Observable Multi-Task Cooperative Multi-Agent Reinforcement Learning for Inventory Management

Hongmai Xu[1], An Zeng[1]*, Qinghua Zhu[1], Yuzhu Ji[1], Baoyao Yang[1], Dan Pan[2], Jiayu Ye[1]
[1]Guangdong University of Technology, Guangzhou 510000, China
[2]Guangdong Polytechnic Normal University, Guangzhou 510000, China
2112305108@mail2.gdut.edu.cn, {zengan, zhugh, yuzhu.ji, ybaoyao}@gdut.edu.cn, pandan@gpnu.edu.cn,
yejiayu97@outlook.com

***Abstract*—Unreasonable Inventory Management (IM) strategies lead to extended delivery times and increased costs in the supply chain. Multi-Agent Reinforcement Learning (MARL) is proposed for the research area of IM. However, existing methods are limited to simple one-to-many supply chain models, which cannot match real-world supply chain. Additionally, in the current use of Cooperative Multi-agent Reinforcement Learning (CMARL) in supply chain model, agents communicate through shared rewards, resulting in insufficient communication efficiency. To address these issues, this paper proposes a many-to-many supply chain model and Observable Multi-task CMARL which leverages a learning paradigm of Centralized Training and Decentralized Execution (CTDE) Based on Shared Rewards and Multi-task CMARL. The learning paradigm that we propose applies shared rewards to each centralized critic, enhancing the communication and collaboration abilities among agents. Furthermore, we propose Multi-task CMARL which improves the performance of CMARL by sharing parameters between tasks. To verify the effectiveness of the algorithm, extensive experiments were conducted on both real-world and simulated datasets. The experimental results show that the Observable Multi-task CMARL algorithm outperforms traditional heuristic algorithms and CMARL in the many-to-many supply chain model, maintaining low inventory levels while improving product sales to an average increase of 131.2%.**

***Index Terms*—Multi-Agent Reinforcement Learning, Multi-Task Reinforcement Learning, Inventory Management, Supply Chain Model.**

I. Introduction

Inventory management, a critical component of supply chain management, involves tracking and controlling materials, products, and other resources to ensure supply chain efficiency. The primary requirement of inventory management is to ensure that products are available at the right time, in the right place, and in the right quantity to meet customer demand. This must be achieved while maximizing operational efficiency and minimizing costs [1] . The results of [2] indicate that the higher the inventory level retained by a company, the lower its return rate. Therefore, future studies should focus on the using of information technology to accelerate and smooth the flow of products in the supply chain [3].

The application of reinforcement learning (RL) has been proposed as an improved approach. Robert N. Boute et al. [4] demonstrates the efficiency of RL in large-scale inventory management. Studies by Oroojlooyjadid et al. [5], Francesco Stranieri et al. [6], and Stranieri et al. [7] prove the effectiveness of Deep Reinforcement Learning (DRL) in solving inventory management problems, particularly in handling dynamic demand patterns and complex decision-making scenarios. Additionally, Madhav Khirwar et al. [8] and Sultana, N.N. et al. [9] have applied MARL to inventory management, demonstrating that MARL offers stronger robustness and adaptability in inventory management compared to single-agent RL and traditional heuristic algorithms. The CMARL method proposed by [8] encourages each independent agent to collaborate through shared rewards and demonstrates superior performance and scalability compared to traditional algorithms in a one-to-many supply chain model. However, in CMARL, the one-to-many supply chain model cannot fully adapt to real-world supply chain scenarios. Additionally, the agents only communicate via shared rewards, failing to observe the states and actions in other agents. Furthermore, with the vertices as agents, the communication among agents becomes a challenge as the scale of the supply chain model increases.

To address these issues, this paper proposes a novel many-to-many supply chain model consisting of three echelons: the factory echelon, warehouse echelon, and store echelon. And, we propose an Observable Multi-task CMARL algorithm. This algorithm builds on CMARL and combines the centralized training and decentralized execution based on shared rewards learning paradigm with multi-task reinforcement learning (MTRL). First, the learning paradigm allows the agents to observe the states and actions of all other agents during training, thus enhancing information sharing among the agents. Additionally, the combination of CMARL and MTRL transforms the agent model from one agent per vertex to one agent per echelon, treating the vertices within each echelon as different tasks. This not only reduces the number of agents and eases the communication burden but also improves the performance of algorithm.

