition, these policies require significant domain knowledge, relying on restrictive modelling assumptions at times. Same is the

The associate editor coordinating the review of this manuscript and approving it for publication was Alvis Fong.

case for linear and dynamic programming methods. Curse of dimensionality is problematic for these as the scale of the problem becomes larger, making the solution intractable [4]. In order to circumvent this difficulty, deep neural networks (DNN) started to receive wider consideration for supply chain inventory management (SCIM) problems. Particularly, Deep Reinforcement Learning (DRL) became notable for supply chain management due to its feasibility in solving problems that do not particularly require specific data distribution information or restrictive assumptions. Demand is highly critical in inventory management. Demand varies as customers' tastes and preferences change, making it challenging for inventory managers to plan the right storage to minimize costs emanating from overstocking and stockouts. Various works were undertaken for inventory optimization with DRL recently. However, there is a considerable lack of research of data-driven inventory control with DRL [5]. In fact, most of previous works utilize demands with explicit pattern from the sinusoidal function or those sampled from fundamental statistical distributions, such as Normal or Poisson distribution. In addition, to the best of one's knowledge, all the DRL methods used for this task were discrete-based DRL, limiting the scope of its application to problems that involve large action spaces. Another problem with these works is that they focus on demonstrating their results based on reward values, without considering key performance indicators (KPIs) that indicate how effectively a given inventory management system is functioning.

### B. CONTRIBUTION

These are the gaps that will be filled by utilizing an advanced DRL model called Soft Actor Critic [6] that will make data-driven decisions for proper inventory management. Furthermore, the approach to the problem is more practical by considering as many retailers as the number of regions is present in the dataset, addressing the need to consider SCIM's realistic number of entities [7]. Particularly, Soft Actor-Critic (SAC) is a suitable model for handling demands with inexplicit pattern and distribution and its extensive exploration of actions allows it to readily deal with those demands for a profitable inventory management system. The contributions of this study are summarized as follows:

- A make-to-stock (MTS) inventory management for apparel products is designed to ensure product traceability from the manufacturer to customers.
- Extensive study with various DRL models is conducted with explicit inventory management performance illustration.
- Data-driven DRL study is conducted with SAC for an optimal inventory management. To the best of one's knowledge, no other study has been undertaken on inventory management with SAC.

### C. ORGANIZATION OF STUDY

The remainder of this study is organized as follows. Related works on multi-echelon-based inventory management and the application of DRL to inventory control systems are discussed in Section II. Section III explains the problem statement and the supply chain model. Section IV explains the SAC method that is used for approaching the problem. In Section V, experiments and numerical results are shown to demonstrate the comparative performance of SAC and how it establishes a well-balanced inventory management. VI contains the experimental results and subsequent graphs. Finally, section VII concludes the study with potential future work.

## II. LITERATURE REVIEW

Various previous works addressed inventory control with non-RL and RL techniques, which includes DRL techniques.

### A. INVENTORY MANAGEMENT WITH MULTI-ECHELON SC

Various mathematical studies for inventory management with multi-echelon SC have been undertaken. Qian *et al.* [8] developed a joint inventory and emission optimization framework for a multi-echelon SC under stochastic demands. Effective operation of a supply chain system can also be influenced by certain regulatory actions. Liu *et al.* [9] addressed the effect of such actions by conducting a new multi-echelon SC viability problem, which is exposed to limited intervention budget. The method involved combination of multiple mathematical approaches. In SC, demand can be formulated under various assumptions. Xiang *et al.* [10] solved a nonstationary stochastic inventory problem using "off-the-shelf" mixed integer linear programming solvers, which proved to be highly robust to demands under such assumptions. A literature gap has been found in these studies where the model is first proposed and validated through data. No work has yet been done in synchronizing the dynamic nature of an apparel supply chain environment with reinforcement learning (RL).

### B. REINFORCEMENT LEARNING IN INVENTORY MANAGEMENT

RL is a sub-area in ML in which the decision is made sequentially following the Markov Decision Process (MDP), requiring no explicit mathematical or statistical assumptions. RL has successfully been applied to various problems associated with SCM. Such problems range from scheduling, manufacturing, to inventory management. This application became prominent due to the limited scalability of traditional methods to larger scale problems, calling the need for AI techniques for this task. In particular, RL techniques have been considered to optimize ordering decisions in SC. [11] conducted a study on RL for optimal ordering decision in the beer distribution game, which consisted of a serial supply chain network. Researchers also discovered the importance of the state configuration as the input to address additional aspects. [12], for instance, discovered the importance of age information for inventory control of perishable products using Q-learning [13], and SARSA [13]. Unfortunately, such algorithms can only handle problems of limited state and action space sizes, encouraging researchers to use DRL algorithms, which uses deep learning models as the

policy approximators. Reference [4] performed joint replenishment under full-truckload transportation method using PPO [14]. [15] developed an agri-food SCM optimization system using Q-learning and DQN. Both algorithms outperform the "heuristics" method, achieving highest convergence in reward values during the training phase. DRL methods were also considered for inventory control of multiple products. In order to drive such problem effectively, [16] utilized multi-agent reinforcement learning (MARL) method to effectively replenish the products without giving unfair treatment to certain products. In a recent study by [17], the reference price was found to be influential for pricing and ordering decisions, and DRL model achieved the highest profit. Most, if not all studies of DRL in inventory control assume that the uncertainty or value of demand is known in advance. Hardly any study was made with demand represented as real-case data. Boute *et al.* [5] addressed this as one of the avenues for future research, claiming that most papers exploring DRL in inventory management neglects the influence of using data in the decision-making process. This literature gap has been fulfilled in the present study with the application of DRL using the raw data of product demand.

