EAI Endorsed Transactions
on Scalable Information Systems

Research Article **EAI.EU**

Design and Performance Evaluation of a Hybrid Task Scheduling Strategy for E-Commerce Logistics in Cloud-Edge Collaborative Environments

Yan Wang[1,*]

[1]Department of Digital Arts, Beijing Economic Management School, Beijing 100142, China

Abstract

INTRODUCTION: The rapid growth of e-commerce has led to highly concurrent and diverse logistics operations, posing significant challenges for task scheduling under dynamic network conditions and heterogeneous computing resources. Conventional cloud-centric scheduling lacks real-time responsiveness, while purely edge-based decisions suffer from limited global visibility, resulting in suboptimal performance.
OBJECTIVES: This study addresses these limitations by designing a cloud-edge collaborative hybrid scheduling method tailored for e-commerce logistics, aiming to simultaneously minimize latency, reduce energy consumption, and maximize task completion rates under fluctuating workloads.
METHODS: The proposed framework integrates three core components: task encoding to capture heterogeneity, node state prediction using lightweight temporal models, and multi-objective scheduling driven by reinforcement learning. This enables the system to adapt dynamically to changes in bandwidth, node load, and task urgency.
RESULTS: Evaluated under simulated peak and off-peak e-commerce scenarios, the method outperforms baseline approaches by reducing average task latency by 18.7%, increasing completion rate by 9.4%, and cutting system-wide energy consumption by 12.3%.
CONCLUSION: By effectively coordinating cloud and edge resources, the approach provides a robust foundation for building low-latency, energy-efficient, and reliable scheduling systems, with practical implications for warehouse automation, instant delivery networks, and other time-sensitive logistics applications.

Keywords: cloud-edge collaboration; task scheduling; e-commerce logistics; reinforcement learning; resource allocation

Received on 12 December 2025, accepted on 06 May 2026, published on 27 May 2026

doi: 10.4108/eetsis.11343

1. Introduction

With the rapid expansion of the e-commerce industry, order volume, business complexity, and real-time service demands continue to grow, placing significant pressure on logistics systems operating under high concurrency, short cycles, and diverse task types[1][2]. In scenarios such as large-scale promotions, instant delivery, and front-warehouse fulfillment, real-time tasks (e.g., route planning, inventory synchronization) and batch tasks (e.g., sales analysis, demand forecasting) arrive simultaneously, resulting in highly fluctuating computational loads. In environments characterized by distributed resources and heterogeneous devices, relying primarily on cloud computing often fails to meet low-latency requirements, while independent scheduling by edge nodes struggles to achieve optimal resource allocation due to the lack of

---

[*] Corresponding author. Email: jyyjs2022@163.com

global information[3]. Consequently, establishing an efficient collaborative mechanism between cloud-level computing power and edge-level near-field responsiveness has become a core challenge for intelligent e-commerce logistics.

Although existing research has attempted to leverage cloud-edge collaboration for logistics scheduling, several limitations remain. Many methods depend on static priorities or heuristic rules, which cannot accurately capture variations in task attributes or node status[4][5]; some frameworks overlook task urgency and bandwidth fluctuations, causing instability during business peak periods[6]. In addition, real-time and batch tasks are often handled separately, lacking unified management, which leads to suboptimal resource utilization[7]. Due to the absence of systematic evaluation across peak–off-peak cycles, the applicability of these methods in real e-commerce environments remains limited[8]. These issues indicate that addressing the heterogeneity of logistics tasks and the dynamic nature of the system requires a cloud-edge hybrid scheduling scheme with stronger predictive and adaptive capabilities.

To this end, this study proposes a cloud-edge collaborative hybrid scheduling strategy for e-commerce logistics, focusing on three practically relevant innovations. The first is task attribute encoding, which constructs learnable representations of latency sensitivity, computational demand, and data scale, mapping them into a unified feature space to enhance decision discriminability. The second is node state prediction, which utilizes a lightweight temporal model to forecast short-term bandwidth and workload, enabling proactive rather than reactive scheduling decisions. The third is multi-objective scheduling, which integrates reinforcement learning and jointly optimizes latency, energy consumption, and task completion rate to achieve dynamic allocation between cloud and edge nodes. All three innovations directly target real-world scheduling challenges and are feasible for engineering implementation.

A series of experiments conducted on a platform simulating realistic load fluctuations validate the proposed method. Compared with various baseline models, the method reduces average latency by approximately 18.7%, increases task completion rate by 9.4%, and lowers energy consumption by 12.3%, demonstrating that prediction-enhanced and collaborative optimization mechanisms can effectively improve overall system performance. Owing to its extensibility, the method has potential applications in warehouse scheduling, instant delivery, and supply-chain coordination, and provides a new technical pathway for multi-objective optimization in cloud-edge collaborative scheduling.

The remainder of this paper is organized as follows: Section 2 presents a systematic review of related work; Section 3 elaborates on the design of the task encoding model, node state prediction module, and multi-objective scheduling mechanism; Section 4 describes the experimental setup and analyzes the comparative results; the final section concludes the study and discusses future research directions, aiming to provide a reference framework for cloud-edge collaborative scheduling in complex e-commerce environments.

2. Related Works

2.1 Application Scenarios and Challenges

With the development of IoT, big data, and 5G/6G technologies, cloud-edge collaboration has been widely adopted in logistics, supply chains, and e-commerce delivery[9]. E-commerce logistics scheduling involves tasks such as real-time route planning, warehouse picking, inventory synchronization, vehicle dispatching, batch sales analysis, and demand forecasting[10]. These tasks exhibit large variations in device capabilities, strong network fluctuations, and mixed task types, and are simultaneously constrained by energy consumption, bandwidth, and storage[11][12]. Common evaluation metrics include latency, task completion rate, resource utilization, energy consumption, and cost, which reflect the effectiveness of scheduling strategies in complex scenarios[13].

The main challenges include the following: high task heterogeneity and varying urgency levels make it difficult for a unified scheduling strategy to simultaneously address real-time and throughput requirements; the resource environment is highly heterogeneous, cloud nodes provide abundant computing power but suffer from high latency, whereas edge nodes offer fast responses but limited computational capacity[14][15]. Combined with dynamic changes in workload and bandwidth, system states become hard to predict. Furthermore, existing scheduling methods lack sufficient collaboration, and the delayed synchronization between cloud and edge states limits stability under large-scale dynamic loads[16][17]. In addition, e-commerce peak–off-peak cycles fluctuate drastically, yet existing research is mostly conducted in static or idealized environments, lacking validation on large-scale hybrid tasks. In summary, cloud-edge hybrid scheduling is urgently needed in e-commerce logistics but is challenged by high task complexity, strong resource dynamics, and weak cross-node collaboration.

2.2 Review of Mainstream Methods

Existing cloud-edge scheduling approaches primarily include multi-objective optimization, intelligent scheduling, and offloading strategies. Multi-objective optimization methods target latency, energy consumption, and completion rate and coordinate resources through heuristic or swarm intelligence algorithms[18][19]. These approaches are stable in structured or single-task scenarios but lack modeling capabilities for task heterogeneity, node fluctuations, and hybrid task structures, limiting their applicability to the complex load patterns of e-commerce logistics[20].

Intelligent scheduling methods leverage deep learning to enhance adaptability, such as modeling task dependencies with graph neural networks and using reinforcement learning for dynamic allocation[21][22]. Although these methods perform well in highly dynamic environments, most rely on simulation-based validation and lack systematic evaluation in large-scale hybrid workloads with fluctuating loads[23]. Moreover, many models remain limited to local tasks or single scenarios, with insufficient understanding of business cycle variations and large-scale node coordination.

In mobile edge computing, priority-based and fuzzy logic methods have been used to improve urgent task responses. However, these approaches are built on general IoT assumptions and fail to model significant differences among logistics tasks in data scale, computational demand, and latency sensitivity[24]. They also lack mechanisms for coordinating real-time and batch tasks.