The main contribution of this paper are:

- This paper proposes a novel many-to-many supply chain model that is more aligned with real-world supply chain scenarios.
- A learning paradigm of centralized training and decentralized execution based on shared rewards is introduced, enabling agents to observe the states and actions of other agents and share rewards during training.

[1]*Corresponding author: An Zeng (Email: zengan@gdut.edu.cn).

- We design a novel Observable Mutli-task CMARL algorithm, which effectively improves the communication and performance of the CMARL algorithm in inventory management.
- The experimental results show that Observable Mutli-task CMARL outperforms CMARL in terms of reward performance. Its reasonable inventory management strategy leads to an average sales increase of 131.2%.

II. Methodology

We introduce a many-to-many supply chain model and an Observable Multi-task CMARL built upon CMARL, incorporating new methods: Centralized Training and Decentralized Execution (CTDE) Based on Shared Rewards and Multi-task CMARL.

A. Preliminary

We define a generalization of MDP to a multi-agent setting, represented by the tuple $< U, \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R}, \gamma >$. Here, $U = \{u|1 \leq u \leq |U|\}$ is the set of agents in the environment. $\mathbf{S}$ is the state spaces of all agents $u \in U$: $\mathbf{S} = s_1 \times s_2 \times \cdots \times s_{|U|}$, and $\mathbf{A}$ is the action spaces of all agents $u \in U$: $\mathbf{A} = a_1 \times a_2 \times \cdots \times a_{|U|}$. $\mathbf{T}$ denotes the state transfer function from $\mathbf{S}$ to $\mathbf{S}'$. $\mathbf{R}$ represents each agent's reward function: $\mathbf{R} = r_1 \times r_2 \times \cdots \times r_{|U|}$, and γ is a scalar discount factor. At each time step t, all agents take actions $\mathbf{a} \in \mathbf{A}$. The objective of each agent in the environment is to maximize its long-term reward by determining its optimal policy $\pi_u : s_u \rightarrow a_u$.

B. Many-to-many Supply Chain Model

We design a many-to-many supply chain model based on the works in [8] and [10]. It consists of three echelons: factories, warehouses, and stores. In the many-to-many supply chain model, the environment update is divided in two stages. The first stage is the replenishment phase, as depicted in Figure 1a. The second stage is the allocation phase, as shown in Figure 1b.

1) Multi-task CMARL Implementation in Supply Chain Model: In our supply chain environment, T=$< fa, wh, st >$ is the tuple of agents that consisted of the factory agent (fa), the warehouse agent (wh), the store agent (st) and $\mathbf{V}_{st}$, $\mathbf{V}_{wh}$, $\mathbf{V}_{fa}$ are the sets of vertices of the store, the warehouse and the factory, respectively.

For a store agent, we can express the state spaces and replenishment actions spaces of the store agent in the following equation:

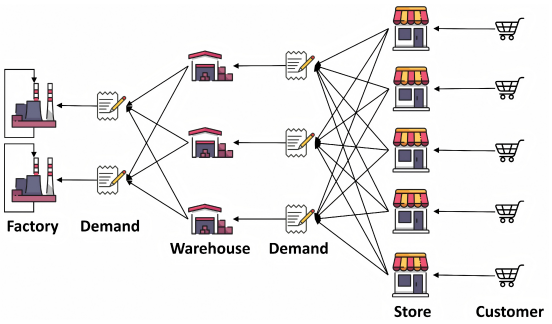
$$s_t = [o_{1,t}, \cdots, o_{K,t}] \tag{1}$$

$$o_{i,t} = [I_1(i,t), \cdots, I_{|\mathbf{V}_{st}|}(i,t), d_v(i,t), d_v(i,t-1), t] \tag{2}$$

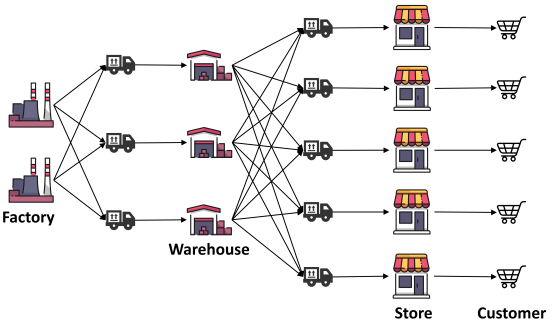
$$X_{r,t} = [R_1(s_t), \cdots, R_{|\mathbf{V}_{st}|}(s_t)] \tag{3}$$

$$R_v(s_t) = [p_{1,1}, \cdots, p_{K,|\mathbf{V}_{wh}|}] \tag{4}$$

where t is the time step, s_t and $X_{r,t}$ are the total state and total actions of store agent, $o_{i,t}$ is the state of the store agent with respect to product i, $I_v(i,t)$ is the inventory level and



(a) Example of supply chain topology in replenishment.