**TABLE 1. Notations and their corresponding descriptions in the model.**

| Notation | Explanation |
|---|---|
| $s_{F,t}$ | manufacturer's inventory level |
| $s_{w_i,t}$ | each retailer's inventory level |
| $p_t$ | production amount |
| $a_{w_i,t}$ | transportation amount to each retailer |
| $Cap_F$ | factory warehouse capacity |
| $Cap_{w_i}$ | each retailer's warehouse capacity |
| $c_t$ | penalty at timestep t |
| $d_{i,t}$ | retailer i's demand at timestep t |
| $W$ | total number of retailers |
| $N$ | length of episode |

## III. PROBLEM STATEMENT

In this section, the characteristics of the apparel inventory management problem has been identified. In addition, the procedure by which the algorithms follow to establish optimal emission policies has been incorporated. The present inventory management problem involves a two-echelon supply chain with single manufacturer and multiple retailers. The product category is apparel, and it is single-type and non-perishable. The Manufacturer follows MTS system, where it continuously produces and sells Q quantity of finished goods to the retailers at every time cycle t. The upcoming demands at each timestep is highly volatile and due to this instability, shortages and surplus tend to occur. These are critical problems in inventory management because its main concern lies on minimizing total inventory cost while satisfying demands [18]. The policies are identified as the production and transportation quantities of goods, which are decided by the RL agent, which is the manufacturer in this case. The agent continuously interacts in the IM environment to optimize its policy by learning to take better actions. Such actions, if suboptimal, lead to costs associated with overstocking

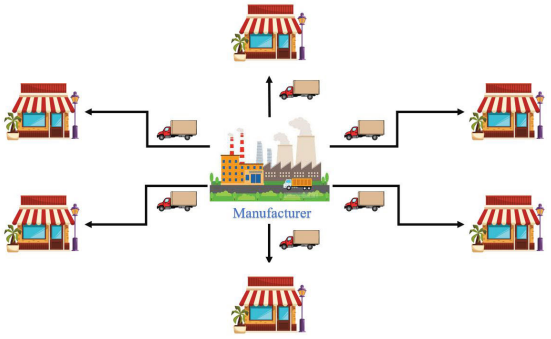



**FIGURE 1. Scenario of SCM environment.**

and stockouts. In order to minimize these incidences, the aforementioned actions must be made adequately so the manufacturer is ready to emit sufficient amounts for retailers to fulfill demands without incurring unnecessary storage costs. The proposed model is visualized in Figure (1). Description of every notation involved in the model is included in Table (1). The MDP formulation of the problem has been discussed. The configurations of the input state, action, and reward functions are demonstrated for a clear picture. The manufacturer and retailers are each assigned their capacities $Cap_F$ and $Cap_{w_i}$, where $\forall \epsilon\{1, 6\}$. The interaction process of the agent begins with the initial inventory levels $s_{F,0}$ and $s_{w_i,0}$, of which values are sampled from their corresponding ranges: $s_{F,0} \sim [0, Cap_F]$ and $s_{w_i,0} \sim [0, Cap_{w_i}]$. In addition to these, demand history is given as additional state information for the agent to anticipate upcoming demands. This demand history consists of each regional demand values of previous seven timesteps from each successive current timestep. Using these values, the agent performs its initial actions and continues to learn to perform actions with new current inventory levels and successive demand history at each successive timestep:

$$s_t = \{\text{Inventory levels, Demand history}\}. \quad (1)$$

$$\text{Inventory levels} = \{s_{F,t}, s_{w_i,t}\}. \quad (2)$$

$$\text{Demand history} = \{d_{i,t-n}\}, \quad n\epsilon\{1, 7\}. \quad (3)$$

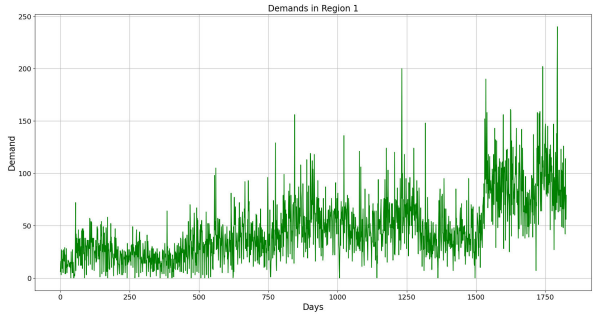
After production and transportations take place, the manufacturer obtains its next state as the leftover storage amounts. The retailers, on the other hand, obtain their next state as the leftover storage after receiving their orders and selling as much as the customers' demand or the amount they are holding:

$$a_t = \{p_t, a_{w_i,t}\}. \quad (4)$$

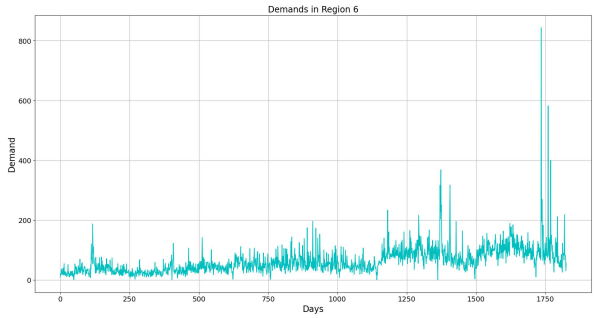
$$s_{F,t+1} = s_{F,t} + p_t - \sum_{i=1}^{R} a_{w_i,t}. \quad (5)$$

$$s_{w_i,t+1} = s_{w_i,t} + a_{w_i,t} - min\{s_{w_i,t} + a_{w_i,t}, d_{i,t}\}. \quad (6)$$

These transitions to new states deliver immediate rewards as $r_t$. The cycle continues throughout the timesteps at each episode, leading to total expected returns that the agent




(a) Region 1's demand plot




(b) Region 6's demand plot

**FIGURE 2. Sample regional demand plots.**

receives to update its parameters and make its policy optimal. Due to the high volatility in the demands, it is critical for the agent to vastly explore the action space. Demands from two regions are illustrated in figure 2. The agent needs to learn to adapt to a wide variety of scenarios given by the dynamic environment as the one in the current study. SAC is the model which particularly encourages extensive exploration of actions for the agent prior to finding the optimal policy.

