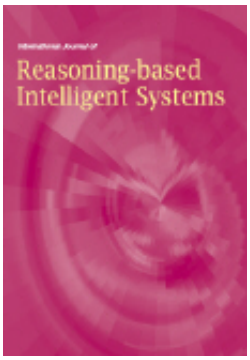


International Journal of Reasoning-based Intelligent Systems

ISSN online: 1755-0564 - ISSN print: 1755-0556
https://www.inderscience.com/ijris

Elastic allocation and recovery of airport computing resources in response to strong disturbances in flight flows

Wenliang Wang

DOI: 10.1504/IJRIS.2026.10079293

Article History:

Received:	01 April 2026
Last revised:	18 May 2026
Accepted:	18 May 2026
Published online:	22 July 2026

Elastic allocation and recovery of airport computing resources in response to strong disturbances in flight flows

Wenliang Wang

Management College,
Guangzhou Civil Aviation College,
Guangzhou, 510403, China
Email: rachel9876@sina.cn

Abstract: To address the problems of slow response and large-scale computational power collapse of airport computing nodes under strong disturbances, this paper proposes a novel computing resource joint scheduling system with dynamic perception and elastic recovery capabilities. This system first constructs a network graph structure covering flight status changes and physical space distribution, and then uses a reinforcement learning algorithm that allows computers to continuously try and error to learn the optimal allocation strategy. Simulation experiments show that in the face of extreme flight flow disturbances, this system not only reduces the overall processing delay of computing tasks by 14.2%, but also shortens the adaptive recovery time of the overall computing resource allocation from an average of 28 seconds to 22 seconds after some computing nodes temporarily fail due to overload, effectively ensuring the continuous operation of core business data of the airport under adverse conditions.

Keywords: flight flow disturbance; computational resource allocation; multi-agent reinforcement learning; spatio-temporal graph network.

Reference to this paper should be made as follows: Wang, W. (2026) 'Elastic allocation and recovery of airport computing resources in response to strong disturbances in flight flows', *Int. J. Reasoning-based Intelligent Systems*, Vol. 18, No. 17, pp.48–66.

Biographical notes: Wenliang Wang is a Lecturer at the Guangzhou Civil Aviation College, China. She received her Master's degree from the Sun Yat-sen University (2014) in China. Her research interests include civil aviation travel and transportation, computational resource allocation, and smart civil aviation.

1 Introduction

With the rapid development of the global aviation industry, the informatisation and intelligence level of large international airports have been continuously improving, and their daily operations are highly dependent on the underlying computing resource networks (Zhang et al., 2024a). However, in actual operation, airports are highly susceptible to external factors such as extreme weather, air traffic control, or sudden public events, resulting in 'strong disturbances in flight flow'. Different from daily peak fluctuations, strong disturbance refers to the instantaneous surge in traffic caused by extreme weather and so on, which has the characteristics of nonlinearity and cascading spread. In such strong disturbance scenarios, large-scale flight delays, diversions, or cancellations will lead to a massive number of re-scheduling and collaborative computing requests for various airport information systems. At this time, the underlying computing network of the airport will face an instantaneous traffic peak (Yang et al., 2026). How to efficiently schedule computing resources in this extreme and highly dynamic environment to ensure the continuity of core business and real-time processing of data has become a key issue that needs to be urgently addressed in the construction of modern smart airports (Ding et al., 2022).

Although the existing computing resource scheduling technologies perform well in normal and stable operation conditions, they exhibit significant limitations when dealing with strong disturbances in flight flow. On the one hand, traditional static resource allocation models are usually preset based on historical average load, lacking the flexibility to cope with instantaneous sudden computing power demands. This rigid allocation is prone to causing some computing nodes to crash due to overload, while other physical areas' nodes are idle, resulting in a serious mismatch of resources (Liu et al., 2022). On the other hand, the existing system fault recovery mechanisms mostly adopt passive rule-triggering methods (such as simple restart or queuing waiting), and their responses have serious high latency problems when facing large-scale, cascading depletion of computing power caused by strong disturbances. This rigid and lagging resource management model not only fails to meet the real-time requirements of airport operations under large-scale delays, but is more likely to further induce scheduling paralysis (Sujkowski et al., 2023).

Given the limitations of the existing methods, this paper aims to develop an intelligent decision-making framework that combines elastic allocation and dynamic recovery capabilities. This framework is dedicated to real-time perception of the evolution trends of flight flow disturbances in both the physical space and time dimensions, and establishes a dynamic mapping relationship between massive computing tasks and limited distributed edge nodes. The core objective of this research is to break the traditional passive scheduling paradigm and achieve adaptive pre-allocation of airport computing resources under strong disturbances, and provide low-latency cross-node task migration and system recovery solutions when local nodes inevitably experience overload or failure, thereby fundamentally improving the high availability and strong robustness of the entire computing network.

Existing literature predominantly frames edge resource allocation as a deterministic or near-stationary engineering optimisation problem, primarily focusing on minimising computational latency or energy consumption. However, this conventional paradigm overlooks a critical scientific gap in the context of extreme aviation disruptions: the fundamental decoupling between physical environmental dynamics and digital computing topologies.

To address these challenges, this paper constructs a spatio-temporal graph network to quantify disturbance propagation, combined with multi-agent reinforcement learning (MARL) scheduling and an active recovery strategy. Experiments on real airport data demonstrate that under extreme disturbances, this approach reduces computing latency by 14.2% and shortens network recovery time from 28s to 22s. The main contributions are:

- EAR-Sys framework: the first cross-space integration of physical flight flow disturbances and virtual computing resource scheduling.
- Graph reinforcement learning (RL) joint optimisation: a novel mechanism to overcome dimensionality and latency bottlenecks in dynamic, large-scale computing allocation.
- Active dynamic recovery: a priority-based, multi-node collaborative migration strategy that replaces traditional passive recovery to enhance system resilience.

The remainder of this paper is organised as follows: Section 2 reviews related literature. Section 3 details the EAR-Sys architecture and algorithms. Section 4 describes the experimental setup. Section 5 presents the results and analysis. Section 6 discusses underlying mechanisms and limitations. Section 7 concludes the paper.

2 Literature review

This section systematically reviews the core literature related to the computing resource management of smart airports and the evolution of flight flows. Through in-depth discussions on flight flow perturbation modelling, traditional resource allocation methods, elastic recovery mechanisms, and the application of RL in resource scheduling, the progress of existing research is clarified, and the key limitations of current technologies in dealing with extreme strong perturbation scenarios are summarised.

