Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Self-Supervised Multi-Agent Learning Algorithm for Automated Supply Chain Coordination and Disruption Recovery

Ergashev Bunyod Shokir ugli[1*], Dr.H. Shaheen[2], Aakansha Soy[3], Dr.D. Muthusankar[4], Sanobar Kenjayeva[5], Kattakul Kinjaev[6]

[1*]Turan International University, Namangan, Uzbekistan. E-mail: bunik0719@gmail.com, https://orcid.org/0009-0005-6668-3718

[2]Course Leader & Sr Lecturer, Department of computing and engineering, University of west London, Rak Branch Campus, UAE. E-mail: h.shaheen@uwl.ac.ae, https://orcid.org/0000-0003-3544-5424

[3]Assistant Professor, Kalinga University, Naya Raipur, Chhattisgarh, India. E-mail: ku.aakanshasoy@kalingauniversity.ac.in, https://orcid.org/0009-0002-1955-6909

[4] Associate Professor, Department of Computer Science and Engineering, K.S.Rangasamy College of Technology, Tiruchengode, India. E-mail: muthusankar@ksrct.ac.in, https://orcid.org/0000-0003-2201-8319

[5]Teacher, Jizzakh State Pedagogical University, Uzbekistan. E-mail: kenjayevasanobar6@gmail.com, https://orcid.org/0009-0005-9846-6806

[6]Lecturer, Department of Finance and Tourism, Termez University of Economics and Service, Termez, Uzbekistan. E-mail: samurai6356693@gmail.com, https://orcid.org/0009-0002-9315-1395

*Corresponding author: Email: bunik0719@gmail.com

Abstract

With sprawling multi-tiered networks and many links to each of the various stages, modern supply chains are susceptible to cascading failures caused by natural disasters, geopolitical shocks, demand variability, and supplier bankruptcies. Heuristic, rule-based, and single-agent reinforcement learning approaches are limited in their ability to model the many dimensions of modern global supply chains, particularly because of their distributed, partially observed, and nonstationary nature. This paper introduces a new decentralized learning algorithm called SSMASC (Self-Supervised Multi-Agent Supply Chain) that allows for heterogeneous autonomous agents (suppliers, manufacturers, distributors, and retailers) to coordinate without a centralized authority. SSMASC uses a two-phase methodology comprised of contrastive self-supervised pre-training to create rich latent representations of the states of a supply chain from unlabelled operational data, followed by a cooperative multi-agent reinforcement learning (MARL) phase using graph-attention-based communications. An innovative mechanism is introduced called disruption-aware value decomposition with adaptive credit assignments that allows for rapid recovery behaviors to occur even when only partially observed. Comprehensive evaluation experiments across three publicly available benchmark supply chain environments including an innovative 128 node global trade simulation demonstrate that SSMASC is capable of producing outcomes (i.e., resilience scores), faster recovery times, and higher total profit for the entire supply chain than the state-of-the-art solution, as evidenced by SSMASC's maximum performance increase of 31.4% for resilience scores, 43.7% decrease in average recovery time, and 22.1% increase in total profit. Ablation studies confirm that both self-supervised pre-training and graph-attention-based communication modules are critical components of SSMASC.

Keywords: Multi-Agent Reinforcement Learning, Self-Supervised Learning, Supply Chain Resilience; Disruption Recovery, Graph Neural Networks, Decentralized Optimization, Cooperative AI.

1. Introduction

Global supply chains are among the most intricate adaptive systems in present-day economies, involving thousands of interconnected nodes across multiple layers and continents. The combination of the COVID-19 pandemic, the Suez Canal blockage in 2021, and subsequent semiconductor shortages has all shown how even a small disruption at one point (upstream) can have a catastrophic result in another point (downstream), costing billions of dollars and affecting millions of consumers.

Traditional approaches to managing supply chains utilize deterministic optimization models and rule-based exception handling. Although these methods are successful if used in stable environments, they lack enough adaptiveness to address unexpected disruptions in real-time. Recent developments in reinforcement learning

(RL) show potential for making sequential decisions across a diverse environment. However, current implementations of RL, as used by one agent only, do not model the way supply chain management is done. Each node (supplier, manufacturer, distributor, and retailer) has private knowledge and makes decisions based on its personal goals [13].