At each timestep, the agent receives different reward value depending on how well the agent minimizes over-storage and stockouts. Firstly, the agent is penalized if each warehouse's opening storage levels, which are stock levels after the manufacturer's warehouse receives produced amounts and retailers' warehouse receive transported amounts, exceed the capacities:

$$c_t \leftarrow c_t - 1 \text{ if } Cap_F > s_{F,t} + p_t \tag{7}$$

$$c_t \leftarrow c_t - 1 \text{ if } Cap_{w_i} > s_{w_i,t} + a_{w_i,t} \tag{8}$$

In case the capacities are not exceeded, the opening storage levels in the retailers' warehouses are taken and compared to the demands at current timestep. The operation of storage in the retailers follows the opening storage method in order to prevent scenario of uniform sales since the clients' visits can occur at different hours during the day. For every opening storage level that is less than the demands, stockout costs are incurred as additional penalty for failing to meet partial demands:

$$c_t \leftarrow c_t - 1 \text{ if } d_{i,t} > s_{w_i,t} + a_{w_i,t} \tag{9}$$

After this comparison, closing storage amounts are obtained by calculating the leftovers in the warehouses. In case of the manufacturer, the leftovers are the storage amounts after total emission takes place and every retailer obtains leftover after it makes sales to the customers:

Manufacturer closing storage: $S_{F,t+1}$

Retailer closing storage: $s_{w_i,t+1}$

$$\text{Manufacturer storage cost} = \frac{(s_{F,t} + p_F) + s_{F,t+1}}{2 \times Cap_F}. \tag{10}$$

$$\text{Retailer storage cost} = \frac{(s_{w_i,t} + a_{w_i,t}) + s_{w_i,t+1}}{2 \times Cap_{w_i,t}}. \tag{11}$$

$$r_t = \text{storage cost} + \text{stockout cost} \tag{12}$$

Figure (3) illustrates the present model in diagrammatic form.

## IV. SOLUTION METHODOLOGY

The solution is benchmarked using multiple popular DRL algorithms, mainly SAC. Orders are placed for the agent to carry out production and transportation to ultimately fill the inventories in the retailers. The apparels are replenished and sold with short lead time, with initially stored products being sold to meet demands. In case there are insufficient stocks to meet demands, there are no backorders. Instead, the unfulfilled demands are considered to be lost sales. Otherwise, the inventory leftovers and opening storages are observed to see whether there are stocks that are remaining and must be handled. These inconvenient scenarios are the ones that the agent attempts to prevent, and it is represented by SAC. During its learning process, SAC follows the soft Markov Decision Process (soft-MDP), a different version of MDP that regular RL models follow.

### A. BASELINE POLICY AND ACTOR CRITIC ALGORITHMS

S-Policy is the baseline traditional inventory management policy in this study. In this policy, the inventory level is periodically observed and if this level drops below the reorder point, a reorder amount Q is made to fill the stocks. The production and transportation amounts are optimized with Bayesian Optimization [19], a widely known global optimizer of black-box functions.

Other baseline models are various "state-of-the-art" (SOTA) actor critic models, including "Advantage Actor Critic" (A2C) [20], Trust Region Policy Optimization (TRPO) [21], and Twin Delayed Deep Deterministic Policy Gradient (TD3) [22]. TD3 is the extended version of "Deep Deterministic Policy Gradient" (DDPG) [23] model which, even though it maintains its deterministic action selection, learns to improve the expected returns from the policy by using the Double Q-Learning trick [22], which will be further discussed later on. As a sanity check, the uniformly random policy, which only takes actions randomly, was experimented with as well.

### B. SOFT ACTOR CRITIC MODELING

SAC, like other SOTA DRL models, is an actor-critic model that includes Policy Network ($\phi$) and Critic Network ($\theta$),

each structured as Multilayer Perceptron (MLP) having two hidden layers. Each of these layers holds 256 hidden nodes. The soft-MDP process allows the agent to learn effectively by considering the entropy of the policy to be maximized along with the episodic expected returns, as shown in equation (13), where $H(\pi(a_t|s_t))$ is the policy entropy. Such inclusion of the entropy encourages extensive exploration and robustness to data. By establishing a tradeoff between rewards and policy entropy, the algorithm learns to optimize its policy on an offline basis by using previous interaction experiences. Doing so allows the algorithm to generalize across various experiences and not overfit to new ones and hence, becoming robust to new scenarios. Its successful performance in various Mujoco environments already reinforce its superior functionality in various tasks [6].

$$J(\pi) = \sum_{t=0}^{T} E_{(s_t,a_t)\sim p_\pi}[R(s_t, a_t) + aH(\pi(a_t|s_t))]. \quad (13)$$