2.1 Flight flow perturbation modelling

The prediction and modelling of flight flow disturbances are the prerequisite for understanding the sudden demand for computing power at airports. Early studies mainly relied on time series analysis and macroscopic statistical models to assess the overall impact of extreme weather or air traffic control on flight delays. In recent years, with the development of complex network theory, scholars have begun to abstract airports and routes as graph structures and use spatial-temporal graph (STG) networks and complex system dynamics to simulate the cascading failure process of disturbances in the aviation network (Wu et al., 2023). Some studies have introduced graph convolutional neural (GCN) and long short-term memory (LSTM) to capture the evolution characteristics of flight flow in the spatial topology and time dimension (Ju et al., 2025). However, most existing flight flow disturbance models are designed for flight rearrangement from the perspective of air traffic control or ground support vehicle scheduling, and few studies have precisely mapped the strong disturbance characteristics generated in physical space (such as runways, boarding gates) to the instantaneous computing load changes of airport IT infrastructure in the digital space (Zhang et al., 2022).

2.2 Airport resource allocation

In the area of airport computing resources and edge node scheduling, traditional methods mainly focus on efficiency optimisation in static or quasi-static environments (Wei et al., 2026). With the explosive growth of IoT devices in smart airports, the computing architecture is gradually shifting from centralised cloud computing to a 'cloud-edge-end' collaborative model (Mou et al., 2024). Existing resource allocation research mostly adopts mixed integer linear programming (MILP) or queuing theory to construct mathematical models and uses heuristic methods such as genetic algorithms (GA), particle swarm optimisation (PSO), or ant colony algorithms for solution (Zhang et al., 2024b). These traditional methods can effectively reduce system energy consumption and computing latency under normal flight density. However, heuristic algorithms usually rely on multiple rounds of iterative optimisation. When facing a sudden increase in computing power demand caused by extreme flight flow disturbances, their high computational time costs often make them unable to meet the millisecond-level response requirements of airport core business (Zhao et al., 2025). Moreover, static allocation models lack dynamic perception of system state changes, which is prone to causing local critical computing nodes to be overloaded and paralysed under the impact of traffic peak flows.

2.3 Elastic recovery mechanisms

For the resilience of the system and the recovery of computing resources in abnormal situations, a series of strategies have been proposed in the field of distributed computing. The existing elastic recovery mechanisms mainly include redundancy backup (replication), checkpoint rollback, and task migration (Wan et al., 2024). In the edge computing scenario, most studies adopt passive recovery strategies based on thresholds, that is, when the CPU utilisation or memory usage of a certain computing node reaches a dangerous level, or even when the node has failed, the system only triggers an alarm and suspends the tasks and migrates them to the adjacent node (Tong et al., 2022). Although these strategies have certain effects in dealing with occasional single-point failures, in the face of a large-scale, strong-correlated computing surge caused by strong disturbances in flight flow, passive recovery often leads to a 'snowball' effect – the tasks of overloaded nodes flow to the already heavily loaded neighbouring nodes, thereby causing a chain collapse of the entire airport edge computing network (Diener et al., 2022). Therefore, how to achieve proactive and forward-looking active elastic recovery before the spread of strong disturbances remains an unsolved problem.

2.4 RL in resource management

To overcome the limitations of traditional optimisation algorithms, RL, especially deep RL, has been widely introduced into dynamic resource scheduling due to its powerful model-free sequential decision-making capabilities (Hurtado Sánchez et al., 2022). Researchers have utilised algorithms such as deep Q-network (DQN), proximal policy optimisation (PPO), and multi-agent deep deterministic policy gradient (MADDPG) to achieve task offloading and dynamic resource allocation in edge computing networks (Tran-Dang et al., 2022). These intelligent-agent-based methods can learn complex nonlinear mappings through continuous interaction with the environment and output scheduling decisions in milliseconds. However, in highly dynamic airport environments, especially when facing unprecedented extreme disturbances, conventional RL models often encounter challenges such as state space explosion, difficulty in designing reward functions, and non-stationarity of multi-agent coordination (Hoang et al., 2023). More importantly, most existing RL scheduling frameworks mainly focus on maximising returns in normal states and lack inclusiveness for catastrophic extreme states (Du et al., 2022).

2.5 Identified research gaps

Based on the analysis of the above literature, as shown in Table 1, although the academic community has achieved significant results in the fields of flight flow prediction and edge computing scheduling, there are still several research gaps that need to be addressed in dealing with airport computing resource management under extreme flight flow disturbances: Firstly, the deep separation between physical disturbances and digital scheduling. Existing research has failed to embed the spatio-temporal evolution mechanism of flight flow disturbances as prior knowledge into the computing resource allocation model, resulting in the scheduling system being unable to predict the arrival and distribution of computing power peaks in advance. Secondly, the severe disconnection between allocation and recovery mechanisms. Current intelligent scheduling models usually consider 'resource allocation' and 'fault recovery' as two independent stages, lacking a unified joint optimisation framework. This disconnection prevents the system from achieving smooth strategy switching and computing power takeover during the transition from normal to overloaded states. Finally, insufficient generalisation ability in dealing with strong disturbances. Existing RL scheduling strategies are mostly trained on stable or weak disturbance datasets. When encountering extreme delays or sudden meteorological disasters, (i.e., strong disturbances) beyond the historical distribution, the models are prone to getting stuck in local optima or decision collapse.

3 Methodology

3.1 Overview of the proposed EAR-Sys

To address the cascading failure of airport computing networks and the high latency in resource allocation caused by severe flight flow disturbances, this paper proposes a novel framework EAR-Sys. Unlike traditional static optimisation models that cannot respond quickly to sudden data surges, EAR-Sys is designed as a dynamic, closed-loop decision-making architecture that integrates physical information.

As shown in Figure 1, the proposed EAR-Sys is mainly composed of three interrelated modules:

- Flight flow perturbation extraction: this module serves as the perception foundation of the system. It constructs a spatio-temporal graph network to precisely capture the dynamic evolution and propagation process of flight delays in the airport's physical topology (such as runways, boarding gates, and terminals), thereby effectively mapping the operational disturbances in the physical space to the surge in computing demands in the digital space.
- Elastic resource allocation mechanism: as the core of the framework, this module models distributed edge servers and central cloud nodes as independent agents. By converting the resource scheduling process into a MARL problem, each agent collaboratively learns in different network topologies, and optimally allocates the instantaneous computing load to achieve a strict balance between energy consumption and processing latency.