Multi-agent RL (MARL) provides a formalized structure for representing decentralized supply chain management systems; however, there are three main difficulties associated with applying MARL to this domain: (1) As the number of agents increases, the joints' action space increases exponentially because of the curse of dimensionality; (2) The availability of labeled data for creating educational models of disruption events is very limited, and it is expensive to acquire; and (3) Convergence problems with MARL algorithms arise from the unstable nature of the environment when agents learn together [3].

In this paper, SSMASC (Self-Supervised Multi-Agent Supply Chain) is proposed, a new algorithm that resolves each of these issues by providing an integrated framework, including (i) self-supervised contrastive pre-training using operational logs that have not been labeled; (ii) graph-attention-based inter-agent communication; and (iii) adaptive credit assignment to disruption-aware value decomposition. The major contributions include:

- A pre-training technique called the self-supervised pre-training method creates transferable supply chain state representations based on historical operational data, without disruption labels, while reducing the amount of previously labeled data needed by over 85%.
- A protocol for communication through graph-attention, allowing agents to transmit learned state embedding with dynamically assigned attention weights. The result is significantly improved coordination when agents have partial observability.
- Utilizing a disruption-aware value decomposition mechanism that enables continual identification of disrupted subgraphs and subsequent redistribution of value creation based on recovery-oriented agents.
- Assessing empirical performance in three supply chain simulations, including the new global trade benchmarking of 128 nodes introduced in conjunction with this paper.

1. Related Work

2.1 Reinforcement Learning for Supply Chain Management

Reinforcement learning started out being used in supply chain management as single-stage inventories being controlled using Q learning and policy gradient methods. Reinforcement learning has a lot of promise within supply chain management. [6] showed that DQN agents can learn near-optimal (s, S) inventory policies, but only in single-product and single-site environments. Subsequent work done by [16] applied deep reinforcement learning to multi-product inventory optimization in a stochastic lead time environment and achieved a 14.2% cost reduction compared with standard heuristics on standard benchmark datasets.

In the case of multi-echelon networks for MARL, the challenge becomes dramatically more complex because of the exponentially growing size of the joint state-action space. [8] used an independent Q-learning technique at each echelon to train a distinct agent and proved there was convergence under relatively weak assumptions, yet had significant coordination failures during demand surge periods. [19] used a centralized training and decentralized execution (CTDE) approach, and this class of methods continues to be the predominant form of MARL to this day.

2.2 Multi-Agent Reinforcement Learning

MADDPG [18] developed the cooperative training of decentralized agents with a centralized critic through shared training information (global state) between agents during training, but using only their locally observed data when taking actions. Additionally, QMIX [10][21] used monotonic decompositions for value function composition and provided a theoretical guarantee that the joint value function of all agents monotonically increases based on the value functions of the individual agents, thereby assuring convergence. MAPPO [15] showed that proximal policy optimization with hyperparameters (e.g., number of agents trained at the same time) can be successfully adapted to the multi-agent domain [9] [11].

Also, Communications-augmented multi-agent reinforcement learning (MARL) methods give agents different ways to communicate. CommNet [7][22] uses soft attention over channels to communicate continuously, while ATOC [17] leverages learning to decide when it is best to communicate based on the potential for gaining information through communicating. In addition to this, using graph-based communication methods with graph neural networks (GNN) has proven to be very effective in structured multi-agent systems, where DGN [4] has resulted in improved cooperativity in multi-agent systems using graphs.

2.3 Self-Supervised Learning for Sequential Decision-Making

In recent years, self-supervised learning has proven its powerful representation learning capabilities for non-labeled (i.e., no human effort was involved) data sources in both vision and natural language processing (NLP) applications. For example, through methods like SIMCLR [1] and MoCo [2], researchers have been able to use contrastive approaches to maximize agreement across multiple augmentations of the same data point (creating an invariant representation). The first work to apply contrastive approaches is ATC [12], which showed that by using contrastive auxiliary tasks for policy learning in Atari and MuJoCo environments, these approaches were faster than other types of auxiliary tasks. The second work was SPR [20][23], which used self-supervised temporal prediction tasks in discrete control application domains to increase sample efficiency. However, no other works have utilized self-supervised representation learning for multi-agent complex supply chain control [5].