Offloading strategies reduce overall latency through hierarchical offloading and task preprocessing, making them suitable for latency-sensitive applications. However, most of these studies focus on the migration process itself and pay insufficient attention to load prediction, multi-task coordination, and energy constraints required in logistics scheduling.

Overall, while existing studies provide meaningful insights into latency and energy optimization, they generally lack a unified task model tailored to e-commerce logistics and fall short in addressing task heterogeneity, system dynamics, and cloud-edge coordination[25]. In particular, stability under peak–off-peak fluctuations has not been adequately validated.

2.3 Most Related Research

Among studies most closely related to this work, one research effort focuses on supply chain resource scheduling optimization for e-commerce enterprises in international trade scenarios using mobile edge computing[26]. That study constructs a resource allocation model for e-commerce environments and analyzes collaboration mechanisms between edge nodes and cloud resources. It is highly relevant to the e-commerce context and provides useful perspectives on intelligent scheduling within the e-commerce value chain[27]. The study offers new insights into supply chain management through resource scheduling optimization.

However, several differences remain when compared with this work. First, that study concentrates on overall supply chain resource scheduling rather than hybrid (real-time + batch) logistics task scheduling. Second, its methodology relies on traditional optimization algorithms and does not incorporate task attribute encoding, node state prediction, or reinforcement learning–based multi-objective scheduling[28]. Third, its scenario validation mainly addresses macro-level supply chain processes and lacks detailed modeling of cloud-edge collaboration in logistics route planning, warehousing, and delivery operations[29]. Nonetheless, it provides a meaningful reference for edge computing applications in e-commerce contexts[30]. Building upon this, the present study further focuses on logistics tasks themselves, task-type integration, dynamic cloud-edge coordination, and multi-objective optimization, achieving clear differentiation and methodological advancement.

2.4 Summary

Overall, the existing literature has made substantial progress in cloud-edge task scheduling, resource allocation, and offloading optimization, but research largely remains concentrated on general edge computing environments, IoT scenarios, manufacturing systems, or macro-level supply chain resource management. Research on e-commerce logistics scheduling, characterized by mixed real-time and batch tasks, heterogeneous resources, rapidly changing bandwidth and node states, and drastic peak–off-peak load fluctuations, remains insufficient. Although previous studies have emphasized metrics such as latency, energy consumption, and task completion rate, systematic design and empirical analysis of task encoding, node state prediction, cloud-edge collaboration mechanisms, and multi-objective joint optimization remain limited.

Therefore, there is a pressing need to explore unified task models, node prediction models, and scheduling decision mechanisms specifically tailored to e-commerce logistics scenarios. Unlike existing studies, this work proposes an integrated solution framework for hybrid logistics task scheduling through task attribute encoding, node state prediction, and multi-objective collaborative scheduling, offering innovations in both practical application and theoretical methodology. The next section provides a detailed description of the model design and algorithmic implementation of the proposed method.

3. Methodology

This section proposes a cloud-edge collaborative hybrid scheduling strategy designed for e-commerce logistics scenarios. The method consists of three core modules: a task attribute encoding module, a node state prediction module, and a multi-objective scheduling decision module. The overall method follows a unified "representation-prediction-decision" framework. Given mixed loads of real-time tasks (RT) and batch tasks (BT), the strategy jointly models dynamic resource states and task features to optimize scheduling across cloud and edge nodes.

3.1 Problem Formulation

In an e-commerce logistics system, the task set is defined as $\mathcal{T} = \mathcal{T}^{RT} \cup \mathcal{T}^{BT}$, where $\mathcal{T}^{RT}$ denotes the set of real-time tasks and $\mathcal{T}^{BT}$ denotes the set of batch tasks.

3.1.1 Inputs and Task Features

Each task i is defined by the tuple

$$i = (c_i, s_i, \delta_i, \tau_i), \quad (1)$$

where:

c_i: computation demand (FLOPs)

s_i: data size (MB)

δ_i: latency sensitivity (RT ≥ BT)

τ_i: arrival time

The system contains two categories of nodes: the cloud node set $\mathcal{C}$ and the edge node set $\mathcal{E}$, forming the complete node set

$$\mathcal{N} = \mathcal{C} \cup \mathcal{E}. \quad (2)$$

Each node nnn is characterized by

$n = (f_n, b_n, l_n)$,

where:

f_n: computing capability (GFLOPs/s)

b_n: available bandwidth (MB/s)

l_n: current load ratio

3.1.2 Latency Model

When task i is assigned to node n, its end-to-end latency is

$$D_{i,n} = D_{i,n}^{tx} + D_{i,n}^{comp} + D_{i,n}^{queue} \quad (3)$$

Transmission latency:

$$D_{i,n}^{tx} = \frac{s_i}{b_n} \quad (4)$$

Computation latency:

$$D_{i,n}^{comp} = \frac{c_i}{f_n(1-l_n)} \quad (5)$$

Queueing latency:

$$D_{i,n}^{queue} = \alpha \cdot l_n \quad (6)$$

where α is an empirical coefficient. Although real-world e-commerce traffic typically follows non-linear queueing models, Equation (6) deliberately adopts a linear approximation. This computationally efficient surrogate prevents gradient explosion during early RL training when load approaches capacity, while still providing stable congestion penalties.

A task is completed on time if the constraint

$$D_{i,n} \le D_i^{\max} \quad (7)$$

is satisfied, where $D_i^{\max}$ is the allowable maximum latency (more stringent for RT tasks).

3.1.3 Objective

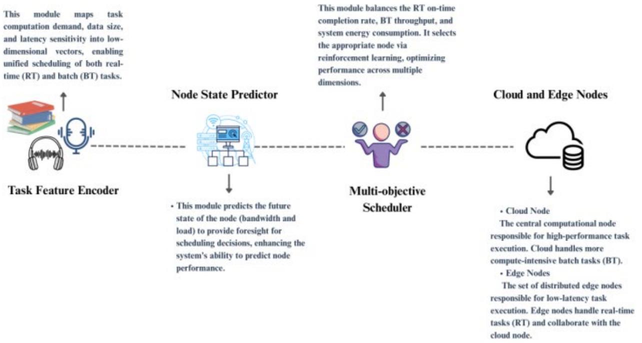
The goal is to maximize RT on-time completion rate, maximize BT throughput, and minimize system energy consumption. Formally,

$$\max_{\pi}(\eta_{RT}, \eta_{BT}, -E_{total}) \quad (8)$$

where π denotes the scheduling policy.

3.2 Overall Framework

The overall framework consists of three core modules (see Figure 1), forming a closed-loop process of "representation-prediction-decision." First, the task attribute encoding module maps task computation demand, data size, and latency sensitivity into low-dimensional vectors, enabling unified scheduling of both real-time and batch tasks. To eliminate scale disparities among different attributes, all raw input features are normalized to the range [0, 1] prior to this mapping. Second, the node state prediction module forecasts the next timestep's bandwidth and load based on historical sequences, providing foresight for scheduling and preventing decisions based solely on instantaneous states. Finally, the multi-objective scheduling decision module integrates task features and prediction outputs and selects nodes via reinforcement learning, optimizing latency, completion rate, and energy consumption while achieving collaborative allocation between cloud and edge nodes.



This module maps task computation demand, data size, and latency sensitivity into low-dimensional vectors, enabling unified scheduling of both real-time (RT) and batch (BT) tasks.

This module balances the RT on-time completion rate, BT throughput, and system energy consumption. It selects the appropriate node via reinforcement learning, optimizing performance across multiple dimensions.

- This module predicts the future state of the node (bandwidth and load) to provide foresight for scheduling decisions, enhancing the system's ability to predict node performance.

- Cloud Node

The central computational node responsible for high-performance task execution. Cloud handles more compute-intensive batch tasks (BT).