(b) Example of supply chain topology in allocation.

Fig. 1. Example of many-to-many supply chain model.

$R_v(s_t)$ is the replenishment quantity for the store $v \in \mathbf{V}_{st}$, and $p_{i,v'}$ is the quantity of replenishment policy of product i from store v to warehouse $v' \in \mathbf{V}_{wh}$, $d_v(i,t)$ is the demand for the store v.

For a warehouse agent, the state spaces and replenishment actions spaces of warehouse agent are the same as those of store agent (1) (2) (3) (4) and the warehouse $v \in \mathbf{V}_{wh}$, the factory $v' \in \mathbf{V}_{fa}$. Its distribution actions spaces are:

$$X_{d,t} = [D_1(s_t), \cdots, D_{|\mathbf{V}_{wh}|}(s_t)] \tag{5}$$

$$D_v(s_t) = [q_{1,1}, \cdots, q_{K,|\mathbf{V}_{st}|}] \tag{6}$$

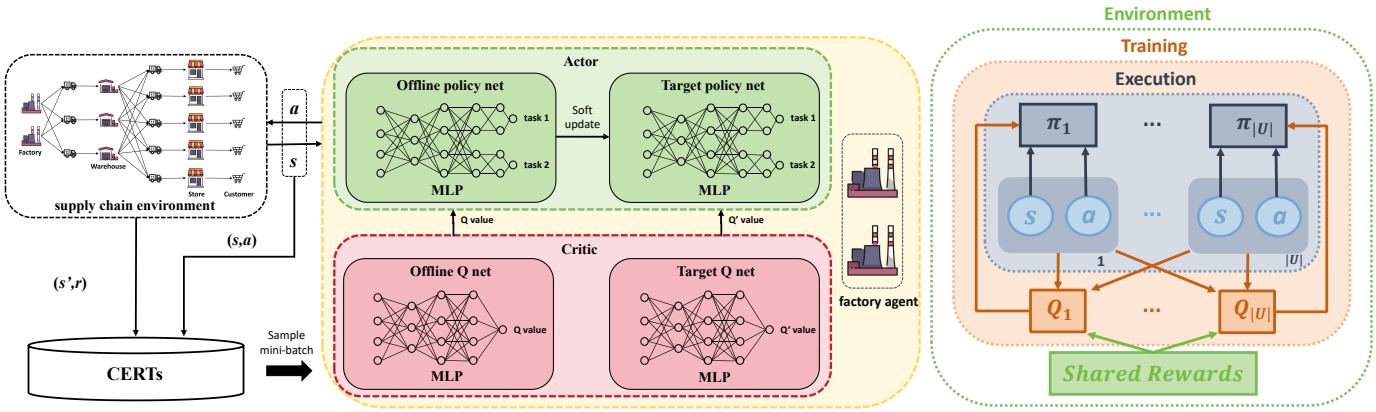
where $X_{d,t}$ is the total distribution actions of warehouse agent, $D_v(s_t)$ is the quantity of distribution policy for the warehouse $v \in \mathbf{V}_{wh}$ and the store $v' \in \mathbf{V}_{st}$.

A factory agent has the same state spaces and replenishment actions spaces as a store agent (1) (2) (3) (4) that the factory $v \in \mathbf{V}_{fa}$ and $v' \in \mathbf{V}_{fa}$, and the same distribution actions spaces as warehouse agent (5) (6) that the warehouse $v' \in \mathbf{V}_{wh}$.

2) Dynamic at Supply Chain: The state transitions in the environment are defined at the vertex level. The demand for store $v \in \mathbf{V}_{st}$ is generated by (15) or real-world datasets.