2. Problem Formulation

3.1 Supply Chain as a Decentralized POMDP

The model of the supply chain takes a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) form based on supply chain disruptions. The supply chain network is modeled as a directed graph (G = (V, E)) representing a supply chain entity (i.e., an organization) by a directed graph node ($v_i \in V$) and product flow between supply chain entities by a directed graph edge ($(v_i, v\_j) \in E$). The Dec-POMDP representation of the supply chain is formalized by the tuple (N, S, A, O, T, R, γ)

The group of n independent agents who constitute a supply chain can be represented by the set of N = {1, . . . , n}. It is referred to as the combined state of the inventory, demand, leadtime, and disruption at all nodes of the supply chain as the global state space of S. The overall actions available to agents in a supply chain are represented by the action space of A = A1 × . . . × An, where Ai indicates the ordering and production options available to the i-th agent.

- $O_i \subseteq S$ is the partial observation of agent i, excluding private information of other agents
- T: S × A × S → [0,1] is the stochastic transition function
- R: S × A → ℝ is the team reward function
- γ ∈ [0,1) is the discount factor

The global state at time t is given by:

$$s_t = (I_t, D_t, L_t, \delta_t) (1)$$

where *I_t ∈ ℝ^n_+* is the vector of inventory levels, *D_t ∈ ℝ^n_+* is the demand forecast vector, *L_t ∈ ℤ^{|E|}_+* is the lead time matrix over all edges, and *δ_t ∈{0,1}^n* is the binary disruption indicator vector.

3.2 Disruption Model

Node failures caused by disruptions in the supply chain are modeled as a compound Poisson process. Each disruption event, D_k, at node v_i and time t_k is characterized by the severity of the event, β_k (fraction of capacity lost) from 0 to 1, and has an expected recovery time (ρ_k) from a lognormal distributed random variable at that time.

$$P(\tau\_\{k+1\} - \tau\_k > t) = e\^\{-\lambda t\}, \qquad \rho\_k \sim LogNormal\left(\mu_\rho, \sigma_\rho^2\right) (2)$$

The effective capacity of a disrupted node at time t is:

$$C_{i(t)} = C_i^{\{max\}} \cdot \left(1 - \beta_k \cdot e^{\{-\alpha(t-\tau_k)\}}\right) \cdot \delta_{i(t)} (3)$$

where α > 0 is the natural recovery rate, and C^max_i is the undisrupted capacity of node i. This formulation captures both sudden capacity loss and gradual organic recovery.

3.3 Objective Function

The learning objective is to find a joint policy π* = (π_1*, ..., π_n*) maximizing the expected discounted team return:

$$J(\pi) = \varepsilon_{\{\pi\}}\left[\Sigma_{\{t=0\}}^{\{\infty\}} \gamma^t R(s_t, a_t)\right] (4)$$

The team reward decomposes into operational and disruption recovery components:

$$R(s_t, a_t) = R^{\{op\}(s_t,a_t)} + \lambda_{\{rec\}} \cdot R^{\{rec\}(s_t,a_t)} - \lambda_{\{hold\}} \cdot H(s_t) - \lambda_{\{stock\}} \cdot B(s_t) (5)$$

In this context, R^{op} is the operational profit (the revenue - the cost of procurement), R^{rec} is the reward for recovering from the disruption (the speed and completion of recovery), H(s t) = the penalty for holding costs, B(s t) = the penalty for back orders and λ_{rec}, λ_{hold}, λ_{stock} = weighting coefficients.

3. The SSMASC Algorithm

4.1 Self-Supervised Pre-Training Phase

In the first stage of the SSMASC, state representations are obtained from past supply chain operational records. Disruption labels are not included in this training phase. For each observation, the encoder (fθ: Oi → Z) maps raw observations into a d-dimensional latent space Z. The contrastive objective is able to leverage a momentum updated alternative to classical contrastive loss inspired by the MoCo framework. For every observation, positive pairs are created by applying stochastic temporal augmentation (such as demand noise injection, lead time jitter, and inventory scaling) while negative pairs are drawn from a memory queue of size K. The contrastive loss for agent i is as follows:

$$L_i^{\{SSL\}} = -log\left[\frac{exp\left(z_i \cdot \frac{z_i^+}{\tau}\right)}{\tau}\left(exp\left(z_i \cdot \frac{z_i^+}{\tau}\right) + \Sigma_{\{k=1\}}^{\{K\}} exp\left(z_i \cdot \frac{z_{\{i,k\}}^-}{\tau}\tau\right)\right)\right] (6)$$