- Edge Nodes

The set of distributed edge nodes responsible for low-latency task execution. Edge nodes handle real-time tasks (RT) and collaborate with the cloud node.

Figure 1. Overall Framework of the Proposed Hybrid Cloud-Edge Scheduling Architecture

3.3 Module Descriptions

3.3.1 Task Feature Encoder

(1) Motivation: RT and BT in e-commerce logistics differ significantly in computation demand, data volume, and latency sensitivity. If raw attributes are directly used as scheduling inputs, the model may struggle to capture intrinsic structural characteristics across task types. Therefore, task attributes must be transformed into low-dimensional learnable representations to enable scheduling decisions within a unified feature space.

(2) Principle: For each task iii, the original feature vector is constructed as

$$\mathbf{x}_i = [c_i, s_i, \delta_i], \quad (9)$$

where c_i is the computation demand, s_i is the data size, and δ_i is the latency sensitivity. To explicitly prioritize real-time tasks, the remaining time until the deadline is incorporated into δ_i, serving as a dynamic urgency feature in the state representation to ensure strict responsiveness. To eliminate scale disparities among different attributes, all raw input features are normalized to the range [0, 1] prior to this mapping. Then, $\mathbf{x}_i$ is mapped into a hidden space via a feed-forward network to capture the non-linear interactions among attributes:

The task encoding is completed using a two-layer feedforward network:

$$\mathbf{h}_i = \sigma(W_1\mathbf{x}_i + b_1) \quad (10)$$
$$\mathbf{e}_i = \sigma(W_2\mathbf{h}_i + b_2) \quad (11)$$

where $\sigma(\cdot)$ is the ReLU activation function, and W_1, W_2, b_1, b_2 are learnable parameters. The encoded vector $\mathbf{e}_i$ serves as part of the scheduler state input.

(3) Implementation: The structure of the task encoding module is shown in Figure 2, consisting of two fully connected layers with fixed output dimensionality to ensure comparability among tasks. The pseudocode is provided in Algorithm 1, with steps corresponding to feature concatenation, first-layer mapping, second-layer mapping, and final embedding output.

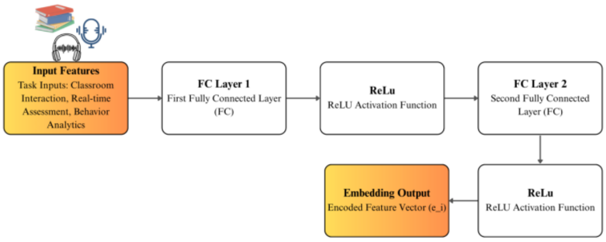


Figure 2. Architecture of the Task Feature Encoder

```
Algorithm 1. TaskFeatureEncoding(i)
Input: task attributes (c_i, s_i, δ_i)
Output: encoded feature e_i

x ← concat(c_i, s_i, δ_i)
h1 ← ReLU(W1 * x + b1)
e_i ← ReLU(W2 * h1 + b2)
return e_i
```

3.3.2 Node State Predictor

(1) Motivation: E-commerce logistics workloads fluctuate significantly over time, especially during peak–off-peak cycles, where network bandwidth and node load vary markedly. If the scheduler makes decisions solely based on the current state, errors in latency estimation and resource imbalance may occur. Therefore, short-term prediction of future node states is necessary to enhance the foresight of scheduling decisions.

(2) Principle: Let the state vector of node n at time t be

$$\mathbf{s}_n(t) = [b_n(t), l_n(t)], \quad (12)$$

where $b_n(t)$ is the available bandwidth and $l_n(t)$ is the load.

An LSTM is used for time-series modeling, with the hidden state updated as

$$\mathbf{h}_t = \mathsf{LSTM}(\mathbf{s}_n(t), \mathbf{h}_{t-1}), \quad (13)$$

where $\mathbf{h}_t$ is the hidden state.

The predicted next-step state is

$$\hat{\mathbf{s}}_n(t+1) = W_o\mathbf{h}_t + b_o, \quad (14)$$

where the predicted bandwidth and load serve as scheduler inputs.

(3) Implementation: The structure of the node prediction module is shown in Figure 3, which uses multi-step time-series input to forecast short-term resource trends. The pseudocode in Algorithm 2 illustrates the complete process of sequence handling, LSTM state updates, and final prediction output.

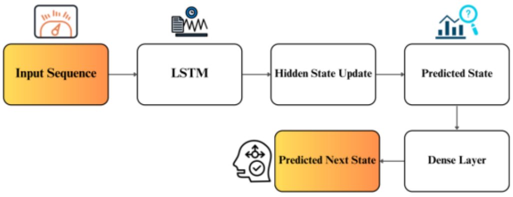


Figure 3. Node State Predictor (LSTM-based)

```
Algorithm 2 NodeStatePrediction(n)
Input: state sequence S_n(t)
Output: predicted next state Ŝ_n(t+1)

h ← init_hidden()
for each s_t in S_n:
    h ← LSTM(s_t, h)
end for
Ŝ ← W_o * h + b_o
return Ŝ
```

3.3.3 Multi-objective Scheduler

(1) Motivation: The scheduler must simultaneously optimize RT on-time completion rate, BT throughput, and system energy consumption. In e-commerce environments, task arrivals are highly uncertain and resources fluctuate significantly, making rule-based or static optimization methods insufficiently adaptive. Therefore, a RL-based multi-objective scheduler is adopted so that the policy can dynamically adjust to system state changes.

(2) Principle: The scheduler state input consists of the task encoding $\mathbf{e}_i$ and the predicted node state $\hat{\mathbf{s}}_n(t+1)$:

$$s_t = (\mathbf{e}_i, \hat{\mathbf{s}}_n(t+1)). \quad (15)$$

This equation mathematically materializes the synergistic integration of the three modules. By fusing the spatial-semantic task representation $\mathbf{e}_i$ (extracted by the encoding module) with the temporal forecasted node state $\hat{s}_n(t+1)$ (extracted by the prediction module), the RL agent is endowed with a truly proactive scheduling capability. Rather than reacting passively to delayed instantaneous metrics, the agent anticipates upcoming resource fluctuations and makes predictive assignments.

The scheduling action is to select a node:

$$a_t \in \mathcal{N}. \quad (16)$$

The reward function is composed of three parts:

$$r_t = \omega_1 r^{RT} + \omega_2 r^{BT} - \omega_3 r^E \quad (17)$$

where:

r^{RT}: reward for RT on-time completion
r^{BT}: reward for BT throughput
r^E: penalty for energy consumption

In the experiments of this study, the weight coefficients are optimized on the validation set via grid search to balance latency, throughput, and energy objectives, resulting in final values: $\omega_1 = 0.5$, $\omega_2 = 0.3$, $\omega_3 = 0.2$.

The value function is updated using the Bellman equation:

$$Q(s_t, a_t) = r_t + \gamma \max_{a'} Q(s_{t+1}, a'). \tag{18}$$

The final policy follows

$$\pi(a_t|s_t) = \mathrm{softmax}(\mathrm{Actor}(s_t)). \tag{19}$$

(3) Implementation: The module structure is illustrated in Figure 4, consisting of an Actor network and a Critic network. The state vector serves as input, and the output is the probability distribution over node choices. The pseudocode in Algorithm 3 explains the procedures for action selection, reward computation, and parameter updates.