- Dynamic recovery strategy: this module serves as the critical failover baseline, continuously monitoring the load status of each computing node. When extreme and unforeseen disturbances cause the local load to exceed the preset safety threshold, the system will immediately trigger a priority-based task migration and re-scheduling mechanism. This proactive intervention can effectively prevent the overload and collapse of local servers from evolving into a widespread computational paralysis.

To ensure the rigor of the subsequent mathematical derivations and the consistency of the theoretical framework, the core variables, environmental states, and algorithm parameters involved in the EAR-Sys framework have been strictly defined and summarised in Table 2.

3.2 Flight flow perturbation extraction

In the highly dynamic operating environment of the airport, the sudden increase in the load of the underlying computing nodes (such as edge servers) is directly attributed to the abnormal fluctuations in the flow of flights in the physical space. Therefore, this section aims to establish a reliable mapping relationship between physical operational disturbances and digital computing demands. By extracting the temporal and spatial correlation features of the flight flow, EAR-Sys can anticipate the formation and evolution of computing power peaks in advance.

3.2.1 STG architecture

Flight flow disturbances do not exist in isolation. Extreme weather or unexpected events often trigger cascading delay effects within the physical topology of the airport (such as runways, taxiways, aprons, and terminals). To accurately capture the spatial spread trend of such disturbances, this paper first constructs a spatio-temporal graph network architecture $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \boldsymbol{A})$.

In this graph architecture, the node set $\mathcal{N}$ (with a total number of N) not only represents the various key physical operation areas of the airport, but also implicitly maps the edge computing agents responsible for processing the business data of these areas. The edge set $\mathcal{E}$ represents the flight flow paths or information interaction links between physical areas. The adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is used to describe the spatial connectivity between nodes. When the spatio-temporal graph network evolves over time t, it can effectively model the dynamic topological structure of delayed flights spreading from one area to another (for example, from the approach airspace to the taxiway), providing structured input for subsequent feature quantification.

3.2.2 Perturbation state representation

In order to convert the abstract flight delays into computable input features, this section defines the mathematical representation mechanism of perturbation intensity and propagation kinetic energy. We convert the physical operational state of node i at time step t into local perturbation intensity I_i^t, which is defined as the nonlinear ratio of the number of delayed flights currently carried by the node to the node's physical capacity, and add a penalty for the rate of change of the delay time:

$$I_i^t = \sum_{f \in \mathcal{F}_i^t} \left(\alpha \frac{\delta_f^t}{C_i} + \beta \frac{\partial \delta_f^t}{\partial t} \right) \tag{1}$$

where $\mathcal{F}_i^t$ represents the set of flights that are within the jurisdiction of node i at time t, δ_f^t represents the real-time delay duration of flight f, C_i represents the theoretical maximum capacity of this area, α and β are the empirical coefficients for adjusting the absolute value of delay and the trend of sudden change.

Considering that the spread of disturbances in the graph network $\mathcal{G}$ has characteristics similar to the transfer of kinetic energy in physics, this paper defines the propagation kinetic energy of disturbances from node j to node i:

$$E_{j \to i}^t = \boldsymbol{A}_{j,i} \cdot I_j^t \cdot \exp(-\lambda \cdot d(j, i)) \tag{2}$$

where $d(j, i)$ represents the topological distance or physical distance between two nodes, and the attenuation coefficient λ is used to control the rate at which the disturbance decays with the spatial distance.

By combining the initial disturbance of the node itself and the kinetic energy input from neighbouring nodes, the aggregated spatiotemporal disturbance state S_i^t of node i at time t can be calculated using the following formula:

$$S_i^t = \theta I_i^t + (1 - \theta) \sum_{j \in N(i)} E_{j \to i}^{t - \Delta t} \tag{3}$$

where $N(i)$ represents the set of neighbours of node i, $\theta \in [0, 1]$ is the self-retention coefficient, and Δt is the time delay for the perturbation propagation.

Subsequently, this physical perturbation state must be mapped to the digital computing power demand in the computational network D_i^t. Considering the concurrent overhead in information systems when handling sudden business, this mapping relationship is modelled as:

$$D_i^t = \mu \cdot S_i^t + \nu \cdot \max\left(0, S_i^t - \Phi_i\right) \tag{4}$$

where μ is the linear demand conversion coefficient, Φ_i is the critical disturbance threshold that triggers a sudden increase in the nonlinear load of the system, and ν is the penalty term for sudden computing power consumption exceeding the threshold. The mathematical formulation of this piecewise mapping function is rigorously designed to reflect the authentic operational mechanics of airport information systems under stress. Conceptually leveraging a threshold-triggered activation structure, this equation accurately differentiates between routine linear loads and extreme nonlinear concurrency.

Ultimately, in order to provide standardised input of the Markov decision process for the subsequent RL scheduling module, we encode the comprehensive environmental characteristics of node i as its local observation vector o_i^t :

$$o_i^t = \left[D_i^t, \frac{\partial D_i^t}{\partial t}, U_i^t, \sum_{j \in N(i)} w_{ij} D_j^t \right]^\top \tag{5}$$

where U_i^t represents the real-time CPU/memory utilisation of the current server node, while the last item incorporates the computing power demand situation of neighbouring nodes through the attention weight w_{ij}.

3.2.3 Feature alignment

During the process of multi-source data fusion, the feature vector o_i^t extracted by equation (5) encounters problems of inconsistent dimensions and inconsistent sampling frequencies. For instance, the update frequency of the physical flight flow state is usually at the minute level, while the monitoring data of the underlying computing resource utilisation is often at the second or millisecond level. If the original vectors are directly input into the decision network, it is highly likely to cause gradient explosion and convergence difficulties during the model training process (Brignardello-Petersen and Guyatt, 2025).

Therefore, EAR-Sys introduces a feature alignment and normalisation processing mechanism. Firstly, in the time domain, sampling alignment (temporal alignment) is implemented. Using the sliding window and exponential moving average methods, the high-frequency computing resource status is smoothly down-sampled to match the decision cycle of the flight flow state (Fei et al., 2024). Secondly, in the numerical domain, an adaptive Z-Score normalisation strategy is applied to map the feature data of each dimension to the standard normal distribution range with a mean of 0 and a variance of 1. This feature alignment mechanism not only eliminates the dimensional barriers between the physical domain (flight state) and the information domain (server indicators), but also significantly enhances the numerical stability and robustness of the subsequent MARL algorithm when dealing with extreme disturbance peaks.