The inventory level of the store v at time step $t+1$ is described by the following equation:

$$I_v(i,t+1) = I_v(i,t) - \min\{d_v(i,t), I_v(i,t)\} + \sum_{i=0}^{K} \sum_{v \in \mathbf{V}_{wh}} q_{i,v,v'} \tag{7}$$



(a) An overview of network architecture of Multi-task CMARL.

(b) An overview of Centralized Training and Decentralized Execution Based on Shared Reward.

Fig. 2. An overview of Observable Multi-task CMARL.

where $q_{i,v,v'}$ is the distribution quantity of product i from warehouse v to store v'. The inventory level of the warehouse v at time step $t+1$ is expressed by the following equation:

$$I_v(i,t+1) = I_v(i,t) - D_v(s_t) + \sum_{i=0}^{K} \sum_{v' \in \mathrm{V}_{fa}} q_{i,v',v}, \quad (8)$$

where $q_{i,v',v}$ is the distribution quantity of product i from factory v' to warehouse v. The inventory level of the factory v at $t+1$ time step is as follows:

$$I_v(i,t+1) = I_v(i,t) - D_v(s_t) + R_v(s_t) \quad (9)$$

3) Reward Function: In [8], a shared reward structure was proposed to address the issue of competition between individual agents. We adopt this reward structure in our experiments. The following are the components of our reward structure:

$$R(t) = R_r(t) + R_h(t) + R_t(t) + R_o(t) + R_u(t), \quad (10)$$

where $R_r(t)$ is sales revenue, $R_h(t)$ is inventory holding cost, $R_t(t)$ is transportation cost, $R_o(t)$ is overfulfilled penalty and $R_u(t)$ is unfulfilled penalty.

C. Observable Multi-task Cooperative Multi-agent Reinforcement Learning

We propose a novel CMARL algorithm, Observable Multi-task Cooperative Multi-agent Reinforcement Learning, which combines two methods: Centralized Training and Decentralized Execution Based on Shared Rewards and Multi-task Reinforcement Learning.

1) Centralized Training and Decentralized Execution Based on Shared Rewards: In the supply chain model, information sharing between upstream and downstream plays a key role in the transmission across the entire supply chain.

In the paper [8], it is mentioned that no agents can observe each other's state space, indicating that is a paradigm of independent learning [11] in MARL. In the independent learning paradigm with $|U|$ agents, each agent $u \in U$ observes only its own state space during training and makes corresponding action decisions. The paper [12] proposes the multi-agent learning paradigm of Centralized Training and Decentralized Execution (CTDE) and the experimental results outperform other traditional RL algorithms. Inspired by the [12], we combined the CTDE learning paradigm with shared rewards to develop a new multi-agent learning paradigm, Centralized Training and Decentralized Execution Based on Shared Reward. In the CTDE based on shared rewards learning paradigm, the agents are mutually observable. The critic policy Q_u of agent u receives the states and actions of all agents, as well as a sum of rewards given the environment for all agents during. Then the critic network uses this information to produce an evaluation output to guide the decision-making of the actor network π_u. The overview of CTDE based on shared rewards learning paradigm is shown in Figure 2b. The policies of $|U|$ agents are parameterized by $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_{|U|}\}$. The the critic policy Q_u and the actor policy π_u are updated as:

$$L(\theta_u^Q) = \frac{1}{N} \sum_{i=1}^{N} ((Q_u(\mathbf{S}, \mathbf{A}) - y))^2 \quad (11)$$

$$y = \sum_{i=1}^{|U|} r_i + \alpha Q_{u'}(\mathbf{S}', \mathbf{A}') \quad (12)$$

$$J(\theta_u^\pi) = \frac{1}{N} \sum_{i=1}^{N} (Q_u(\mathbf{S}, (a_1, a_2, \ldots, a_{|U|})|_{a_u = \pi_u(s_u)})) \quad (13)$$

where N is the batch size of samples randomly drawn from the experience pool, **S** and **A** are the state spaces and the action spaces of all agents $u \in U$, **S'** and **A'** are the state spaces and the action spaces of all agents $u \in U$ at time step $t+1$, Q'_u is the target policy with delayed parameter θ'_u.

In our experiments, the results demonstrate that the application of the CTDE based on shared rewards learning paradigm significantly improves the performance of CMARL.