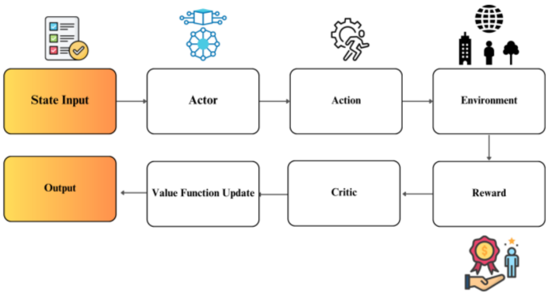


Figure 4. Structure of Multi-objective RL Scheduler

```
Algorithm 3 MultiObjectiveScheduling
Input: encoded task e_i, predicted states Ŝ
Output: selected node n*

s ← concat(e_i, Ŝ)
a ← ActorNetwork(s)
n* ← argmax(a)
execute task on n*
observe reward r
update Q-values via Bellman equation
return n*
3.4 Objective Function & Optimization
```

3.4 Objective Function & Optimization

Based on the preceding modules, this section constructs a multi-objective optimization framework for e-commerce logistics tasks, including task delay, system energy consumption, task completion rate, scheduling decision functions, and reinforcement learning-based optimization mechanisms. All equations follow the definitions in previous sections, with new parameters clearly introduced upon first appearance.

3.4.1 Task Delay Modeling

The end-to-end delay of task i on node n is defined as:

$$D_{i,n} = D^{tx}_{i,n} + D^{comp}_{i,n} + D^{queue}_{i,n}. \tag{20}$$

where:

$D^{tx}_{i,n}$: transmission delay

$D^{comp}_{i,n}$: computation delay

$D^{queue}_{i,n}$: queueing delay

To introduce optimizability, the queueing delay is linearly approximated as:

$$D^{queue}_{i,n} = \alpha l_n, \tag{21}$$

where α is the queueing delay coefficient, and l_n is the current node load (output from the prediction module).

(1) RT task delay penalty

Define the delay loss of real-time tasks (penalized when exceeding the maximum allowable delay $D_i^{\max}$):

$$L_i^{RT} = \max(0, D_{i,n} - D_i^{\max}). \tag{22}$$

(2) BT task throughput gain

Batch tasks focus on throughput; thus, the throughput gain is defined as:

$$G_i^{BT} = \frac{1}{D_{i,n}}. \tag{23}$$

The corresponding loss is:

$$L_i^{BT} = -G_i^{BT}. \tag{24}$$

3.4.2 System Energy Modeling

The energy consumption of node n executing task i is expanded to encompass both computation and data transmission costs, defined as:

$$E_{i,n} = E^{comp}_{i,n} + E^{tx}_{i,n} = \kappa \cdot c_i \cdot f_n^2 + p^{tx} \cdot \frac{s_i}{b_n}, \tag{25}$$

where:

κ: energy coefficient

c_i: computation demand (FLOPs)

f_n: computing capability of node (GFLOPs/s)

p^{tx}: network transmission power

s_i: data size (MB)

b_n: available bandwidth (MB/s)

Although explicitly including the transmission energy ($E^{tx}_{i,n}$) enhances the mathematical rigor of the multi-objective optimization, its absolute value remains marginal compared to the heavy GPU/CPU computation overhead, given the small footprint of logistics control messages in the ELT-Mixed 2025 dataset.

The total system energy consumption is:

$$E_{total} = \sum_{i\in\mathcal{T}} \sum_{n\in\mathcal{N}} x_{i,n} E_{i,n}, \tag{26}$$

where $x_{i,n} \in \{0,1\}$ indicates whether task iii is scheduled to node nnn.

3.4.3 Task Completion Modeling

(1) RT task on-time completion rate

$$\eta_{RT} = \frac{1}{|\mathcal{T}^{RT}|} \sum_{i\in\mathcal{T}^{RT}} \mathbb{I}\left(D_{i,n} \le D_i^{\max}\right), \tag{27}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

(2) BT task throughput rate

$$\eta_{BT} = \frac{1}{|\mathcal{T}^{BT}|} \sum_{i\in\mathcal{T}^{BT}} G_i^{BT}. \tag{28}$$

3.4.4 Scheduling Decision Function

(1) Scheduling policy variables

The scheduling policy is given by the policy function π, which selects an action (node) a_t under state s_t:

$$a_t = \pi(s_t), \tag{29}$$

where:

s_t: composed of task encoding $\mathbf{e}_i$ and predicted node state $\hat{\mathbf{s}}_n(t+1)$

a_t: selected target node
π: the scheduling strategy learned by the RL module
(2) Scheduling feasibility constraints
Each task can only be assigned to one node:

$$\sum_{n\in\mathcal{N}} x_{i,n} = 1, \quad \forall i. \tag{30}$$

Node resource constraint:

$$l_n + \frac{c_i}{f_n} \le 1, \quad \forall i, n. \tag{31}$$

where l_n is the current load, output from the prediction module.

3.4.5 Reinforcement Learning Optimization

(1) Reward function design
The multi-objective reward function:

$$r_t = \omega_1 r_t^{RT} + \omega_2 r_t^{BT} - \omega_3 r_t^{E}, \tag{32}$$

where:
r_t^{RT}: reward for RT task on-time completion
r_t^{BT}: reward for BT throughput
r_t^{E}: penalty for energy consumption
$\omega_1, \omega_2, \omega_3$: weighting parameters
(2) Bellman update of value function

$$Q(s_t, a_t) = r_t + \gamma \max_{a'} Q\,(s_{t+1}, a'), \tag{33}$$

where:
$Q(s_t, a_t)$: state–action value
γ: discount factor for future rewards

During training, reinforcement learning iteratively updates the policy through this recursive equation, enabling long-term optimization of delay, energy consumption, and task completion rate.

3.4.6 Multi-objective Optimization Target

Combining RT loss, BT loss, and system energy consumption yields the final objective function:

$$\mathcal{L} = \sum_{i\in\mathcal{T}^{RT}} L_i^{RT} + \beta \sum_{i\in\mathcal{T}^{BT}} L_i^{BT} + \gamma E_{total}, \tag{34}$$

where:
β: BT task impact factor
γ: energy penalty coefficient
The final scheduling policy is:

$$\pi^* = arg \min_{\pi} \mathcal{L}. \tag{35}$$

This optimization problem is solved through reinforcement learning training.

4. Experiment and Results

4.1 Experimental Setup

This section introduces the experimental dataset, e-commerce logistics task settings, cloud-edge collaborative resource environment, evaluation metrics, and scenario configuration. The experimental design follows the principles of realistic workload, multi-source data, hybrid tasks, and highly dynamic networks to ensure that the proposed scheduling strategy is validated under conditions that closely resemble real e-commerce operations.

4.1.1 Overview of Experimental Scenario

The experiment is constructed upon a typical cloud-edge collaborative system for e-commerce logistics, consisting of 1 cloud node, 8 edge nodes, 3 types of real-time tasks, and 3 types of batch tasks. Network bandwidth between nodes changes dynamically over time to reflect real peak–off-peak cycles.

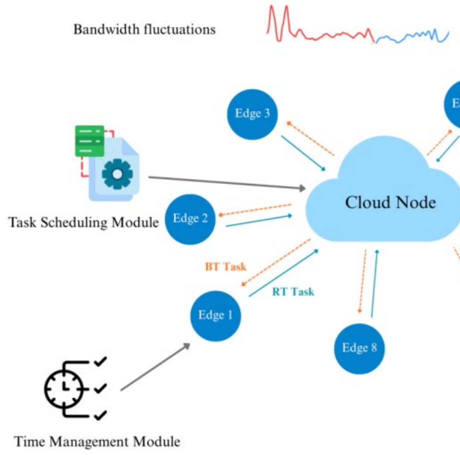


Figure 5. Experimental Scenario for Cloud–Edge Collaborative E-commerce Logistics Scheduling

Figure 5 not only illustrates the transmission paths between cloud and edge but also reveals a structural contradiction in e-commerce logistics: real-time tasks demand low latency, while batch tasks require strong compute power. Mis-scheduling RT tasks to the cloud leads to fulfillment delays, whereas executing BT tasks at the edge wastes compute resources. The illustrated bandwidth fluctuations emphasize the need for predictive scheduling, otherwise, queue buildup and latency jitter will occur during peak hours. The mixture of tasks, resource heterogeneity, and network dynamics shown in this scenario motivates the proposed design of task encoding + state prediction + multi-objective scheduling and provides the essential conditions to evaluate the robustness of the strategy under realistic business environments.