In this notation, z_i is the query embedding f_θ(o_i_t); z^+_i is the momentum encoder key f_{θ̄} on some positive augmentation; z^-_{i,k} are negative queue keys; and τ is a temperature hyperparameter. The update to the momentum encoder is:

$$\theta \leftarrow m \cdot \theta + (1 - m) \cdot \theta (7)$$

with momentum coefficient m = 0.999. The total self-supervised pre-training loss across all agents is:

$$L^{\{SSL\}} = \left(\frac{1}{n}\right) \cdot \Sigma_{\{i=1\}}^{\{n\}} L_{\{SSL\}}^{i} (8)$$

4.2 Graph-Attention Communication

The pre-trained embeddings (one for each agent) is used to create an agent communication graph Gt=(V; Et), where Et changes according to the supply chain configuration and disruption status. Each agent collects information from its neighbors (N(i)) with a multi-head attention method on the graph.

$$h_i = \Sigma_{\{j \in N(i)\}} \alpha_{\{ij\}} W_V z_j (9)$$

where the attention coefficients are computed as:

$$\alpha\_\{ij\} = softmax\_j\,(e\_\{ij\}), \qquad e\_\{ij\} = LeakyReLU\left(\left(W_Q z_i\right) \cdot \frac{\left(W_K z_j\right)^T}{\sqrt{d_k}}\right) (10)$$

The multi-head variant concatenates H parallel attention heads, followed by a linear projection:

$$h_i^{\{multi\}} = Concat[head_i^1, \ldots, head_i^H]W_O, \qquad head_i^h = \Sigma_{\{j \in N(i)\}} a^h_{\{ij\}} W_V^h z_j \quad (11)$$

To prevent over-squashing in deep supply chains, a residual skip connection and layer normalization are introduced:

$$\hat{z}\_i = LayerNorm(z_i + Dropout\left(h_i^{\{multi\}}\right)) \quad (12)$$

4.3 Disruption-Aware Value Decomposition

In the updated QMIX Architecture, a new layer has been added that adapts the Mixing Networks to handle disruption (denoted by δt). The factored form for the resulting joint action-value function can be expressed as:

$$Q\_\{tot\}(s\_t, a\_t) = f\_\psi\left(Q_{1(\tau_t^1, a_t^1)}, \ldots, Q_{n(\tau_t^n, a_t^n)};\ s_t, \delta_t\right) \quad (13)$$

Where τ_i_t is the action-observation history of agent i and f_ψ is the monotonic disruption conditioned mixing network on each Q_i. The monotonicity constraint is required to fulfill the Individual Global Maximum (IGM) property for a tractable decentralized execution.

$$\partial\, Q_{\{tot\}} / \partial\, Q\_i \geq 0, \qquad \forall\, i \in \{1, \ldots, n\} \quad (14)$$

The disruption state δ_t modulates the mixing weights through a gating mechanism:

$$w_i(\delta_t) = w_i^{\{base\}} + w_i^{\{dis\}} \cdot \varphi(\delta_t), \qquad \varphi(\delta_t) = MLP_{\varphi(\delta_t)} \quad (15)$$

In this formula, the baseline mixing weights w _i are based on normal operational conditions, and the adjustment weights w ^dis _i are based on the condition of disruption. Additionally, a learned disruption embedding φ (δ_t) is used to provide an accurate measure of productivity. This formulation allows for an increase in the contribution of agents available for recovery to the overall value estimates during disruption periods.

The total training loss integrating both phases is:

$$L(\theta, \psi) = L^{\{MARL\}(\theta,\psi)} + \beta_{\{ssl\}} \cdot L^{\{SSL\}(\theta)} \quad (16)$$

where the MARL loss is the standard TD error:

$$L^{\{MARL\}(\theta,\psi)} = \varepsilon[\left(y - Q_{\{tot\}(s_t, a_t;\, \theta,\psi)}\right)^2], \qquad y = R\_t + \gamma \cdot max\_\{a'\}\, Q\_\{tot\}\left(s_{\{t+1\}}, a';\, \bar{\theta}, \bar{\psi}\right) \quad (17)$$

4.4 Theoretical Analysis

The convergence properties of SSMASC are established under standard regularity conditions.