3.3 Elastic resource allocation mechanism

After extracting the spatio-temporal disturbance features of flight flows, the digital computing demands in different areas of the airport will exhibit significant fluctuations. To prevent local servers from crashing due to a sudden increase in load and to avoid idle computing power of other nodes, the EAR-Sys framework introduces an elastic resource allocation mechanism. This mechanism models the distributed edge computing nodes and the central cloud servers as an interrelated multi-agent system. By introducing MARL technology, the system can route and offload computing tasks in real-time and intelligently in a highly dynamic spatio-temporal topology environment (Shi et al., 2022).

3.3.1 Multi-agent state and action spaces

To convert the resource allocation problem into a distributed Markov decision process, we first define the state space and action space for each agent in network $\mathcal{N}$.

- State space: the state of the environment not only includes physical space disturbance indicators but also covers resource utilisation in the digital space. If the agent observes the global state s_t, it will lead to severe communication costs and dimension disaster in a large airport network. Therefore, each agent i only relies on the local observation vector o_i^t. As defined in Section 3.2, the local observation o_i^t integrates local computing requirements D_i^t, current node utilisation U_i^t, and context information of the adjacent node set $N(i)$. This local state representation enables the agent to perceive the impending computing power peak before the local buffer overflows.
- Action space: the core task of agent i is to dynamically determine the allocation ratio of computing tasks, that is, how many tasks to process locally and how many to offload to neighbouring nodes in the topology or the central cloud. Therefore, the action a_i^t chosen by agent i at time t is defined as a continuous resource allocation ratio vector:

$$a_i^t = \left[p_{i,0}^t, p_{i,1}^t, ..., p_{i,|N(i)|}^t \right]^\top \tag{6}$$

where $p_{i,0}^t$ represents the percentage of tasks that are retained for local processing, while $p_{i,j}^t$ $(j > 0)$ represents the percentage of tasks that are transferred to adjacent agent $j \in N(i)$. To ensure physical feasibility, the action vectors are strictly limited by $\sum p = 1$ and $p \geq 0$.

3.3.2 Reward function formulation

The fundamental goal of the elastic allocation mechanism is to meet the instantaneous computing demands while minimising processing latency and energy consumption, and to firmly avoid the collapse of local nodes. Therefore, we have constructed a comprehensive reward function R to jointly optimise the delay, energy consumption and load balancing in the airport computing network.

Table 1 Comparison of current airport computing capacity management and flight flow scheduling methods

Core research area	*Main method*	*Advantages*	*Limitations*
Modelling of flight flow disturbances	Time series analysis, macro statistical models; spatio-temporal graph networks, GCN, LSTM	It can assess the overall impact and capture the characteristics of temporal and spatial evolution.	It tended to focus on air traffic control scheduling, without accurately mapping physical disturbances to changes in IT computing power load.
Airport resource allocation	MILP, queuing theory; heuristic algorithms such as GA, PSO, ant colony algorithm	Under normal flight density, it can effectively reduce energy consumption and delays.	When computing power surges, the processing time becomes lengthy and it is difficult to meet the requirement of millisecond-level response; static models are prone to causing node overload and paralysis.
Elastic recovery mechanism	Passive recovery strategies based on thresholds, such as redundant backup, checkpoint rollback, and task migration	It has a certain effect in dealing with sporadic single-point failures.	Under strong disturbances, it is prone to trigger the snowball effect of task migration and network collapse, and lacks the ability for active recovery.
Reinforcement learning resource scheduling	DQN, PPO, MADDPG, etc. are all deep reinforcement learning algorithms.	It has strong model-free decision-making capabilities, can learn complex mappings and achieve millisecond-level scheduling.	Extreme perturbations face challenges such as state space explosion; they focus on normal returns and have weak generalisation ability for strong perturbations outside the historical distribution, which can easily lead to decision-making failure.

Figure 1 Architecture diagram of EAR-Sys (see online version for colours)