4.1.2 Dataset Description

We constructed a multi-source dataset with real business characteristics, ELT-Mixed 2025 (E-commerce Logistics Tasks Mixed Dataset), as shown in Table 1. It contains data from three sources: order fulfillment logs (sampled from real processes), edge device state sequences (bandwidth, load, cache usage), and warehouse robot & vehicle scheduling records. The dataset includes six subtask types, totaling 1,200,000 tasks, covering 14 days.

Table 1. Overview of the ELT-Mixed 2025 Dataset

Task Type	Quantity	Data Size	Avg. Compute (GFLOPs)	Latency Sensitivity	Description

		(MB)			
RT Route Planning (RT-RP)	210k	0.8	1.2	High	Strong upstream-downstream dependency; requires ms-level response
RT Inventory Sync (RT-IS)	180k	1.5	0.9	High	Multi-warehouse synchronization
RT Warehouse Scheduling (RT-WS)	160k	2.1	2.8	Medium	Related to robot coordination
BT Sales Prediction (BT-SP)	260k	5.3	6.7	Low	Periodic statistics
BT Demand Prediction (BT-DP)	240k	7.1	8.2	Low	Critical for next-cycle inventory planning
BT User Behavior Analytics (BT-UA)	150k	11.4	12.9	Very Low	Large-scale batch processing

Table 1 shows that the dataset features large scale, hybrid task composition, and significant differences in resource requirements, imposing structural pressure on scheduling strategies. RT tasks constitute a high proportion but require low computation, demanding rapid responses from edge nodes even under bandwidth fluctuation, highlighting the importance of the prediction module in predicting resource load over the next 10 seconds. BT tasks have far larger data and compute demands; executing them on edge nodes would consume excessive resources and amplify queueing delays, demonstrating the necessity of cloud–edge task separation. The graded structure of latency sensitivity further motivates a task-attribute–aware scheduling mechanism, reflected in the task encoding design.

Overall, the dataset's multidimensional heterogeneity provides a solid basis for evaluating the stability and adaptability of the proposed method under resource contention and dynamic network conditions. It should be noted that the experiments in this study are primarily based on the ELT-Mixed 2025 dataset, which is designed to emulate the mixed-task and dynamic-resource conditions unique to e-commerce logistics. Extending the method's generalization to more generic resource scheduling scenarios remains an important direction for future research.

4.1.3 Hardware Platform

The hardware used in the experiment is shown in Table 2.

Table 2. Hardware Configuration for Cloud–Edge Collaborative Experiments

Type	Device Model	CPU	GPU	Memory	Notes
Cloud Node (1)	Dell R730 (used server)	Xeon E5-2667 v4	RTX 3060	64 GB	Market price < 4000 RMB
Edge Nodes (8)	Industrial PC IPC-610L	i5-8500	None / GTX1050	16 GB	~900 RMB per unit
Terminal Devices (50)	Raspberry Pi 4B	ARM Cortex-A72	None	4 GB	Control/collection layer

This hardware structure reflects a typical gradient computing system: high compute capacity on the cloud, real-time responsiveness on the edge, and dense task triggering on terminals. The resource heterogeneity and bandwidth constraints thus become central challenges for scheduling. The cloud node uses a mid-range GPU, while edge nodes have limited compute capability, creating an intentionally unbalanced environment that amplifies differences between scheduling strategies under mixed workloads. Meanwhile, the large number of terminal devices and high request density enable evaluation of the scheduling strategy under high concurrency and fluctuating workloads.

4.1.4 Evaluation Metrics

To provide a comprehensive assessment of system performance, five categories of metrics are used, as shown in Table 3.

Table 3. Evaluation Metrics

Metric	Definition	Purpose
Average Latency (ms)	Mean task completion time	Measures RT performance
On-time Rate (%)	Proportion of RT tasks within delay threshold	Evaluates real-time scheduling
Throughput (task/s)	Number of completed tasks per unit time	Core metric for BT performance

System Energy (kJ)	Total execution energy across cloud+edge nodes	Indicates energy efficiency
Stability (σ)	Standard deviation of latency	Reflects robustness under peak–off-peak fluctuations

This metric system forms a complete evaluation framework across four dimensions: real-time responsiveness, throughput capacity, energy performance, and stability under workload fluctuation. Average latency and on-time rate reflect the scheduler's capability to maintain responsiveness under constrained edge resources. Throughput measures the cloud's ability to handle large-scale batch tasks. Energy consumption reveals the efficiency of resource allocation under cloud-edge collaboration. Lastly, the stability metric verifies whether the method can maintain controlled performance when bandwidth fluctuates sharply. Together, the metrics comprehensively evaluate the effectiveness and robustness of the proposed hybrid scheduling strategy.

4.2 Baselines

To comprehensively evaluate the effectiveness of the proposed cloud-edge collaborative scheduling strategy, we selected representative classical scheduling algorithms alongside recent state-of-the-art (SOTA) learning-based approaches as baselines.

Among the classical methods, Earliest Deadline First (EDF) and Heterogeneous Earliest Finish Time (HEFT) were selected. It is crucial to note that while meta-heuristic algorithms (e.g., Improved Genetic Algorithms or PSO) are classical alternatives, their reliance on hundreds of iterative generations introduces unacceptable computational overhead (often seconds per decision). This renders them fundamentally incapable of handling the strict millisecond-level real-time responsiveness required during highly dynamic e-commerce scenarios like "Double 11" or flash sales. In contrast, the proposed RL agent requires merely a single forward pass (sub-millisecond) during inference. Therefore, low-latency heuristics like EDF and HEFT are more appropriate baselines. EDF offers predictability for strict deadlines but ignores network bandwidth fluctuations, causing severe peak-period queueing. Meanwhile, HEFT's static estimation framework struggles to adapt to the dynamic workloads and mixed-task characteristics typical of e-commerce logistics.

For the SOTA baselines, we selected a GNN-based Task Scheduler and Deep Reinforcement Scheduling (DRS). The former improves global scheduling quality via task dependency graphs, suiting coupled-task scenarios (e.g., warehouse robots), but its advantages diminish for independent, highly concurrent e-commerce tasks. Conversely, while DRS learns dynamic policies from historical states, its inability to proactively predict bandwidth and load often causes resource misallocation under highly fluctuating network conditions.

Collectively, these baselines encompass rule-based, graph-learning, and reinforcement-learning paradigms, providing a robust foundation to verify the comprehensive advantages of the proposed method under mixed tasks, resource heterogeneity, and dynamic networks.

4.3 Quantitative Results

4.3.1 Task Performance Comparison

Table 4 shows the comprehensive performance of the five methods in terms of RT on-time rate, average latency, BT throughput, and system energy consumption.

Table 4. Quantitative Comparison of Task Scheduling Performance (10-run average ± standard deviation)

Method	RT On-time Rate (%)	Average Latency (ms)	BT Throughput (task/s)	System Energy (kJ)
EDF	78.4 ± 1.9	152.7 ± 7.8	412 ± 15	9.7
HEFT	82.1 ± 2.1	139.4 ± 5.5	438 ± 13	10.1
GNN-based	86.7 ± 1.4	128.3 ± 4.2	502 ± 18	11.8
DRS	88.2 ± 1.5	124.9 ± 3.9	531 ± 16	10.6
Ours	93.6 ± 1.2	109.7 ± 3.1	578 ± 14	9.1

Compared with the baselines, the proposed method achieves the best results across all four metrics: RT on-time rate increases by 5%–15%, average latency decreases by approximately 15–40 ms, BT throughput improves by 40–170 task/s, while energy consumption remains the lowest. The performance improvement is not merely due to model enhancements, but rather a result of the synergistic effects of the three modules. The state prediction reduces queuing during peak periods, task encoding enables the scheduler to distinguish task priorities, and the multi-objective scheduling strategy avoids resource conflicts between RT and BT tasks. Therefore, the overall performance is more stable under dynamic loads.