2) Multi-task CMARL Network: In the supply chain, not only is information sharing between different levels important, but information sharing between vertices within the same level is equally crucial. The CMARL mentioned in paper [8] models each vertex as an agent, but this approach fails to fully leverage the information sharing between vertices.

To enhance information sharing between vertices, we apply this approach to our supply chain environment and construct an Multi-task CMARL model by considering the output of each vertex $v \in \{1 \leq v \leq |V|\}$ in every supply echelon as an independent task T_v, and each echelon of the supply chain representing an agent. Our approach bases on [13], which uses a shared backbone, branching action output network architecture. This design allows agents to collect more information from vertices and enhances the coupling between vertices through parameter sharing in MTRL. The network architecture of factory agent is shown in Figure 2a. Deep deterministic policy gradient (DDPG) [14] is applied to each agent in MT-CMARL. We set $\boldsymbol{\phi} = \{\phi_1, \phi_2, \ldots, \phi_{|V|}\}$ as the parameters for each task branch. The action output of the actor network is described as:

$$a_u = \pi_u(s_u) = \{\pi_u(s_u)|_{\theta=\phi_1}, \ldots, \pi_u(s_u)|_{\theta=\phi_{|V|}}\} \quad (14)$$

The update procedures for the centralized action-value function Q_u and actor policy π_u are identical to those in (11), (12) and (13). Concurrent experience replay trajectories (CETRs) [15] store all experience samples of each agent $<$ **S**, **A**, **S'**, **R** $>$ at the same time step.

Our experimental results demonstrate that combining MTRL with CMARL can effectively improve the performance of CMARL.

III. Experiments

In this section, we will explore and answer the following research questions through experiments:

- **RQ1:** Dose Observable Multi-task CMARL outperform CMARL on the supply chain scenario?
- **RQ2:** How does Observable Multi-task CMARL improve the performance of the algorithm?
- **RQ3:** Does CTDE Based on Shared Rewards and MTRL improve the performance of CMARL?

A. Datasets

In our experiments, we use real-world datasets and simulated datasets.

1) Real-world Datasets: The product demand in the real-world datasets is based on actual sales data from enterprises. Specifically, the datasets comprises product data from a mold enterprise cluster in the Yangtze River Delta region. Due to confidentiality restrictions, this datasets is not publicly accessible. For the purposes of training, we selected the monthly sales data of an iron sleeve mold over a four-year period, with each month corresponding to one time step in the many-to-many supply chain environment.

2) Simulation Datasets: To evaluate the performance of the agent, the demand is modeled as a cosine function with stochastic components to simulate a simple seasonal demand pattern, as defined by the following equation:

$$d_v(i,t) = max\{0, \frac{d_{max_i}}{2}\sin(\frac{2\pi(t+3i)}{T}) + \frac{d_{max_i}}{2} + N\} \quad (15)$$

where d_{max_i} is the maximum demand of product i, $N \in \{0, d_{var_i}\}$ is a random variable, $d_{var_i} = \alpha d_{max_i}$ represents the variations in demand, $P(N=0) = P(N=d_{var_i}) = 0.5$, $\alpha \in [-0.1, 0.1]$ is noise factor.

We designed five types of products that are popular during different periods within a year. At each time step t, the demand may vary for any product i, as is shown in Figure 3.

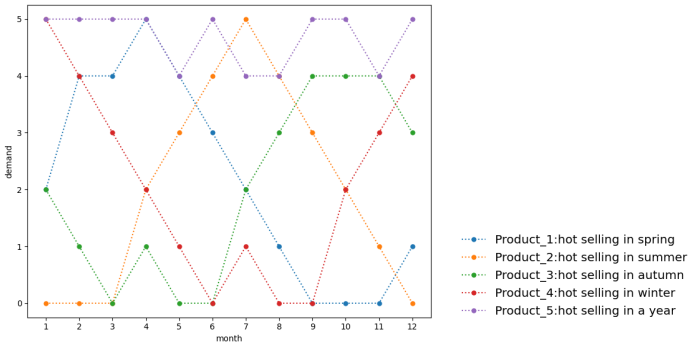


Fig. 3. Variations in demand for 5 products.