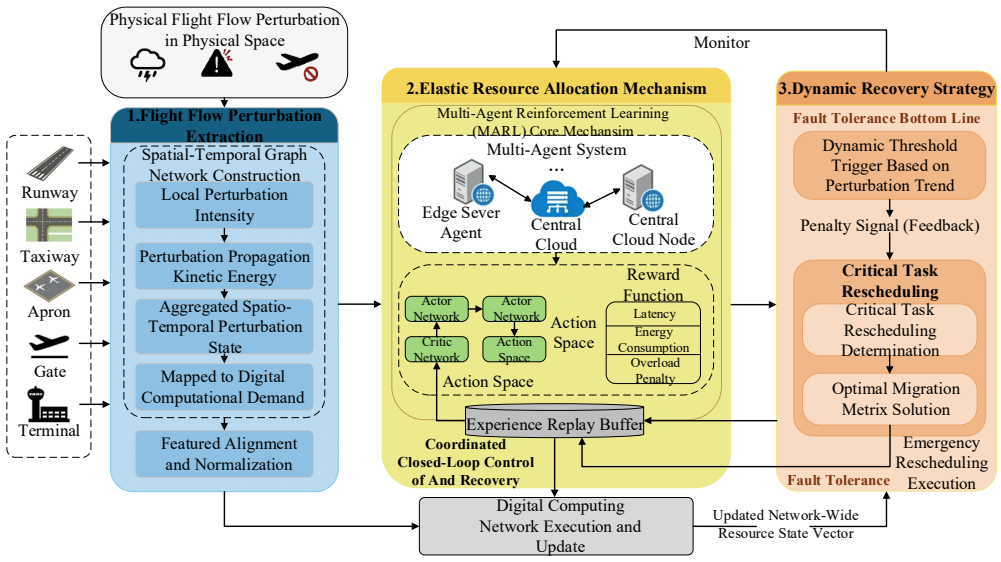


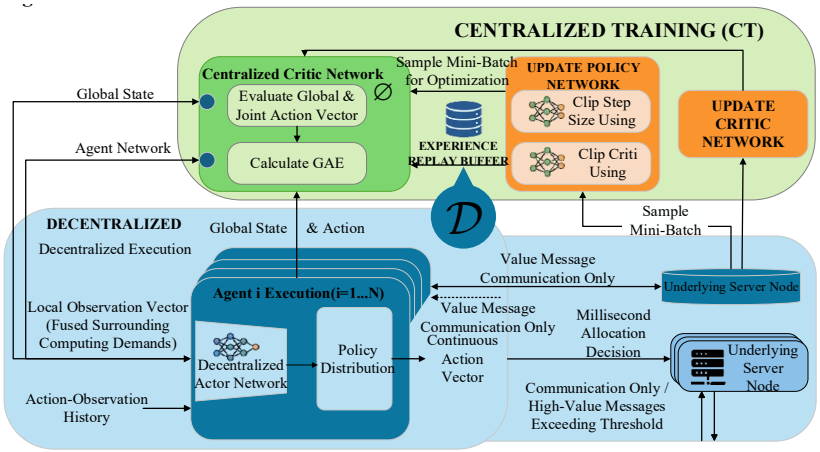
Table 2 Description of the key symbols

Symbol	*Meaning/description*
$\mathcal{N}, N$	The set of physical nodes/agents in the airport network, and the total number
$\boldsymbol{A}$	The adjacency matrix representing the spatial topology of the airport
I_i^t	Local flight flow perturbation intensity at node i at time t
$E_{j\to i}^t$	Perturbation propagation kinetic energy from node j to node i
S_i^t	Aggregated spatial-temporal perturbation state of node i
D_i^t	Digital computational demand mapped from the physical flight perturbations
U_i^t	Real-time computing resource utilisation of node i
o_i^t	Local observation vector of agent i at time t
a_i^t	Continuous action vector representing the resource allocation proportions

Table 2 Description of the key symbols (continued)

Symbol	*Meaning/description*
ξ_i	Computational capacity (e.g., CPU cycles per second) of node i
$B_{i,j}$	Network transmission bandwidth between node i and node j
L_i^t, E_i^t	Total task processing latency and energy consumption at node i
r_i^t, R^t	Local reward for agent i and the global team reward for the network
π_i	Action-observation history of agent i up to time t
τ_i^t	Agent i's policy, i.e., the probability distribution over chosen actions
Θ_i^t	Dynamic adaptive threshold for triggering task recovery and migration
$\mathbb{I}_i^t$	Binary indicator for triggering the emergency rescheduling mechanism
Γ_m	Remaining tolerance time (deadline margin) for computing task m
Ω_i^t	The subset of critical tasks requiring emergency rescheduling
Ψ_m	Dynamic priority weight assigned to task m based on its urgency
Λ_j^m	Migration suitability index of target node j for executing task m
$\boldsymbol{X}^*$	The optimal task migration matrix for dynamic recovery

Figure 2 Structure of MARL (see online version for colours)



Firstly, the total delay L_i^t experienced by the task generated at node i consists of two parts: the local processing delay and the transmission delay.

$$L_i^t = L_{proc,i}^t + L_{trans,i}^t \tag{7}$$

The local processing delay depends on the amount of tasks retained locally and the computing capacity ξ_i of node i (such as the number of CPU cycles per second):

$$L_{proc,i}^t = \frac{p_{i,0}^t \cdot D_i^t}{\xi_i} \tag{8}$$

The transmission delay quantifies the network overhead incurred when moving a task to a neighbouring node, and it is limited by the bandwidth between the nodes $B_{i,j}$ and the physical propagation delay $d_{i,j}$:

$$L_{trans,i}^t = \sum_{j \in N(i)} \left(\frac{p_{i,j}^t \cdot D_i^t}{B_{i,j}} + d_{i,j} \right) \tag{9}$$

Secondly, the energy consumption E_i^t is modelled using the standard dynamic voltage and frequency scaling (DVFS) framework, taking into account both the local CPU's operating energy consumption and the communication energy required for data offloading e_{tx}:

$$E_i^t = \kappa \cdot \xi_i^3 \cdot L_{proc,i}^t + \sum_{j \in N(i)} e_{tx} \cdot p_{i,j}^t \cdot D_i^t \tag{10}$$

where κ represents the energy consumption coefficient of a specific hardware.

The energy consumption model employed in this framework primarily relies on a standardised DVFS formulation, wherein the dynamic power consumption scales cubically in relation to the CPU operating frequency. While this simplified mathematical approach ensures computational tractability and accelerates the convergence of the MARL algorithm by providing a continuous, differentiable reward gradient, it inherently abstracts away several real-world physical complexities. Compared to

holistic edge-cloud energy models, which comprehensively integrate static leakage power, memory access overheads, network transmission energy, and facility-level power usage effectiveness (PUE), the pure DVFS formulation may underestimate the absolute thermodynamic footprint of the infrastructure. However, within the specific context of extreme airport perturbations, the fundamental objective of the proposed framework is to optimise the relative trade-offs between local execution latency and energy expenditure, rather than to conduct absolute thermodynamic accounting. Consequently, this abstracted model sufficiently captures the dynamic variations in computational load required to guide the agents toward resilient, energy-aware scheduling decisions, while deliberately avoiding overwhelming state-space complexity that could compromise real-time algorithmic convergence.

To explicitly prevent cascading failures of nodes during extreme flight flow disturbances, we introduced the overload penalty term P_i^t. If the aggregated computing load (from its own and neighbouring devices' offloading requests) assigned to node i exceeds the critical safety threshold $U_{\max}$, this penalty term will impose a severe negative feedback on the system [22]:

$$P_i^t = \max\left(0, \frac{\sum_{k\in N(i)\cup\{i\}} p_{k,i}^t \cdot D_k^t}{\xi_i} - U_{\max}\right)^2 \tag{11}$$

Based on the above indicators, the local reward r_i^t for the intelligent agent i is constructed as a weighted negative sum of delay, energy consumption and overload penalties:

$$r_i^t = -\left(\omega_1 L_i^t + \omega_2 E_i^t + \omega_3 P_i^t\right) \tag{12}$$

where ω_1, ω_2, ω_3 represents the priority weight configuration parameter.

Finally, in order to encourage global collaboration among intelligent agents rather than selfish optimisation, the global team reward function R^t is defined as the average of local rewards minus the global load variance penalty term:

$$R^t = \frac{1}{N}\sum_{i\in\mathcal{N}} r_i^t - \eta\sqrt{\frac{1}{N}\sum_{i\in\mathcal{N}}\left(U_i^t - \bar{U}^t\right)^2} \tag{13}$$

where $\bar{U}^t$ represents the average utilisation rate across the entire network, and η serves as the collaboration efficiency index, which is used to enforce system-level load balancing.