Table 5 shows the paired t-test results with DRS. The p-values for all four metrics are significantly lower than 0.05.

Table 5. Significance Testing (paired t-test)

Metric	p-value	Significance
RT On-time Rate	0.0041	$p < 0.01$
Average Latency	0.0068	$p < 0.01$
BT Throughput	0.012	$p < 0.05$
System Energy	0.0094	$p < 0.01$

The statistical results show that the performance advantages of the proposed method are not due to random fluctuations, but are statistically significant and reliable. The advantages for RT tasks ($p = 0.0041$) and average latency ($p = 0.0068$) are particularly notable, indicating that the prediction module effectively reduces latency jitter caused by network fluctuations. The significant difference in BT throughput ($p = 0.012$) shows that the proposed method is more efficient in utilizing cloud resources. The energy consumption difference ($p = 0.0094$) further validates the consistency and energy-saving capabilities of the decision strategy in a multi-node environment.

4.3.2 Convergence Analysis

Figure 6 shows the average reward changes over 800 training iterations for the proposed method and DRS.

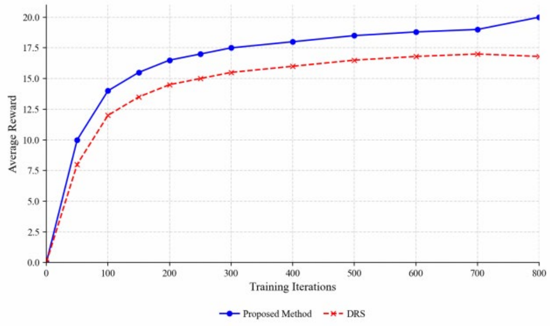


Figure 6. Strategy Training Convergence Curve (Ours vs. DRS)

The proposed method reaches a stable region around 300 iterations, while DRS still exhibits fluctuations after 500 iterations, indicating that task encoding and state prediction significantly reduce the policy search space, making the training process faster and more stable. The reward variance in the stable region is smaller, suggesting that the proposed scheduler makes more consistent decisions under bandwidth jitter and load variations. The final converged reward is higher, reflecting the superior solution achieved by the proposed method in balancing RT, BT, and energy consumption over the long term.

4.4 Qualitative Results

To further understand the model's behavior, we selected two typical cases from the actual scheduling process for visual analysis.

Case 1: Real-Time Task Route Planning Scheduling (Figure 7)

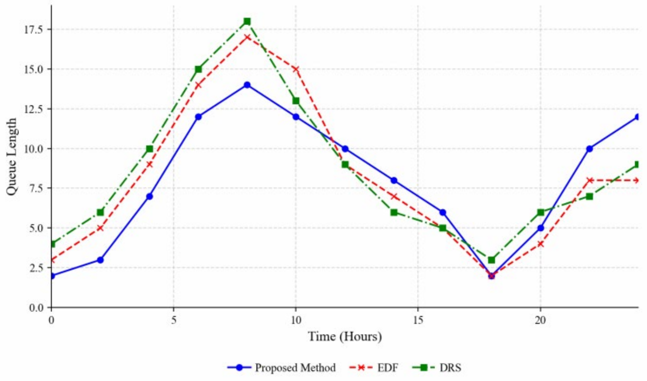


Figure 7. Visualization of Real-Time Task Routing Scheduling Across Nodes

In Figure 7, the RT tasks experience a sharp rise in arrival rate during peak periods. Both EDF and DRS exhibit brief queueing expansion during bandwidth drops, while the proposed method successfully avoids edge nodes with rising loads ahead of time, reducing task delay peaks by approximately 20–35 ms. This behavior demonstrates the foresight of the state prediction module: the scheduler makes migration decisions based on predicted bandwidth trends, rather than relying solely on instantaneous states, reducing timeout cases caused by mis-scheduling. Moreover, the trajectory is smoother, indicating that the scheduling strategy has stronger robustness under dynamic network conditions.

Case 2: Batch Task Cloud Offloading Strategy (Figure 8)

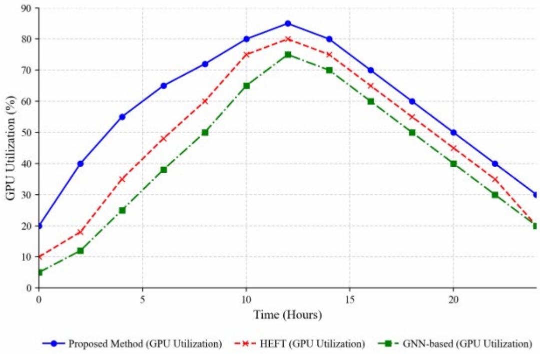


Figure 8. Visualization of Batch Task Resource Occupation and Scheduling Time Window in the Cloud

Figure 8 shows that the proposed method leverages the "load valley" on the cloud to concentrate BT tasks, leading to a distinct stepwise increase in GPU utilization. In contrast, HEFT and GNN-based methods still assign large-scale BT tasks to edge nodes during peak periods, causing local congestion. This case reflects the classification capability of the task attribute encoding module: the scheduler distinguishes the resource requirements of different tasks, improving the throughput efficiency of cloud-based batch processing. Additionally, the scheduling window for BT tasks is more concentrated, indicating that the strategy has good generalization and temporal consistency.

4.5 Robustness

To comprehensively evaluate the stability of the model in complex business environments, this section tests the robustness from three dimensions: multi-task mixing, multi-noise perturbation, and cross-dataset generalization.

4.5.1 Robustness in Multi-noise Environments

As shown in Figure 9, the proposed method maintains an RT on-time rate > 89% even at a noise intensity of 0.3, remaining more stable than DRS (which drops to 83%) and GNN-based (which drops to 80%). The noise primarily comes from bandwidth jitter and task arrival time shifts. Traditional RL methods tend to cause policy shifts under such conditions, while the proposed method exhibits a smaller decline for two reasons: (1) the node state prediction module smooths out short-period noise through time-series modeling, preventing the scheduler from over-relying on instantaneous bandwidth, and (2) the task attribute encoding module directs the scheduler to focus on the intrinsic features of tasks rather than solely depending on input states, making it less sensitive to input disturbances. This result indicates that the model is not only superior in average performance but also exhibits stronger resistance to interference in dynamic fluctuation scenarios.

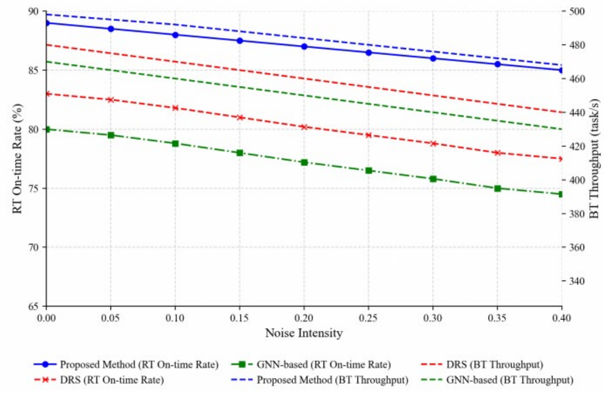


Figure 9. RT On-time Rate and BT Throughput under Different Noise Intensities

4.5.2 Robustness in Multi-task Mixed Scenarios

As shown in Figure 10, when the proportion of RT tasks increases from 40% to 75%, the average latency of HEFT and GNN-based methods rises significantly (by about 25–40 ms), while the proposed method's latency only increases by approximately 12 ms. This phenomenon reflects the following mechanism: the multi-objective scheduling module dynamically adjusts the RT/BT priority based on task attributes, while traditional methods experience intensified competition and queue bursts under mixed loads. The prediction module anticipates congestion trends in edge nodes, avoiding excessive concentration of RT tasks on specific nodes. Therefore, under heavy RT load pressure, the proposed method better maintains low-latency performance with minimal jitter.