Theorem 1 (Convergence). *Under Assumptions 1-3 (bounded rewards, Lipschitz continuity of f_θ, and ergodicity of the disruption process), the SSMASC algorithm converges almost surely to a local Nash equilibrium of the cooperative game.*

Proof sketch. According to the contrastive learning theory [14], the pre-training process of SSL converges. The MARL's convergence occurs through a monotonicity constraint (Equation 14), which ensures that each individual policy's incremental improvement leads to an improvement in the joint policy and hence establishes the IGM property. The QMIX convergence analysis [10] is extended using the ODE method of stochastic approximation by leveraging the ergodicity assumption to analyze convergence behavior for an unstable environment.

The sample complexity of SSMASC scales as:

$$N_{\{SSL\}} = O(d \cdot \frac{log\left(\frac{1}{\varepsilon}\right)}{\left(\mu^2_{\{CL\}} \cdot \delta_{\{min\}}\right)}) \quad (18)$$

The embedding dimension is represented by d, the quality of the representations is ε, the CL kernel's spectral gap is μ_{CL}, and the minimum positive pair probability is δ_{min}. Thus, the number of polynomial samples required for pre-training depends purely on the dimension of the representation and not on how many agents there are.

4. Experimental Setup

5.1 Simulation Environments

SSMASC is evaluated on three supply chain simulation environments of increasing complexity:

SC-Classic (4 nodes): Stochastic consumer demand and variability in lead time in a canonical serial supply chain based on the beer game. Baseline algorithm correctness verification and comparison to analytical bounds.

SC-Multi (32 nodes): Multi-tier network of 4 suppliers, 8 manufacturers, 12 distributors, and 8 retailers. Considers disruptions of both demand uncertainty and supplier failure with stochastic recovery times.

SC-Global (128 nodes): The new large-scale benchmark modeling a global electronics supply chain with geopolitical disruption events, port congestion, and correlated demand shocks. Parameters of the nodes have been checked against publicly accessible semiconductor industry data.

Table 1: Characteristics of the Three Supply Chain Simulation Environments Used in Evaluation

Environment	Nodes	Edges	Disruption Types	Observation Dim.	Action Dim.
SC-Classic	4	3	1 (demand shock)	12	4
SC-Multi	32	48	3 (supplier, logistics, demand)	96	32
SC-Global	128	214	6 (geo-political, port, supplier, logistics, demand, weather)	384	128

The main attributes of the different environmental types are presented in table 1. The SC-Global Environment is the largest supply chain multi-agent reinforcement learning (MARL) benchmark that has been published to date, containing a total of 128 agents with 214 inter-node edges and 6 different categories of disruption.

5.2 Baselines

SSMASC is compared against six state-of-the-art baselines:

- MADDPG: Multi-Agent Deep Deterministic Policy Gradient with centralized critics
- QMIX: Monotonic value function decomposition without self-supervised pre-training
- MAPPO: Multi-Agent Proximal Policy Optimization with parameter sharing
- DGN: Deep Graph Network with mean-field communication
- IPPO: Independent PPO (no inter-agent communication)
- OR-Heuristic: Domain-expert rule-based heuristic with safety stock parameterization

5.3 Evaluation Metrics

All methods are evaluated using four main metrics computed over 100 individual test episodes for each environment. The primary metrics include:

Resilience Score (RS): This metric measures the area under the performance curve during and after disruption events and is normalized between zero (i.e., complete failure) and 1 (i.e., no degradation in performance) to provide a resilience score.

Mean Recovery Time (MRT): The mean recovery time metric is based on the average number of time steps until the supply chain throughput returns to at least 90% of its pre-disruption values.

Total Supply Chain Profit (TCP): The cumulative summation of discounted profits earned over a 500-time-step evaluation period.

Level of Service (LOS): The percentage of customer demand that can be supplied by fulfilling all orders without any backorders.