The policy network determines the policy to iterate and optimize whereas the critic network determines the state-action values $Q_\theta(s_t, a_t)$, which evaluates the policy of the agent by approximating the total expected returns obtained by taking an action $a_t$ in state $s_t$. In case of the Critic Network, there are two current networks ($\theta$) and two target networks ($\theta^-$). Outputs from each hidden layer is activated with the ''ReLU'' function. Every network is parameterized with different weights. In the training phase, the networks' parameters are updated with mini-batch gradient descent using sampled mini-batches of experience tuples $(s_t, a_t, r_t, s_{t+1})$ from the experience replay $D$. The hyperparameters of the neural networks and training setup are shown in table 2. Such parameters are updated with the aim to minimize the policy and critic loss values. The Critic loss function is defined in equation (14) as the total mean squared error (MSE) between each current-Q, $Q_\theta(s_t, a_t)$ and $y_t$ (15). $y_t$ is the future expected total rewards. Clipped Double Q-learning trick from TD3 is applied. This technique enforces the usage of the lower of the two target Q-values, $Q_{\theta_i^-}(s_{t+1}, a_{t+1})$, each given by separate target networks, to form $y_t$. The purpose of using such trick is to prevent Q-value overestimation and minimize the effect of positive bias when improving the policy, stabilizing the learning process.

$$L(\theta) = \frac{1}{|B|} \sum_{(s_t,a_t,r_t,s_{t+1})\epsilon B} (Q_{\theta_i}(s_t, a_t) - y_t)^2. \quad (14)$$

$$y_t = R_t + \gamma \begin{pmatrix} \min Q_{\theta_i^-}(s_{t+1}, a_{t+1}(s_{t+1}; \phi)) \\ -\alpha log \pi_\phi(a_{t+1}|s_{t+1}) \end{pmatrix}, \quad i=1,2 \quad (15)$$

Another crucial aspect of the parameter update in the Critic Networks is that, the target Critic Network's parameters must be updated in a stable manner. Hence, a soft update, in which the target networks are updated towards current networks, is made: $\theta_i^- \leftarrow \tau\theta_i + (1-\tau)\theta_i^-$, where $\tau$ is the parameter value that defines the rate at which the soft update is made, $\tau\epsilon(0, 1)$. Meanwhile, the policy network outputs the same quantity of mean $\mu_i$ and standard deviation $\sigma_i$ as the number of actions. This policy is modeled as Gaussian distribution which, using $\mu_n$ and $\sigma_n$, ultimately sample the actions. The actions are sampled with the objective of ensuring that the action distribution is proportional to the soft Q-function, curtailing the KL-divergence between this and $\pi_\phi(.|s_t)$ as illustrated in equation (16). This equation can be rewritten as equation (17). Both the critic and policy network parameters are optimized with Adam optimizer [24].

$$E_{s_t\sim B}[D_{KL}(\pi_\phi(.|s_t)||\frac{exp(Q_\theta(s_{t,.}))}{Z_\theta(s_t)})]. \quad (16)$$

$$L(\phi) = \frac{1}{|B|} \sum_{s\epsilon B} (Q_{\theta_i}(s_t, a_t) - \alpha log \pi_\phi(a_t|s_t)). \quad (17)$$

The update of the networks' parameters is made at a rate determined by the learning rate $\alpha\epsilon(0, 1)$.

**TABLE 2. Hyperparameters table.**

| Hyperparameter | Explanation |
|---|---|
| Number of hidden layers | 2 |
| Number of hidden nodes | 256 |
| learning rate | 0.001 |
| Discount factor ($\gamma$) | 0.5 |
| Buffer size | $100k$ |
| $\tau$ | 0.005 |
| Batch size | 32 |

## V. NUMERICAL STUDY (EXPERIMENTS)

For the numerical study, the interaction of each baseline model and SAC in the inventory management environment was conducted. After the agent performs the actions, changes and leftovers in the manufacturer and each retailer's warehouse stock levels are observed and compared to each period's demands to record the KPI values. The demand data during the training phase belong to the sales taking place during four-year span, ranging from January $1^{st}$, 2015 to December $31^{st}$, 2018. The models are each trained for 1000 episodes, each episode having length of one year. Table (2) shows the hyperparameter sets used to train SAC. All the DRL models were implemented in Pytorch 1.10.0 and ran with GTX Ti-1080 GPU.

### A. DATASET

The dataset used for the demand is a real-case data. Specifically, it is a time-series sales data of Italian garments sold in Korea. Both the name of the company and process through which it was obtained are not disclosed due to confidentiality.

### B. KEY PERFORMANCE INDICATORS

Stable convergence of the episodic rewards towards, higher or lower values depending on the task, generally indicates competitive performance of the DRL models. On the other hand, due to the black-box nature of deep learning models, such convergence cannot explicitly validate the quality of