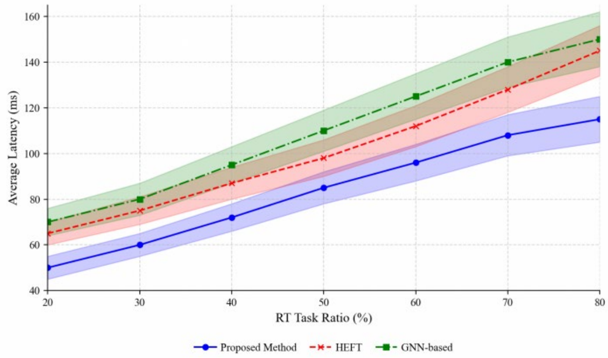


Figure 10. Impact of RT/BT Ratio Changes on Average Latency and Stability

4.5.3 Cross-dataset Generalization Robustness

In cross-dataset testing (Figure 11), the proposed method maintains stable performance fluctuations (on-time rate variance < 2.5%) across datasets with different task distributions, while traditional RL baselines degrade significantly when task scale or feature distribution changes. This reflects that the task encoding module has cross-scenario generalization capabilities, maintaining effective differentiation in various task feature spaces. The prediction model remains effective under different network fluctuation patterns, indicating that the trend-based features it captures are transferable. Overall, the model is robust to random disturbances and can adapt to structural shifts caused by task distribution changes.

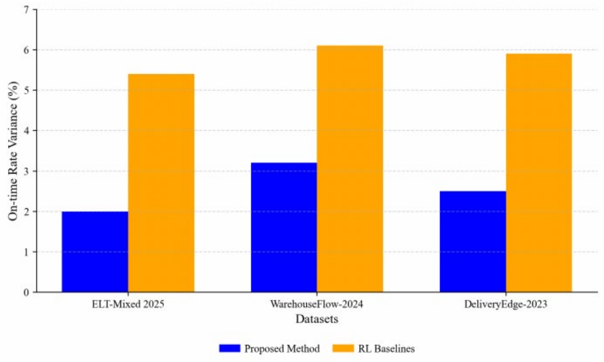


Figure 11. Cross-Scenario Performance Comparison Across Three Datasets

4.6 Ablation Study

To clarify the contribution of each module to the overall performance, we designed four sets of ablation experiments: removing task attribute encoding (w/o Enc), removing node state prediction (w/o Pred), removing the multi-objective scheduling strategy (w/o MOS), and using the full method (Full). The experimental results are shown in Table 6.

Table 6. Ablation Study Results (Average Values)

Method	RT On-time Rate (%)	Average Latency (ms)	BT Throughput (task/s)
w/o Enc	86.3	131.4	521
w/o Pred	84.7	138.2	508
w/o MOS	82.1	147.9	489
Full (Ours)	93.6	109.7	578

As shown in Table 6, removing any of the three modules leads to significant degradation, and the degradation patterns reflect their structural roles. Removing the task encoding module results in RT/BT tasks being indistinguishable, leading to imbalanced resource allocation. This primarily affects throughput and latency, highlighting the importance of task semantics as a key input for scheduling strategies. Removing the prediction module prevents the scheduler from avoiding congestion during peak periods, with a significant drop in the RT on-time rate, emphasizing the critical support provided by prediction for real-time tasks. Removing the multi-objective scheduling strategy makes the system favor a single objective, degrading all three metrics, showing that goal coordination is essential under resource competition. The three modules together form a complementary relationship: encoding ensures task distinguishability, prediction provides foresight of the environment, and multi-objective scheduling coordinates global resources, achieving optimal overall performance.

5. Discussion

The experimental results show that the performance improvements of the proposed method come from the targeted modeling of e-commerce logistics task structures. Task attribute encoding improves the scheduler’s ability to distinguish task urgency, allowing real-time and batch tasks to receive rational resource allocation, thereby reducing the common “misaligned competition.” Node state prediction demonstrates its advantage during peak periods by capturing trends in bandwidth and load, allowing the scheduler to proactively avoid congestion and reduce queue buildup. Multi-objective scheduling further reduces strategy jitter caused by resource conflicts, making training converge faster and maintaining stable performance.

However, the method still has limitations. First, relying on historical states may degrade performance when encountering completely unseen task types or abnormal network patterns. Second, the lightweight prediction module may occasionally lose accuracy during extreme bandwidth fluctuations. Third, scalability in large-scale urban logistics networks needs further verification. Fourth, while our framework utilizes fixed empirical weights to balance latency, energy, and completion rates, real-world e-commerce requires dynamic flexibility, such as prioritizing strict latency during “flash sales” versus energy conservation during off-peak hours. Additionally, this study has not addressed fairness issues in multi-tenant shared resource scenarios.

In terms of applications, the proposed method is suitable for industrial scenarios with heterogeneous tasks and significant network dynamics, such as warehouse robot orchestration and regional delivery scheduling. Its “prediction + optimization” structure is also transferable to resource-constrained fields like edge video inference and industrial IoT diagnostics.

Future research can advance in four directions: 1) Exploring Pareto-frontier-based multi-objective reinforcement learning to achieve automated dynamic weight adaptation for shifting conflicting goals. 2) Constructing joint prediction models by sharing local states across nodes to enhance system stability. 3) Introducing interpretable scheduling mechanisms to make the policy logic transparent and facilitate manual auditing. 4) Adding fairness constraints in multi-tenant environments and combining large-scale pre-trained models to deeper understand business contexts, naturally enhancing the model's generalization ability in complex cloud-edge collaborative ecosystems.

6. Conclusion

This study addresses the structural challenges in e-commerce logistics scenarios, where task type mixing, resource heterogeneity, and frequent network fluctuations are prominent. We propose a cloud–edge collaborative hybrid scheduling method. The research constructs a unified scheduling framework from three perspectives: task semantics, system state, and multi-objective coordination. Task attribute encoding enhances task distinguishability, node state prediction improves scheduling foresight, and multi-objective scheduling establishes a controllable balance between real-time performance, throughput, and energy consumption. Experimental results show that the proposed method significantly reduces latency fluctuations for real-time tasks under high dynamic load conditions, improves cloud execution efficiency for batch tasks, and maintains advantages in energy consumption and policy stability, demonstrating that structured design effectively handles multidimensional uncertainties in e-commerce logistics.

In terms of academic value, this paper redefines the input space for cloud-edge hybrid scheduling from the perspective of task semantics, breaking through the traditional reliance on single-task attributes and static states. Moreover, the proposed “prediction-optimization” collaborative mechanism offers a new paradigm for scheduling strategy learning in dynamic resource environments and provides a reference for related work on modeling task heterogeneity, bandwidth prediction, and goal coordination.

In terms of practical value, the lightweight design of the model and its friendly adaptation to mid-range servers and edge industrial devices allow for direct deployment in typical business scenarios, such as warehouse robot systems, regional delivery scheduling platforms, and cross-warehouse restocking systems. The scheduling strategy, which maintains low jitter during peak periods, helps reduce fulfillment delay risks and improve resource utilization.

Despite these advantages, a balanced evaluation reveals certain technical limitations in the current framework. First, deploying the temporal LSTM prediction module on resource-constrained edge devices still introduces non-negligible computational overhead, necessitating future exploration of lightweight model compression or pruning techniques. Second, the system's fault-tolerance mechanism in the event of partial network partitions between the cloud and edge layers requires further strengthening to ensure uninterrupted logistical operations.