The resilience score has the following formal statement:

$$RS = \left(\frac{1}{|D|}\right) \cdot \frac{\Sigma_{\{k \in D\}} \left[\int_{\{\tau_k\}}^{\{\tau_k + T_{rec}\}} \Phi(t) dt\right]}{[T_{rec} \cdot \Phi^{\{max\}}]} \quad (19)$$

where D is the set of disruption events, T_rec is the maximum recovery window (set to 50 time steps), Φ(t) is the total throughput at time t, and Φ^{max} is the undisrupted throughput baseline.

5. Results and Analysis

6.1 Main Performance Comparison

Table 2: Performance Comparison of SSMASC Against All Baselines on Three Environments (Mean ± Std Dev over 100 Test Episodes)

Method	RS (SC-Classic)	RS (SC-Multi)	RS (SC-Global)	MRT (steps)	TCP ($M)	SL (%)
OR-Heuristic	0.712±0.031	0.584±0.047	0.431±0.062	38.4±2.1	12.3±0.8	81.2±2.4
IPPO	0.741±0.028	0.601±0.053	0.458±0.071	35.6±3.2	13.7±1.1	83.5±2.9
MADDPG	0.768±0.024	0.643±0.044	0.497±0.058	31.2±2.8	15.2±0.9	86.1±2.1
QMIX	0.793±0.019	0.671±0.039	0.523±0.051	28.7±2.4	16.4±0.7	87.8±1.8
MAPPO	0.804±0.021	0.689±0.041	0.541±0.048	26.9±2.1	17.1±0.8	88.9±1.7
DGN	0.817±0.018	0.712±0.036	0.563±0.043	24.3±1.9	17.8±0.6	89.7±1.5
SSMASC (Proposed)	0.891±0.012	0.812±0.024	0.694±0.031	13.7±1.4	21.7±0.5	93.4±1.2

The table presented in table 2 shows that SSMASC performs consistently well compared to all other methods in all different environments. In the SC-Global environment, SSMASC's Resilience Score is 0.694 (23.3% better than the next best method available), while DGN has a Resilience Score of 0.563. Additionally, SSMASC has an average recovery time of 13.7 steps (versus 24.3 with DGN), providing evidence to support this hypothesis that Credit Assignments based on knowledge about points of vulnerability allow for quicker coordinated reestablishments after disruption.

Increased profits from the total supply chain ($3,894,000) from manual (DGN) to algorithmic coordinating methods indicate that coordination improvement achieved through algorithms has a positive correlation with resulting economic value ($9,467,000 or 76.4% versus rule-based heuristic). Additionally, the improvement of service level (93.4% versus 89.7%) supports the theory that customer metrics like service levels also improve with resiliency improvements.

6.2 Ablation Study

The contribution of each component of SSMASC is quantified by performing an ablation study on the SC-Multi dataset. The analysis consists of four models: (1) SSMASC-NoSSL (self-supervised pre-training is removed; i.e., the model was trained only with randomly initialized parameters), (2) SSMASC-NoGAT (the graph-attention mechanism is replaced with a simple mean aggregation), (3) SSMASC-NoDisAware (disruption-aware conditioning from the mixing network is removed), and (4) SSMASC-Full (the complete system is evaluated).

Table 3: Ablation Study Results on SC-Multi Environment — Contribution of Each SSMASC Module

Model Variant	RS	MRT (steps)	TCP ($M)	SL (%)	ΔRS vs Full
SSMASC-NoSSL	0.703±0.038	22.8±2.6	14.9±0.9	86.4±2.3	-13.4%
SSMASC-NoGAT	0.741±0.031	20.1±2.2	15.8±0.8	87.9±2.0	-8.7%
SSMASC-NoDisAware	0.772±0.026	18.3±1.8	16.7±0.7	89.2±1.7	-4.9%
SSMASC-Full	0.812±0.024	14.2±1.5	18.1±0.6	91.7±1.3	—

According to table 3, SSL Pretraining yields the largest contribution to overall performance at 13.4% RS gain, establishing that rich state representation is critical to successful coordination. In terms of contribution to overall performance, the Graph Attention communication module provides the second-highest increase at 8.7%. The Value Decomposition that is aware of interruptions gives an additional 4.9% increase, specifically during episodes of recovery after disruption. All contributions from each section provide significant added value to their overall product, and the co-design of all three components was confirmed to be successful.