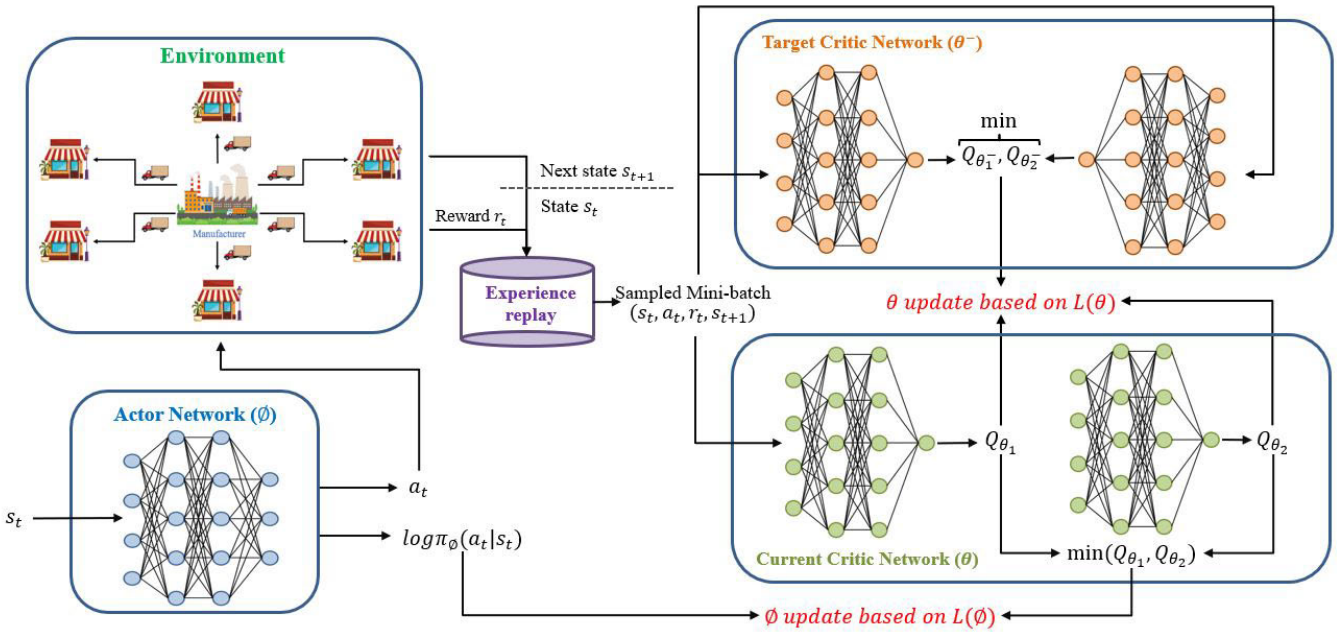



**FIGURE 3.** Framework of the apparel SAC-SCM method.

performance of the actions throughout the episodes. In other words, explainability of the policy is limited. Therefore, essential KPIs of inventory management were considered for each of the model's performance evaluation.

- Sell-through rate: Sell-through rate measures the amount of inventory that is sold within a given period relative to the inventory receival amount within such period

$$\frac{(\sum_{t=1}^{N}\sum_{i=1}^{6} min((s_{w_{i,t}} + a_{i,t}), d_{i,t}))}{\sum_{t=1}^{N}\sum_{i=1}^{6}(s_{w_{i,t}} + a_{i,t})} * 100$$

- Service level: Service levels measures the firm's ability to meet customer demands.

$$\frac{(\sum_{t=1}^{N}\sum_{i=1}^{6} min((s_{w_{i,t}} + a_{i,t}), d_{i,t}))}{\sum_{t=1}^{N}\sum_{i=1}^{6}(d_{i,t})} * 100.$$

- Inventory to Sales ratio: This KPI measures the firm's efficiency in clearing its inventories when it is meeting demands
  Beginning inventory: $(s_{w_{i,t_1}} + a_{i,t_1})$
  Ending inventory: $\max(s_{w_{i,t_N}} + a_{i,t_N}, d_{i,t_N})$
  Average inventory: $\frac{(s_{w_{i,t_1}} + a_{i,t_1}) + max(s_{w_{i,t_N}} + a_{i,t_N}, d_{i,t_N})}{2}$
  Net sales: $\sum_{n}^{N}\sum_{i}^{W}[min(d_{w_{i,t_n}}, s_{w_{i,t_n}} + a_{i,t_n})]$
  Inventory to sales ratio: $\frac{Average\ inventory}{Net\ sales}$

## VI. EXPERIMENTAL RESULTS

### A. COMPARISON BETWEEN SAC AND OTHER ALGORITHMS

For experimental results, the convergence of average training rewards and KPI values were mainly considered to evaluate