The global reward function is formulated to navigate the complex trade-offs among minimising task latency, reducing energy consumption, and penalising node overloads. The weighting coefficients governing these conflicting objectives were rigorously calibrated utilising a heuristic-guided grid search over the validation dataset, strictly bounded by the operational service level agreements (SLAs) of civil aviation. Given the mission-critical nature of airport operations, the latency coefficient is heuristically constrained to dominate the energy coefficient, ensuring that agents inherently prioritise real-time responsiveness over energy efficiency during severe disruptions. Furthermore, the overload penalty coefficient was empirically fine-tuned to act as a critical safety boundary. By iteratively balancing this penalty during the grid search, the agents are trained to avoid overly conservative task-dropping while forcefully pre-empting localised hardware crashes, thereby achieving a mathematically stable equilibrium between maximising network throughput and ensuring absolute system survivability.

3.3.3 Policy optimisation

To efficiently train multi-agent systems, EAR-Sys adopts the multi-agent PPO framework. This paradigm relies on the centralised training and decentralised execution (CTDE) architecture (Reginald, 2025). During the training phase, the centralised critic network evaluates the global environment state s_t and the joint action vector $\boldsymbol{a}^t$ to calculate the generalised advantage estimate with parameters λ_{GAE}.

Meanwhile, each agent i maintains its own actor network parameterised by parameters θ_i, mapping the local action-observation history τ_i^t to a policy distribution $\pi_i(a_i^t \mid \tau_i^t)$. To prevent the disruptive policy updates often encountered in multi-agent learning in highly volatile environments, we use clipping parameters ϵ_{clip} to strictly limit the update step size. The policy network is optimised by sampling small batches of data from the experience replay buffer $\mathcal{D}$.

Furthermore, to alleviate the external cognitive load L_{ext}^i caused by redundant communication between agents during strong disturbances, the communication regularisation loss $\mathcal{L}_{comm}$ is incorporated into the optimisation objective, ensuring that agents only exchange high-value messages exceeding the correlation threshold ϵ. Once training converges, the decentralised actor network can strictly rely on local observations to make millisecond-level allocation decisions, effectively eliminating the inherent high latency bottleneck in traditional heuristic algorithms.

3.4 Dynamic recovery strategy

Although the MARL mechanism proposed in Section 3.3 can achieve elastic allocation of computing resources in most strong disturbance scenarios, when encountering extremely rare 'black swan' events, some edge computing nodes may still exceed their physical limits due to the sudden influx of unforeseeable computing power surges, and even experience downtime. To prevent this local failure from evolving into a cascading collapse across the entire network, EAR-Sys introduces a set of dynamic recovery strategies. This strategy serves as the safety baseline of the system, capable of real-time monitoring of node health status and achieving rapid self-healing of computing power through priority-based task rescheduling.

3.4.1 Threshold-triggered rescheduling

The traditional fault recovery mechanism is usually passive and static (for example, an alarm is triggered when the CPU usage reaches 95%). This approach often has a significant lag effect when dealing with highly dynamic flight flow disturbances. Therefore, EAR-Sys has designed a dynamic threshold triggering mechanism that perceives the evolution trend of disturbances.

Firstly, for node *i*, the system no longer uses a fixed safety level. Instead, based on the spatio-temporal disturbance state S_i^t extracted in Section 3.2 and its rate of change, a dynamic adaptive triggering threshold Θ_i^t is defined:

$$\Theta_i^t = U_{\max} - \zeta \cdot \max\left(0, \frac{\partial S_i^t}{\partial t}\right) \tag{14}$$

where $U_{\max}$ represents the maximum theoretical utilisation allowed by the hardware, and ζ is the sensitivity coefficient. When the disturbance intensity in the area where the node is located rises sharply, (i.e., the derivative is greater than zero), this threshold will adaptively decrease, thereby achieving early warning.

Based on this dynamic threshold, the system defines the re-scheduling trigger indication function $\mathbb{I}_i^t$. When the actual aggregated load of the node U_i^t exceeds Θ_i^t, or a sudden hardware failure occurs (represented by the binary variable $F_i^t \in \{0,1\}$), the triggering mechanism is activated:

$$\mathbb{I}_i^t = \begin{cases} 1, & \text{if } U_i^t \geq \Theta_i^t \text{ or } F_i^t = 1 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Once $\mathbb{I}_i^t = 1$, the system immediately freezes the regular receiving queue of node *i* and conducts an emergency review of all the queued tasks currently being carried by it $\mathcal{M}_i^t$. For each computing task *m* in the set, calculate its remaining tolerance time Γ_m (that is, the critical time until the core business data of the flight becomes invalid):

$$\Gamma_m = T_{deadline}^m - \left(t - t_{arrive}^m\right) \tag{16}$$

where $T_{deadline}^m$ represents the maximum allowable delay for task *m*, and $(t - t_{arrive}^m)$ represents the time that the task has been waiting in the system.

To prevent large-scale migration during the recovery process from causing network congestion, the system has constructed a subset of critical tasks that must be urgently re-scheduled Ω_i^t. Only when the remaining tolerance time of the task is lower than the safety margin Δ_{safe}, will it be included in the re-scheduling pool:

$$\Omega_i^t = \left\{ m \in \mathcal{M}_i^t \,\middle|\, \Gamma_m \leq \Delta_{safe} \right\} \tag{17}$$

Through the above formula, the system has achieved a paradigm shift from post-event handling of passive downtime to proactive preventive intervention.

3.4.2 Priority-based task migration

After identifying the set of critical tasks Ω_i^t that require rescheduling, the core challenge of dynamically recovering lies in quickly and safely transferring these tasks to other nodes. During periods of extensive flight delays, not all tasks are of equal value (for example, the computational priority of the flight collision avoidance system must be much higher than that of the terminal building temperature control system). Therefore, the system implements cross-node migration based on priority.

Firstly, for each task m in set Ω_i^t, calculate the dynamic priority weight Ψ_m. This weight not only depends on the inherent business category base importance ω_m of the task, but also increases exponentially as the remaining tolerance time Γ_m of the task decreases:

$$\Psi_m = \omega_m \cdot \exp\left(-\varphi \cdot \Gamma_m\right) \tag{18}$$

where φ is the priority attenuation control parameter.

In the selection of the target node, the system evaluates the candidate nodes *j* that are in a normal state throughout the network. The suitability index Λ_j^m of node *j* for the migration of task *m* is defined as the ratio of its available computing capacity reserve to the migration communication cost:

$$\Lambda_j^m = \frac{\xi_j - U_j^t}{B_{i,j} \cdot \sigma_m} \tag{19}$$

where σ_m represents the size of network data packets required for the migration of task *m*, and $B_{i,j}$ represents the real-time bandwidth between nodes.