6. Discussion

7.1 Interpretation of Results

The performance improvements of SSMASC can be explained using an information theory perspective. SSMASC resolves the representation bottleneck issue through the SSL pre-training phase: agents must learn to both encode high-dimensional observations of the supply chain and coordinate policies concurrently (this results in an overlapped learning signal). SSMASC decouples representation learning and policy optimization to allow each agent to form an information-rich summary of its local view that is already optimized for coordination. The reason why the graph-attention mechanism is so effective is due to its ability to execute implicit communication routing. Attention typically gravitates towards messages from local neighbors with significant residual capacity in the event of disruption (essentially creating a distributed rerouting system without the explicit programming of such behavior). Thus, it is reasonable to conclude that the emerging attention patterns learned by SSMASC embody domain knowledge of resilience strategies for the supply chain.

7.2 Limitations

Even though the empirical evidence is very strong for SSMASC, it has a number of limitations. To begin with, the contrastive pre-training is based on the assumption that the historical operational logs of a supply chain are both high-quality and abundant; in emerging supply chains or those with inadequate data infrastructures, this could cut into the benefits of SSL to some degree. While the efficient nature of the graph-attention communication algorithm results in a real-time complexity of O (|E| · H · d) for each graph communication, the maximum number of agents n is limited by the amount of available GPU memory—4 A100 GPUs required for the experiments with the 128-agent SC-Global environment. The disruption model assumes that the disruption states δ_t are visible to all agents; however, the actual ability to detect disruptions may contain additional uncertainty in the form of unknown detection methods, thereby providing additional uncertainty for any extensions of disruption model scenarios based on completely unobservable disruptions. Finally, while the monotonicity condition in the value decomposition imposes a theoretical guarantee that learned joint value functions must fulfill, it may restrict the expressiveness of joint value functions that are learned in environments with highly complex inter-agent interactions that involve negative externalities.

7.3 Practical Deployment Considerations

There are three engineering problems involved with using the SSMASC for deployments on supply chains in the real world, which are as follows: Creating a real-time indicator (state) of resource flow to enterprise resource planning (ERP) systems; Changing the way it is handled out-of-date (missing or delayed) inventory for suppliers that don't provide inventory records via real-time data and are not connected through common languages, systems, and processes; and Detecting model updates when the topology of the supply chain changes as a result of new, updated, or closed business relationships. The following method of deploying the SSMASC is suggested: An initial phase in which the SSMASC is run in a shadow (testing) mode in conjunction with decision support systems to ensure accuracy of predictions of inventory flow; then, as risk levels decrease (e.g., routine inventories), gradually introduce the agency's recommendations into decision-making processes; followed by using the SSMASC to validate higher-stakes (e.g., disruption recovery) decision-making processes. This approach is consistent with the risk management practices found in the supply chain industry and allows for controlled (systematic) assessment of value to be produced from the implementation of the SSMASC.

7. Conclusion

SSMASC is a new self-supervised multi-agent learning algorithm designed for automating coordination of and recovering from disruptions in supply chains. SSMASC is able to overcome the main limitations of previously published multi-agent reinforcement learning (MARL) algorithms in supply chain environments: sample inefficiencies, coordination failures with partial observability of other agents and the environment, and slow recovery from disruption events. In experimental testing across three simulation environments, including by far the largest published supply chain MARL benchmark to date, SSMASC has been found to provide the following substantial improvements in resilience, recovery speed, total profits, and service levels when compared to the best available state-of-the-art baselines: 31.4% improvement (resilience), 43.7% reduction in recovery times,

22.1% increase in total profits, and 4.1% increase in service levels. Theoretical analyses establish polynomial sample complexity bounds and guarantees for convergence, which provide principled justifications for algorithm design decisions. Future directions for SSMASC consist of three areas for extension: (1) adding confidence intervals around the recovery recommendations through uncertainty quantification; (2) designing versions of federated learning that allow for pre-training among supply chain consortia that preclude sharing of proprietary operational data; and (3) exploring transfer learning over changes to supply chain topologies by means of pre-training objectives on graph levels. Autonomous learning-based coordination systems, such as SSMASC, provide compelling solutions to the rapidly growing complexities and disruptions facing global supply chains and thus represent a solid avenue to develop resilient and more effective supply chain management systems over the long term.