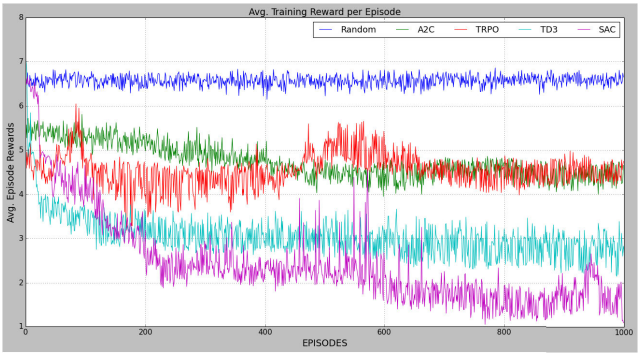
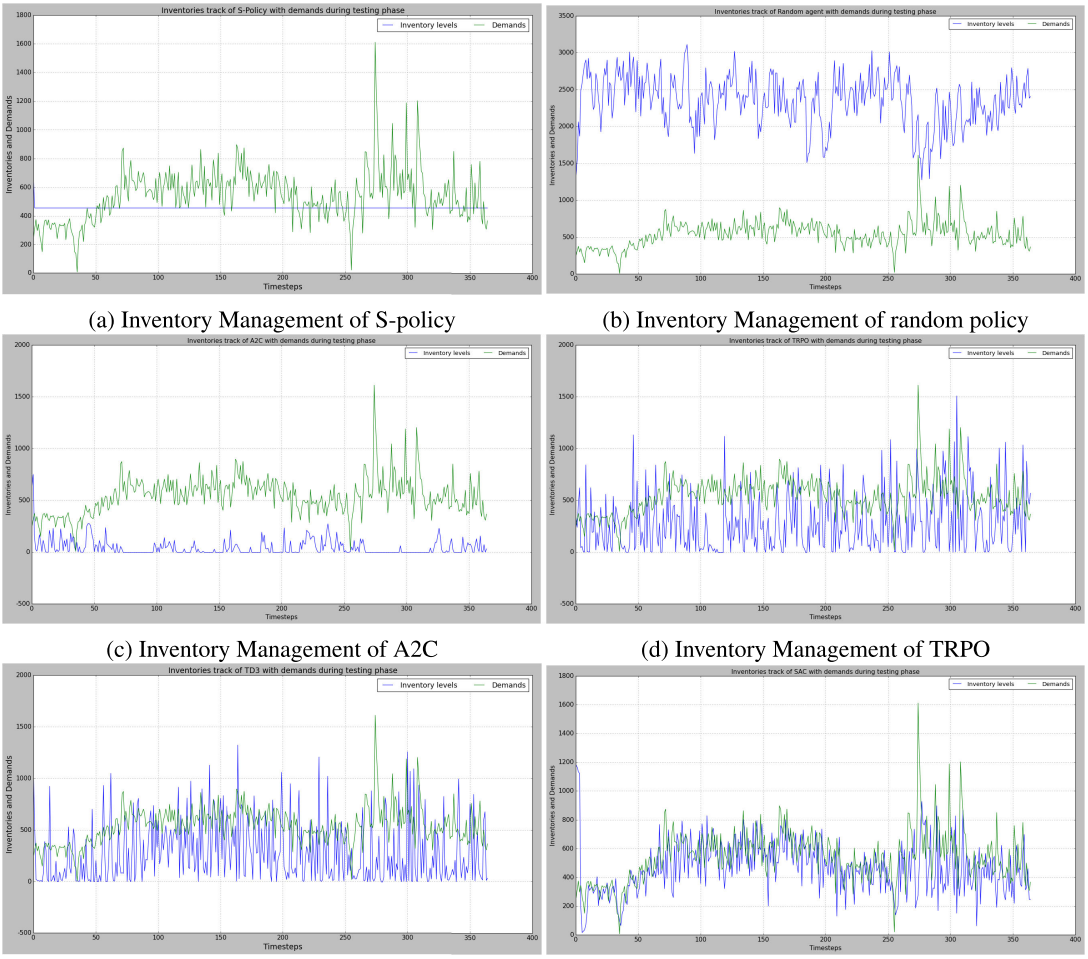



**FIGURE 4.** Training reward (penalties) of DRL models.

the performance of every algorithm. In the case of this study, downward convergence of training rewards towards zero is ideal since the objective is to minimize total cost. Figure (4) shows that the average training rewards of SAC converges at a lower value in comparison to all other models. The values for the S-policy were not included because this method selects the optimal transportation strategy to maintain constant storage levels during the training and testing phases. The interpretation behind the disparities in the convergence of the models will be discussed next. Figure (5) illustrates how properly each model is ensuring that the total inventory levels in the retailers are keeping up with the demands during the testing phase. The demands in this case are the total daily demand values during 2019. Out of all the models, SAC is conducting a balanced storage overall. In fact, it is conducting the distribution amounts much more appropriately, fulfilling




(a) Inventory Management of S-policy

(b) Inventory Management of random policy

(c) Inventory Management of A2C

(d) Inventory Management of TRPO

(e) Inventory Management of TD3

(f) Inventory Management of SAC

**FIGURE 5.** **Inventory management of each model during testing phase.**

most of the periodic demands despite the abrupt changes in the demand flow. A2C is making sales, but it has continuously missed out on the vast majority of the demands with the ongoing time span. TRPO shows better signs of attempting to follow and meet the demands. However, it still failed to fulfill a significant portion of the demand, particularly when considering larger portion of the time span. TD3 outperformed TRPO in sell-through rate and service level. However, it did not achieve optimal values and also obtained inferior inventory-to-sales ratio than S-Policy, TRPO, and SAC. SAC, on the other hand, leverages on its sample efficiency and action exploration to generalize across the demand patterns and perform proper allocation. In fact, SAC is ensuring that the retailers' inventory stock levels are keeping up with the demand much more intelligently overall. Although its sell-through rates throughout testing phase is lower than those of A2C and service levels are lower than those of random policy, SAC did not sacrifice one KPI for the other. Entropy maximization allowed SAC to not overfit to the training data and fit to the test data pattern much more effectively. Table 3 demonstrates the KPI values for every model involved in the experiment.

### *B. RESULTS AND DISCUSSION*

Although imperfect, SAC outperformed other models in several aspects, demonstrating its relevance and adequacy in inventory management tasks. First of all, although it achieved a lower sell-through rate than A2C, it ensured more customer demands are fulfilled by achieving a much higher service level, outperforming the one of A2C by 971.7%. Random policy achieved higher service level than SAC. On the other hand, SAC avoided overstocking that could incur troubling inventory management exertions. It did so by achieving 272.7% higher sell-through rate. Additionally, SAC achieved 7% lower inventory-to-sales ratio than S-policy, doing a better job ensuring that the total amount of sales is on par with the inventory amounts present in the retailers. Overall, A2C and random policy are the sole baseline models that outperformed SAC in either sell-through rate or service level. This indicates both model's failure in learning from the overall reward