B. Experimental Settings

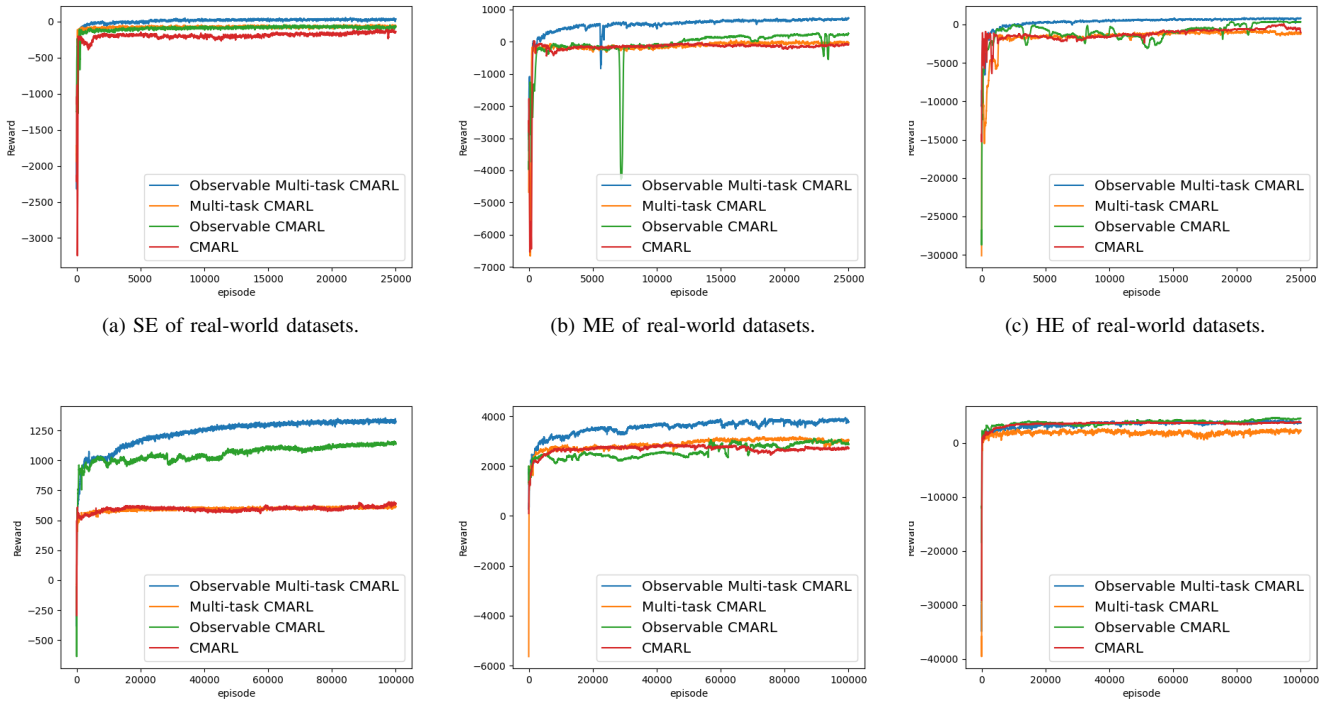
Our experiments were conducted on a GeForce RTX 3090 GPU, with the experimental environment consisting of Python 3.8.19, CUDA 11.2, and PyTorch 1.10.1. In our experimental setup, each agent is initialized with a specific inventory level and product demand at the beginning of each training episode and $\gamma = 0.95$. The experimental environment is divided into three levels: simple, middle, and hard. (1) Simple Environment (SE): It consists of 1 factory, 2 warehouses, and 2 stores. (2) Middle Environment (ME): It consists of 2 factories, 3 warehouses, and 5 stores. (3) Hard Environment (HE): It consists of 3 factories, 5 warehouses, and 10 stores.

We compare five algorithms in our experiment: The (σ, Q)-policy [16], CMARL [8], Observable CMARL, Multi-task CMARL and Observable Multi-task CMARL.

C. Experimental Results

We evaluate the performance of the algorithms by comparing the changes in rewards during the training phase and the rewards, inventory levels, and sales levels during the testing phase for each algorithm.

1) Training Results: The four RL algorithms introduce action noise during training to explore the possibilities brought by different actions. We collected 1.2 million timesteps in total for both the real-world and simulated datasets, segmenting them into 25k episodes (**t**=48) for the former and 100k



(a) SE of real-world datasets. (b) ME of real-world datasets. (c) HE of real-world datasets.