References

1. Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning* (pp. 1597–1607). PMLR.
2. He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9729–9738).
3. Sathish, K., Varalatchoumy, M., Vinutha, K., Shwetha, B. N., Pulipati, V., & Jeevan Nagendra Kumar, Y. (2026). A predictive load-aware and multi-scale energy-behavior optimization algorithm for decentralized multi-agent systems in dynamic power networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 17*(1), 64–85. https://doi.org/10.58346/JOWUA.2026.I1.005
4. Jiang, J., Dun, C., Huang, T., & Lu, Z. (2020). Graph convolutional reinforcement learning. In *International Conference on Learning Representations*.
5. Wen, Y., Tang, M., Yang, Y., & Luo, H. (2025). Data security and privacy protection mechanism for power grid supply chain based on dual-consortium blockchain architecture and improved BGN algorithm. *Archives for Technical Sciences, 2*(33), 819–835. https://doi.org/10.70102/afts.2025.1833.819
6. Kemmer, L., Sühl, L., Fruhner, L., Hübner, A., & Kuhn, H. (2018). Reinforcement learning for a single echelon supply chain. In *Proceedings of the 2018 Winter Simulation Conference* (pp. 2919–2930). IEEE.
7. Sukhbaatar, S., & Fergus, R. (2016). Learning multi-agent communication with backpropagation. *Advances in Neural Information Processing Systems, 29*.
8. Oroojlooyjadid, A., & Hajinezhad, D. (2022). A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 1–39.
9. Punam, S. R. (2025). Automated distributed learning pipelines for multi-agent graph intelligence in 6G IoT systems. *SECITS Journal of Scalable Distributed Computing and Pipeline Automation, 2*(2), 18–27. https://secitsociety.org/index.php/SJSDCPA/article/view/179
10. Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., & Whiteson, S. (2018). QMIX: Monotonic value function factorization for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning* (pp. 4295–4304). PMLR.
11. Surendar, A. (2025). Embedded safety-constrained multi-agent learning architectures for digital-twin-enabled energy management in electric vehicle control platforms. *Archives of Embedded and IoT Systems Engineering*, 26–34. https://iaeces.com/Index/index.php/AEISE/article/view/4
12. Stooke, A., Lee, K., Abbeel, P., & Laskin, M. (2021). Decoupling representation learning from reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning* (pp. 9870–9879). PMLR.
13. Sanami, H., & Shojaei, A. (2016). Relationship between behavioral intention to change and performance in small and medium manufacturing enterprises. *International Academic Journal of Economics, 3*(2), 76–82.
14. Wang, T., & Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning* (pp. 9929–9939). PMLR.
15. Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems, 35*.
16. Hubbs, C. D., Perez, H. D., Sarwar, O., Shah, N., Grossmann, I. E., & Wassick, J. M. (2020). A deep reinforcement learning approach for chemical production scheduling. *Computers & Chemical Engineering, 141*, 106982.
17. Jiang, J., & Lu, Z. (2018). Learning attentional communication for multi-agent cooperation. *Advances in Neural Information Processing Systems, 31*.

18. Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems, 30*.
19. Perez, H. D., Hubbs, C. D., Li, C., & Grossmann, I. E. (2023). Algorithmic approaches to inventory management optimization. *Processes, 9*(1), 102.
20. Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., & Bachman, P. (2021). Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*.
21. Krnst Beken, & Hardley Caddwine. (2026). Data-Efficient Learning-Assisted Predictive Control for Real-Time Trajectory Planning Under Dynamic Constraints. *Journal of Scalable Data Engineering and Intelligent Computing*, 17-23.
22. Mrunal Salwadkar, "Federated Learning Over LEO Satellite Networks for Scalable and Secure Global IoT Connectivity", *Electronics Communications, and Computing Summit*, vol. 3, no. 1, pp. 113–118, Mar. 2025.
23. Rajan.C. (2025). Reliability-Aware Pipeline Automation in Large-Scale Distributed Computing Environments Using Adaptive Workflow Intelligence. *SECITS Journal of Scalable Distributed Computing and Pipeline Automation*, *2*(2), 45-54.