**TABLE 3. KPIs comparison table.**

| model | Time span | Sell-through rate | Service level | Inventory to sales ratio |
|---|---|---|---|---|
| S-Policy | 8 weeks | 54.98 | 59.32 | 0.0043 |
| | 13 weeks | 61.34 | 53.85 | 0.0043 |
| | 26 weeks | 59.96 | 52.63 | 0.0043 |
| | 52 weeks | 61.09 | 51.67 | 0.0043 |
| Random | 8 weeks | 12.83 | 99.84 | 0.099 |
| | 13 weeks | 16.56 | 97.74 | 0.048 |
| | 26 weeks | 20.8 | 97.14 | 0.017 |
| | 52 weeks | 21.74 | 94.57 | 0.009 |
| A2C | 8 weeks | 76.33 | 23.91 | 0.072 |
| | 13 weeks | 78.78 | 13.41 | 0.061 |
| | 26 weeks | 82.93 | 7.43 | 0.045 |
| | 52 weeks | 87.64 | 6.87 | 0.023 |
| TRPO | 8 weeks | 45.16 | 31.03 | 0.030 |
| | 13 weeks | 49.82 | 28.42 | 0.019 |
| | 26 weeks | 56.25 | 25.69 | 0.007 |
| | 52 weeks | 53.54 | 27.22 | 0.007 |
| TD3 | 8 weeks | 60.44 | 30.71 | 0.091 |
| | 13 weeks | 65.8 | 31.2 | 0.043 |
| | 26 weeks | 72.36 | 42.56 | 0.016 |
| | 52 weeks | 69.79 | 36.18 | 0.0072 |
| SAC | 8 weeks | 72.18 | 75.66 | 0.043 |
| | 13 weeks | 77.33 | 73.08 | 0.024 |
| | 26 weeks | 80.58 | 75.37 | 0.008 |
| | 52 weeks | 81.04 | 73.63 | 0.004 |

function, prioritizing on some components. SAC, on the other hand, was able to consider every cost involved to achieve a more profitable inventory management.

## VII. CONCLUSION AND FUTURE WORKS

SC researchers had so far applied MDP-based DRL models to solve inventory management problems. In a recent study by Chen *et al.* [15], production and allocation decisions were optimized by using Q-learning and DQN. Demand value approximator was a sinusoidal function. In this study, a real-case time series dataset for the demands were incorporated for a soft-MDP based inventory management. The numerical results in this study show that DRL can be a better tool for prediction of product demand to conduct a relatively large SCM system. Furthermore, explicit visualization of the variations in the stock levels in comparison to demand flow was provided, enhancing the reliability behind the results of the experiments. Experimental results demonstrated the importance of entropy optimization, which allowed the agent to not suffer from the drastic fluctuations of the demand history throughout the episodes. Figure (3), in particular, demonstrated that SAC fulfilled higher portion of the customer demands as compared to other DRL models while maintaining appropriate storage amounts during the process. From this study, the followings findings are illustrated:

1) SAC achieved the best service level and competitive performance in other KPIs with extensive exploration to stay in track of all the possible oscillations that the demand values may bring to the agent. Results in Table (3) show higher service level achieved by SAC with the value near to 100 for each span during the test phase.
2) The results in figure (5) reflect a more reliable illustration of the performance of DRL than those conducted by Chen *et al.* [15]. The study by such authors only demonstrate the flow of the stock levels whereas present study demonstrates such flow as well as the demand flow. Hence, this study demonstrates whether the models indeed perform the appropriate inventory management.

The problem in this study can be extended to supply chain with ML and soft computing (SC) as proposed by Arora and Majumdar [25]. In particular, no study has been undertaken by merging RL and SC in inventory management to the best of one's knowledge. In another view, the present study can also be extended to fresh food waste reduction. Environmental sustainability can be largely enhanced with such reduction as suggested by Miguéis *et al.* [26]. The extension can also branch out to minimization of refrigerated fruit waste based on the temperature conditions and biochemical degradation of fruits during storage as proposed by Defraeye *et al.* [27].