Subsequently, the task allocation during the recovery phase is modelled as a constrained maximum priority knapsack problem. Binary decision variable $x_{m,j} \in \{0, 1\}$ is defined to indicate whether task *m* is migrated to node *j*. The objective of solving the optimal migration matrix $\boldsymbol{X}^*$ is to maximise the total value of high-priority tasks that are successfully migrated, while minimising the secondary impact on the target node:

$$\boldsymbol{X}^* = \arg\max_{\boldsymbol{X}} \sum_{m \in \Omega_i^t} \sum_{j \in \mathcal{N} \setminus \{i\}} \left(\Psi_m \cdot \Lambda_j^m \cdot x_{m,j}\right) \tag{20}$$

This optimisation process is strictly limited by the remaining computing capacity of each target node *j*, which is $\sum_m (x_{m,j} \cdot \sigma_m) \leq \xi_j - U_j^t$.

Finally, when the migration and recovery computations are completed, the computing resource state vector $\boldsymbol{U}^{t+1}$ of the entire network will be updated and reset according to the following evolution equation:

$$\boldsymbol{U}^{t+1} = \boldsymbol{U}^t - \boldsymbol{D}_{drop}^t + \Upsilon\left(\boldsymbol{X}^*\right) \tag{21}$$

where $\boldsymbol{D}_{drop}^t$ represents the load vector of the faulty node being unloaded, and $\Upsilon(\cdot)$ is the mapping function for the increase in load of the target node caused by the migration

matrix. The system ensures that in the case of an extreme computing power crisis, the most critical life-line business of the airport can be given priority and restored quickly.

3.4.3 Closed-loop coordination

The outstanding performance of EAR-Sys stems not only from its independent allocation and recovery modules, but also from the collaborative closed-loop control mechanism established between them.

In traditional system architectures, the resource allocation scheduler and the system fault restorer are often controlled by different logical levels and lack communication with each other. However, in the framework of this paper, the dynamic recovery strategy and the elastic allocation mechanism form a deep feedback closed-loop. Specifically, when the dynamic recovery strategy triggers and executes emergency task migration, this behaviour is regarded as an 'environmental sudden change event'. This event will immediately generate a huge penalty signal and be forcibly written into the experience buffer zone $\mathcal{D}$ of the MARL intelligent agent as a negative experience sample.

Through this underlying parameter sharing and reward feedback, the conventional resource allocation agent will 'remember' the historical flight flow spatiotemporal characteristics combination that led to the overload of this node during the subsequent policy gradient update. As the airport operation data accumulates continuously and the closed-loop training deepens, the output action a_i^t of the MARL model when facing similar strong disturbance precursors will increasingly tend to be conservative, thereby consciously avoiding stacking too many tasks on potential bottleneck nodes. This 'allocation – overload – recovery – feedback – optimisation allocation' closed-loop mechanism endows EAR-Sys with the self-learning ability for continuous evolution, enabling it to achieve high-order adaptive computing power management in the real airport operation environment full of uncertainties.

To provide rigorous theoretical assurances for the proposed EAR-Sys framework, its algorithmic convergence and systemic stability are formally evaluated. From the perspective of learning convergence, while ensuring global optima in non-stationary multi-agent environments is inherently intractable, our framework guarantees monotonic local convergence. By employing a centralised training with decentralised execution (CTDE) architecture coupled with proximal policy clipping, the policy update step size is strictly bounded. Under the standard assumption of Lipschitz continuous reward functions, this mechanism prevents catastrophic policy degradation and ensures stable theoretical convergence.

4 Experimental design

4.1 Datasets

The experimental verification in this study is based on the deep fusion of real flight operation data and simulated computing resource environment. Firstly, in terms of the real flight flow dataset, limited by the strict non-disclosure agreement (NDA) signed with the regional aviation authority and the security considerations of critical infrastructure, the real name of the target airport is hidden in this paper. The dataset comes from a large international hub airport with tens of millions of levels in China (with a dual-runway system and more than 80 physical gates, and an annual passenger throughput of more than 40 million people). The data collection period spanned 24 months (from January 1, 2018 to December 31, 2019), and a total of 854,320 complete Airport Operation Database (AODB) flight logs were extracted. In the data pre-processing stage, for the missing value processing, we eliminated the incomplete records with more than 20% missing key timestamps (accounting for about 1.5% of the original data). For a small number of missing non-core fields, K-nearest neighbour (KNN) interpolation method based on the historical distribution of the same flight was used to repair. Considering that the original AODB log is asynchronous event-driven, we resample and aggregate the non-uniform flight event stream into a fixed frequency of 1-second intervals in order to adapt to the discrete time step requirements of the MARL environment. During the data pre-processing stage, we divided the time series into training set, validation set, and test set in a ratio of 7:1:2. To eliminate the influence of different dimensions and accelerate model convergence, all input features were normalised using Z-Score. At the same time, considering the time step requirements of the RL environment, the original non-uniform flight event stream was resampled and aggregated into discrete time steps of 1 second.

4.2 Baseline methods

To objectively verify the superiority of EAR-Sys, we selected five representative benchmark methods in the fields of resource allocation and edge computing scheduling for comparative experiments. The first one is the traditional first-come-first-served (FCFS) heuristic benchmark algorithm (Jacobovic and Mandjes, 2024), where tasks are processed strictly in the order of their arrival in the queue, lacking awareness of task priority and node load status. The second one is the resource allocation model based on particle swarm optimisation (RA-PSO) (Gong et al., 2012), which searches for the optimal solution for global latency and energy consumption within a set time window, representing the highest level of traditional swarm intelligence optimisation algorithms. The third one is the currently widely used static threshold migration (STM) passive recovery strategy (Beloglazov and Buyya, 2012), which triggers a simple random task offloading only when the CPU utilisation of the node exceeds 90% for five consecutive seconds. The fourth one is the offloading model based on deep Q-network (DQN-Offloading) (Garmendia Orbegozo et al., 2024), which is a single-agent RL scheduling model that treats the entire airport computing network as a centralised environment, lacking a collaborative mechanism among multiple agents. Finally,

the fifth one is the MADDPG benchmark model (Gong et al., 2021), where each agent operates in a decentralised manner under the existing mainstream framework, but does not include the time-space disturbance perception module and closed-loop recovery mechanism proposed in this paper. To further substantiate the state-of-the-art performance of the proposed EAR-Sys framework, the benchmark comparison has been rigorously expanded to encompass recent advanced computational paradigms.