(d) SE of simulation datasets. (e) ME of simulation datasets. (f) HE of simulation datasets.

Fig. 4. Training results in real-world datasets and simulation datasets.

episodes (**t**=12) for the latter. We compared the algorithms based on the reward values from each episode, and the reward curves during training are shown in Figure 4.

As shown in the Figure 4, Observable Multi-task CMARL demonstrates the best training performance and the most stable training across three different environments in both real-world datasets and simulation datasets.

*2) Testing Results (**RQ1**):* In the test experiments, action noise is removed from the four RL algorithms, and demand data from 4800 time steps is used to evaluate the performance of five algorithms based on their rewards. The average reward during testing is shown in Table I.

Observable Multi-task CMARL achieves the highest rewards in three different environments across two datasets, demonstrating that combining the CTDE based on shared rewards learning paradigm with the multi-task reinforcement learning can effectively enhance the performance of the CMARL algorithm in supply chain scenarios.

The total sales quantity of all stores in the supply chain is shown in Table II.

Observable Multi-task CMARL achieves the highest total sales quantity across all environments expect in SE of the real-world datasets, while the (σ, Q)-policy achieves the highest total sales quantity in SE of the real-world datasets. However, as shown in Figure 5, Observable Multi-task CMARL maintains a lower inventory level to reduce inventory costs and the (σ, Q)-policy maintains a higher inventory level leading to a higher inventory cost. Therefore, as seen in Table I, Observable

TABLE I
AVERAGE REWARD. THE MAXIMUM VALUE FOR EACH TASK IS DISPLAYED IN BOLD.

Method	Average Reward					
	Real-World			Simulation		
	SE	ME	HE	SE	ME	HE
the (σ, Q) policy	-591	-1186	-2137	684	2167	2714
CMARL	333	119	-1067	673	2830	3448
Observable CMARL	387	355	313	1163	2631	4139
Multi-task CMARL	389	254	-125	628	3346	4011
Observable Multi-task CMARL	**878**	**912**	**1365**	**1349**	**4167**	**5234**

Multi-task CMARL achieves a higher reward than the (σ, Q)-policy in SE of the real-world datasets. As illustrated in Figure 5, Observable Multi-task CMARL demonstrates the ability to adjust its inventory levels promptly in response to fluctuations in demands and minimizes the discrepancy between the quantity of successfully sold products and demand, thereby optimizing profit. The experimental results indicate that Observable Multi-task CMARL exhibits stronger adaptability compared to CMARL.

*3) Ablation Study (**RQ3**):* Based on Table I, we investigate whether CTDE Based on Shared Rewards and MTRL can improve the performance of CMARL.

- **CTDE Based on Shared Rewards:** Table I reveals

TABLE II
SALES QUANTITY IN AN EPISODE. THE MAXIMUM VALUE FOR EACH TASK IS DISPLAYED IN BOLD. (**RQ2**)

Method	Sales Quantity					
	Real-World			Simulation		
	SE	ME	HE	SE	ME	HE
the (σ, Q) policy	**269**	464	815	311	822	1510
CMARL	99	263	1132	147	792	1354
Observable CMARL	100	544	1092	418	837	1400
Multi-task CMARL	104	466	1036	148	875	1779
Observable Multi-task CMARL	220	**985**	**1932**	**468**	**1215**	**2022**

that CTDE Based on Shared Rewards significantly improves the performance of both CMARL and Multi-task CMARL across three distinct environments in the real-world datasets. For example, CTDE Based on Shared Rewards enhances the performance of CMARL and Multi-task CMARL by 16.2% and 125.7% in SE. These results provide strong evidence that CTDE Based on Shared Rewards facilitates the agents' acquisition of more comprehensive global information.

- **Multi-task Reinforcement Learning:** Table I demonstrates that MTRL significantly enhances the performance of both CMARL and Observable CMARL across three environments within the real-world datasets by sharing parameters across tasks. Specifically, in the simple environment, MTRL improves the performance of CMARL and Observable CMARL by 19.5% and 126.8%.