## REFERENCES

[1] Z. Peng, Y. Zhang, Y. Feng, T. Zhang, Z. Wu, and H. Su, "Deep reinforcement learning approach for capacitated supply chain optimization under demand uncertainty," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 3512–3517, doi: 10.1109/CAC48633.2019.8997498.
[2] Z. Chen and R. Rossi, "A dynamic ordering policy for a stochastic inventory problem with cash constraints," *Omega*, vol. 102, Jul. 2021, Art. no. 102378, doi: 10.1016/J.OMEGA.2020.102378.
[3] K. van Donselaar, R. Broekmeulen, and T. de Kok, "Heuristics for setting reorder levels in periodic review inventory systems with an aggregate service constraint," *Int. J. Prod. Econ.*, vol. 237, Jul. 2021, Art. no. 108137, doi: 10.1016/J.IJPE.2021.108137.
[4] N. Vanvuchelen, J. Gijsbrechts, and R. Boute, "Use of proximal policy optimization for the joint replenishment problem," *Comput. Ind.*, vol. 119, Aug. 2020, Art. no. 103239, doi: 10.1016/J.COMPIND.2020.103239.
[5] R. N. Boute, J. Gijsbrechts, W. van Jaarsveld, and N. Vanvuchelen, "Deep reinforcement learning for inventory control: A roadmap," *Eur. J. Oper. Res.*, vol. 298, no. 2, pp. 401–412, Apr. 2022, doi: 10.1016/J.EJOR.2021.07.016.
[6] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018, *arXiv:1801.01290*.
[7] F. Stranieri and F. Stella, "A deep reinforcement learning approach to supply chain inventory management," 2022, *arXiv:2204.09603*.
[8] H. Qian, H. Guo, B. Sun, and Y. Wang, "Integrated inventory and transportation management with stochastic demands: A scenario-based economic model predictive control approach," *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117156, doi: 10.1016/J.ESWA.2022.117156.
[9] M. Liu, Z. Liu, F. Chu, A. Dolgui, C. Chu, and F. Zheng, "An optimization approach for multi-echelon supply chain viability with disruption risk minimization," *Omega*, vol. 112, Oct. 2022, Art. no. 102683, doi: 10.1016/J.OMEGA.2022.102683.
[10] M. Xiang, R. Rossi, B. Martin-Barragan, and S. A. Tarim, "A mathematical programming-based solution method for the nonstationary inventory problem under correlated demand," *Eur. J. Oper. Res.*, vol. 304, no. 2, pp. 515–524, Jan. 2023, doi: 10.1016/J.EJOR.2022.04.011.
[11] A. Oroojlooyjadid, M. Nazari, L. Snyder, and M. Takáč, "A deep Q-network for the beer game: A deep reinforcement learning algorithm to solve inventory optimization problems," 2017, *arXiv:1708.05924*.
[12] A. Kara and I. Dogan, "Reinforcement learning approaches for specifying ordering policies of perishable inventory systems," *Expert Syst. Appl.*, vol. 91, pp. 150–158, Jan. 2018.
[13] R. S. Sutton and A. G. Barto, "Reinforcement learning," in *An Introduction Complete Draft*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018, pp. 1–3. Accessed: Jul. 14, 2022. [Online]. Available: https://mitpress.mit.edu/books/reinforcement-learning-second-edition

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. K. Openai, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[15] H. Chen, Z. Chen, F. Lin, and P. Zhuang, "Effective management for blockchain-based agri-food supply chains using deep reinforcement learning," *IEEE Access*, vol. 9, pp. 36008–36018, 2021, doi: 10.1109/ACCESS.2021.3062410.
[16] H. Meisheri, N. N. Sultana, M. Baranwal, V. Baniwal, S. Nath, S. Verma, B. Ravindran, and H. Khadilkar, "Scalable multi-product inventory control with lead time constraints using reinforcement learning," *Neural Comput. Appl.*, vol. 34, no. 3, pp. 1735–1757, Feb. 2022, doi: 10.1007/s00521-021-06129-w.
[17] Q. Zhou, Y. Yang, and S. Fu, "Deep reinforcement learning approach for solving joint pricing and inventory problem with reference price effects," *Expert Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116564, doi: 10.1016/j.eswa.2022.116564.
[18] L. A. San-José, J. Sicilia, M. González-De-la-Rosa, and J. Febles-Acosta, "Analysis of an inventory system with discrete scheduling period, time-dependent demand and backlogged shortages," *Comput. Oper. Res.*, vol. 109, pp. 200–208, Sep. 2019, doi: 10.1016/J.COR.2019.05.003.
[19] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," 2012, *arXiv:1206.2944*.
[20] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," 2016, *arXiv:1602.01783*.
[21] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 3, Feb. 2015, pp. 1889–1897, doi: 10.48550/arxiv.1502.05477.
[22] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018, *arXiv:1802.09477*.
[23] T. P. Lillicrap, "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, Sep. 2015, pp. 1–5, doi: 10.48550/arxiv.1509.02971.
[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[25] S. Arora and A. Majumdar, "Machine learning and soft computing applications in textile and clothing supply chain: Bibliometric and network analyses to delineate future research agenda," *Expert Syst. Appl.*, vol. 200, Aug. 2022, Art. no. 117000, doi: 10.1016/J.ESWA.2022.117000.
[26] V. L. Miguéis, A. Pereira, J. Pereira, and G. Figueira, "Reducing fresh fish waste while ensuring availability: Demand forecast using censored data and machine learning," *J. Cleaner Prod.*, vol. 359, Jul. 2022, Art. no. 131852, doi: 10.1016/J.JCLEPRO.2022.131852.
[27] T. Defraeye, G. Tagliavini, W. Wu, K. Prawiranto, S. Schudel, M. Assefa Kerisima, P. Verboven, and A. Bühlmann, "Digital twins probe into food cooling and biochemical quality changes for reducing losses in refrigerated supply chains," *Resour., Conservation Recycling*, vol. 149, pp. 778–794, Oct. 2019, doi: 10.1016/J.RESCONREC.2019.06.002.

**JI WON CHONG** received the B.S. degree in agricultural and consumer economics from the University of Illinois, Urbana-Champaign, in 2013, and the M.S. degree in international management from the ESADE Business School, Barcelona, Spain, in 2015. He is currently pursuing the Ph.D. degree in industrial engineering with Yonsei University, Seoul, South Korea. His main research interests include natural language processing, machine learning, and reinforcement learning.

**WOOJU KIM** received the Ph.D. degree in operations research from KAIST, South Korea, in 1994. He is currently a Professor with the School of Industrial Engineering, Yonsei University. His main research interests include natural language processing, reliable knowledge discovery, big data intelligence, machine learning, and artificial intelligence.

**JUNESEOK HONG** received the Ph.D. degree in management information systems from KAIST, South Korea, in 1997. He is currently a Professor with the Department of Management Information Systems, Kyonggi University. His previous works were published in *Information*, *Journal of Computer Networks and Communications*, *International Journal of Information Technology and Decision Making*, and *Expert Systems with Applications*. His research interests include intelligent decision making using big-data analysis, deep-learning technique, and intelligent agent systems.