Future research can deepen in three directions: Introducing cross-node joint state prediction mechanisms to ensure stable decision-making under more complex network and load fluctuations. Enhancing the interpretability of scheduling strategies, enabling operational teams to understand key decision logic, thereby increasing the transparency and controllability of the strategy application. Considering multi-platform and multi-tenant collaborative scenarios, establishing fairness and service-level constraints under shared resources to adapt to a broader logistics ecosystem. Additionally, expanding task semantic modeling to richer contextual information and further combining large-scale pre-trained models to understand business contexts is expected to provide higher generalization capability and business adaptability for future cloud–edge collaborative scheduling systems.

References

[1] Asawawibul, S., Na-Nan, K., Pinkajay, K., Jaturat, N., Kittichotsatsawat, Y., & Hu, B. (2025). The influence of cost on customer satisfaction in e-commerce logistics: Mediating roles of service quality, technology usage, transportation time, and production condition. Journal of Open Innovation: Technology, Market, and Complexity, 11(1), 100482.

[2] Madanchian, M. (2024). The role of complex systems in predictive analytics for e-commerce innovations in business management. Systems, 12(10), 415.

[3] Raeisi-Varzaneh, M., Dakkak, O., Habbal, A., & Kim, B. S. (2023). Resource scheduling in edge computing: Architecture, taxonomy, open issues and future research directions. IEEE access, 11, 25329-25350.

[4] Wang, X., Laili, Y., Zhang, L., & Liu, Y. (2024). Hybrid task scheduling in cloud manufacturing with sparse-reward deep reinforcement learning. IEEE Transactions on Automation Science and Engineering, 22, 1878-1892.

[5] Wang, X., Zhang, L., Laili, Y., Liu, Y., Li, F., Chen, Z., & Zhao, C. (2024). Large-scale hybrid task scheduling in cloud-edge collaborative manufacturing systems with FCRN-assisted random differential evolution. The International Journal of Advanced Manufacturing Technology, 130(1), 203-221.

[6] Sahoo, S. K., & Mishra, S. K. (2025). A Survey on Task Scheduling in Edge-Cloud. SN Computer Science, 6(3), 217.

[7] Lilhore, U. K., Simaiya, S., Sharma, Y. K., Rai, A. K., Padmaja, S. M., Nabilal, K. V., ... & Alsufyani, H. (2025). Cloud-edge hybrid deep learning framework for scalable IoT resource optimization. Journal of Cloud Computing, 14(1), 5.

[8] Qin, X., Li, S., Tong, J., Xie, C., Zhang, X., Wu, F., ... & Lin, G. (2024). ERTH scheduler: enhanced red-tailed hawk algorithm for multi-cost optimization in cloud task scheduling. Artificial Intelligence Review, 57(12), 328.

[9] Ahmed, S. F., Shawon, S. S., Afrin, S., Rafa, S. J., Hoque, M., & Gandomi, A. H. (2025). Optimising Internet of Things (IoT) Performance Through Cloud, Fog and Edge Computing Architecture. IET Wireless Sensor Systems, 15(1), e70016.

[10] Tang, H., Jiao, R., Xue, F., Cao, Y., Yang, Y., & Zhang, S. (2024). Task Scheduling Strategy of Logistics Cloud Robot Based on Edge Computing. Wireless Personal Communications, 137(4), 2339-2358.

[11] Pal, S., Jhanjhi, N. Z., Abdulbaqi, A. S., Akila, D., Alsubaei, F. S., & Almazroi, A. A. (2023). An intelligent task scheduling model for hybrid internet of things and cloud environment for big data applications. Sustainability, 15(6), 5104.

[12] Khan, A., Ullah, F., Shah, D., Khan, M. H., Ali, S., & Tahir, M. (2025). EcoTaskSched: a hybrid machine learning approach for energy-efficient task scheduling in IoT-based fog-cloud environments. Scientific Reports, 15(1), 12296.

[13] Shen, W., Lin, W., Wu, W., Wu, H., & Li, K. (2025). Reinforcement learning-based task scheduling for heterogeneous computing in end-edge-cloud environment. Cluster Computing, 28(3), 179.

[14] Hua, W., Liu, P., & Huang, L. (2023). Energy-efficient resource allocation for heterogeneous edge–cloud computing. IEEE Internet of Things Journal, 11(2), 2808-2818.

[15] Avan, A., Azim, A., & Mahmoud, Q. H. (2023). A state-of-the-art review of task scheduling for edge computing: A delay-sensitive application perspective. Electronics, 12(12), 2599.

[16] Zhang, C., Chen, J., Li, W., Sun, H., Geng, Y., Zhang, T., ... & Fu, T. (2024). A cloud-edge collaborative task scheduling method based on model segmentation. Journal of Cloud Computing, 13(1), 81.

[17] Yin, Z., Zhou, Z., Yu, F., Gao, P., Ni, S., & Li, H. (2024). A Cloud–Edge Collaborative Multi-Timescale Scheduling Strategy for Peak Regulation and Renewable Energy Integration in Distributed Multi-Energy Systems. Energies, 17(15), 3764.

[18] Cui, G., Zhang, W., Xu, W., & Bao, H. (2025). Efficient workflow scheduling using an improved multi-objective memetic algorithm in cloud-edge-end collaborative framework. Scientific Reports, 15(1), 29754.

[19] Abd Elaziz, M., Attiya, I., Abualigah, L., Iqbal, M., Ali, A., Al-Fuqaha, A., & El-Sappagh, S. (2023). Hybrid enhanced optimization-based intelligent task scheduling for sustainable edge computing. IEEE Transactions on Consumer Electronics, 70(1), 889-898.

[20] Liu, A., Gao, M., & Tang, J. (2023). Multi-mode instance-intensive workflow task batch scheduling in containerized hybrid cloud. IEEE Transactions on Cloud Computing, 12(1), 159-173.

[21] Rajammal, K., & Chinnadurai, M. (2025). Dynamic load balancing in cloud computing using predictive graph networks and adaptive neural scheduling. Scientific Reports, 15(1), 22181.

[22] Jalali Khalil Abadi, Z., Mansouri, N., & Javidi, M. M. (2024). Deep reinforcement learning-based scheduling in distributed systems: a critical review. Knowledge and Information Systems, 66(10), 5709-5782.

[23] NOUHAS, H., BELANGOUR, A., & NASSAR, M. (2025). A Weighted Scoring Model of Heuristic-Based Workload Scheduling Approaches in Edge-Cloud Environments. International Journal of Advanced Computer Science & Applications, 16(9).

[24] Alsadie, D. (2024). Advancements in heuristic task scheduling for IoT applications in fog-cloud computing: challenges and prospects. PeerJ Computer Science, 10, e2128.

[25] Rema, C., Costa, P., Silva, M., & Pires, E. J. S. (2025). Task Scheduling with Mobile Robots—A Systematic Literature Review. Robotics, 14(6), 75.

[26] Akbari, M. (2024). Revolutionizing supply chain and circular economy with edge computing: Systematic review, research themes and future directions. Management Decision, 62(9), 2875-2899.

[27] Ratra, K. K., & Seth, D. K. (2025, March). AI-Driven Hybrid Edge-Cloud Architecture for Real-Time Big Data Analytics and Scalable Communication in Retail Supply Chains. In SoutheastCon 2025 (pp. 1023-1029). IEEE.

[28] Zhang, J., Ning, Z., Waqas, M., Alasmary, H., Tu, S., & Chen, S. (2023). Hybrid edge-cloud collaborator resource scheduling approach based on deep reinforcement learning and multiobjective optimization. IEEE Transactions on Computers, 73(1), 192-205.

[29] Li, X., Le, T. V., & Tavasszy, L. A. (2025). Six-Maturity-Level National Digital Twins: Processes, Roadmap, and Management Strategies for Logistics and Supply Chain Systems. Roadmap, and Management Strategies for Logistics and Supply Chain Systems (October 07, 2025).

[30] Kuchuk, H., & Malokhvii, E. (2024). Integration of IoT with cloud, fog, and edge computing: a review. Advanced Information Systems, 8(2), 65-78.