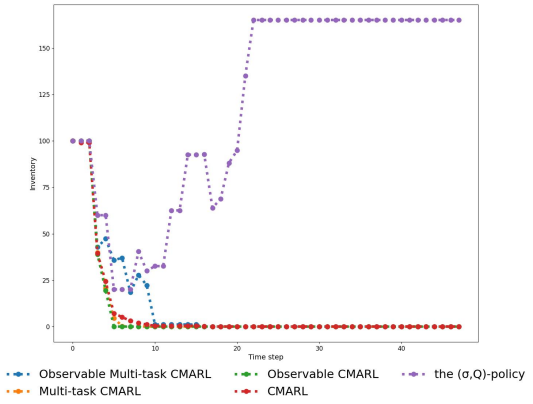


Fig. 5. The variation curve of the total inventory of all vertices in the supply chain within an episode in the SE of simulation datasets.

IV. CONCLUSION

In this paper, we study inventory management problems in supply chain scenarios. We design a many-to-many supply chain model and a Observable Multi-task CMARL algorithm. In Observable Multi-task CMARL, we combine CMARL with two innovative approaches: Centralized Training and Decentralized Execution Based on Shared Rewards and Multi-task Reinforcement Learning. We conduct extensive experimental studies on both real-world datasets and simulation datasets. The experimental results show that Observable Multi-task CMARL outperforms CMARL and traditional heuristic algorithms across three supply chain scenarios in both datasets. Notably, the results analysis indicates that Observable Multi-task CMARL significantly improves sales performance, demonstrating its ability to maintain lower inventory levels while satisfying order demand, thus maximizing sales quantity.

V. ACKNOWLEDGMENT

This work was supported by National Key Research and Development Program of China (Grant No. 2023YFB3308700).

REFERENCES

[1] J. T. Mentzer, W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith, and Z. G. Zacharia, "Defining supply chain management," *Journal of Business logistics*, vol. 22, no. 2, pp. 1–25, 2001.

[2] D. P. Koumanakos, "The effect of inventory management on firm performance," *International journal of productivity and performance management*, vol. 57, no. 5, pp. 355–369, 2008.

[3] G. P. Cachon and M. Fisher, "Supply chain inventory management and the value of shared information," *Management science*, vol. 46, no. 8, pp. 1032–1048, 2000.

[4] R. N. Boute, J. Gijsbrechts, W. van Jaarsveld, and N. Vanvuchelen, "Deep reinforcement learning for inventory control: A roadmap," *European Journal of Operational Research*, vol. 298, no. 2, pp. 401–412, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721006111

[5] A. Oroojlooyjadid, M. Nazari, L. V. Snyder, and M. Takáč, "A deep q-network for the beer game: Deep reinforcement learning for inventory optimization," *Manufacturing & Service Operations Management*, vol. 24, no. 1, pp. 285–304, 2022.

[6] F. Stranieri and F. Stella, "A deep reinforcement learning approach to supply chain inventory management," *arXiv preprint arXiv:2204.09603*, 2022.

[7] F. Stranieri, F. Stella, and C. Kouki, "Performance of deep reinforcement learning algorithms in two-echelon inventory control systems," *International Journal of Production Research*, pp. 1–16, 2024.

[8] M. Khirwar, K. S. Gurumoorthy, A. A. Jain, and S. Manchenahally, "Cooperative multi-agent reinforcement learning for inventory management," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2023, pp. 619–634.

[9] N. N. Sultana, H. Meisheri, V. Baniwal, S. Nath, B. Ravindran, and H. Khadilkar, "Reinforcement learning for multi-product multi-node inventory management in supply chains," *arXiv preprint arXiv:2006.04037*, 2020.

[10] L. Kemmer, H. von Kleist, D. de Rochebouët, N. Tziortziotis, and J. Read, "Reinforcement learning for supply chain optimization," in *European Workshop on Reinforcement Learning*, vol. 14, no. 10, 2018.

[11] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent cooperation and competition with deep reinforcement learning," *PloS one*, vol. 12, no. 4, p. e0172395, 2017.

[12] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[13] A. Kumar, R. Agarwal, X. Geng, G. Tucker, and S. Levine, "Offline q-learning on diverse multi-task data both scales and generalizes," *arXiv preprint arXiv:2211.15144*, 2022.

[14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[15] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2681–2690.

[16] F. B. S. L. P. Janssen, R. M. J. Heuts, and T. de Kok, "The value of information in an (r, s, q) inventory model," 1996.