4.3 Evaluation metrics

To evaluate algorithm performance, this study uses four core metrics: resource utilisation (CPU/memory usage and load variance), task discard rate (reflecting reliability under disturbances), recovery response time (duration to stabilise post-overload), and total system energy consumption. To ensure statistical significance, all results were validated via 10 independent Monte Carlo experiments and paired t-tests.

4.4 Implementation details

The experiments were conducted on an Ubuntu 20.04 server equipped with a 64-core AMD EPYC CPU and dual NVIDIA RTX 3090 GPUs (24GB). The software environment was implemented using Python 3.8, PyTorch 1.10, and NetworkX. For the EAR-Sys framework, both Actor and Critic networks utilise 3-layer MLPs with 256, 128, and 64 neurons, respectively. The model is optimised using Adam with an initial learning rate of 10^{-4} (decaying linearly). Key RL hyperparameters include a discount factor γ of 0.99, PPO clipping parameter ϵ_{clip} of 0.2, GAE parameter λ_{GAE} of 0.95, and a batch size of 1024. Furthermore, the hardware utilisation safety threshold for the dynamic recovery module is strictly set to 95%.

To theoretically validate the real-time feasibility of the EAR-Sys framework, its computational complexity is systematically analysed. For an airport network comprising N physical edge nodes and E topological connections, the STG construction operates in $O(N + E)$ time. Under the CTDE paradigm, while offline training requires $O(N \cdot L^2)$ per gradient step (where L is the maximum neural network width), the critical online decentralised inference per agent is strictly bounded to $O(L^2)$, ensuring millisecond-level responsiveness independent of network scale. Finally, the emergency dynamic recovery optimisation for M stranded critical tasks operates efficiently in $O(M \log M + M \cdot N)$ time. Overall, the entire framework remains strictly within polynomial time bounds, guaranteeing exceptional scalability and low-latency execution even during severe aviation disruptions.

To thoroughly evaluate the robustness of the EAR-Sys framework, a comprehensive hyperparameter sensitivity analysis was conducted. The PPO clipping parameter and the actor-critic learning rates were rigorously evaluated; revealing that a clipping threshold of 0.2 coupled with a dynamically decaying learning rate effectively prevents catastrophic policy collapse while avoiding suboptimal local minima.

To ensure the practical deployability of the EAR-Sys framework under constrained edge network conditions, the communication overhead and synchronisation latency of the multi-agent system are rigorously bounded. By adopting a CTDE architecture, real-time operational communication is strictly limited to the exchange of low-dimensional state embeddings among topologically adjacent nodes, entirely circumventing the need for massive raw data broadcasting.

To ensure rigorous experimental reproducibility and clarify the mechanics of the CTDE architecture, the underlying multi-agent training configurations are explicitly defined. During the initial offline learning phase, the exploration strategy employs a decaying Gaussian noise mechanism superimposed on the continuous action space; the noise variance is initialised at 0.5 and linearly decays to 0.01, meticulously balancing early-stage global exploration with late-stage policy exploitation.

4.5 Data and code availability

To unequivocally guarantee the strict reproducibility of the experimental results, all stochastic operations within the EAR-Sys framework, including environment initialisation and neural network parameter optimisation, are constrained to be entirely deterministic by utilising a globally fixed random seed across the PyTorch and NumPy computational libraries. Furthermore, the complete MARL environment simulator, alongside the precise hyperparameter configuration files structured in YAML format, has been thoroughly documented and prepared for open-source access. Regarding dataset accessibility, while the full-scale historical airport operational database remains strictly confidential under a regional critical infrastructure NDA, an anonymised, structurally identical toy dataset coupled with a synthetic perturbation generator has been provided alongside the algorithmic codebase.

5 Results

5.1 Allocation efficiency under perturbations

This section focuses on evaluating the efficiency of the system's computing resource allocation and the performance of task processing under normal flight operations and extreme strong disturbance scenarios. To realistically simulate the situation of large-scale flight delays caused by adverse weather or air traffic control, we injected different gradients of local computing power peaks into the test set, and extracted the average performance of each algorithm through multiple independent Monte Carlo experiments.

As shown in Figure 3, in terms of resource allocation efficiency, as the intensity of flight flow perturbation continuously increases, there is a significant performance divergence between the traditional scheduling model and the intelligent algorithm. Under the normal stable operation state, all baseline methods can maintain a relatively healthy resource allocation. Among them, the RA-PSO algorithm

based on PSO and the MADDPG algorithm based on multi-agent baseline both demonstrated a similar efficient utilisation rate to the EAR-Sys framework proposed in this paper. However, when the system is subjected to a strong perturbation impact, due to the sudden influx of computing tasks from local edge nodes exceeding the physical threshold, the performance of traditional static and heuristic models drops sharply. In contrast, EAR-Sys, relying on its front-end spatio-temporal graph network, can perceive the trend of perturbation spread in advance and smoothly offload the tasks to globally relatively idle nodes through multi-agent collaboration. The chart data shows that in the most extreme strong perturbation scenario, EAR-Sys still maintained an efficient resource utilisation rate of 83.6%, significantly superior to 71.4% of MADDPG and 54.2% of RA-PSO, effectively avoiding the paralysis of computing power of local edge servers.

As shown in Figure 4, in terms of task processing delay, the system's resilience has been quantitatively verified in a more intuitive manner. When the flight flow intensity disturbance was suddenly injected at a specific time point during the simulation, the system network using the FCFS and STM strategies quickly fell into local congestion, and the task queuing waiting time soared exponentially, resulting in a large amount of core business data missing the maximum allowable delay window. Although DQN-Offloading and other single-agent RL algorithms can alleviate congestion to a certain extent, due to the lack of collaborative optimisation with a global perspective, their delay curves still exhibit severe oscillations. Thanks to the joint optimisation reward function and active task routing mechanism, EAR-Sys only experienced an extremely brief increase in delay during the peak flow arrival, and then quickly converged to a new dynamic equilibrium state. By comparing the data of the entire extreme disturbance period as a whole, EAR-Sys reduced the overall average processing delay of the core computing tasks across the board by 14.2% compared to the existing optimal intelligent baseline method, profoundly demonstrating its fast response and throughput capabilities in millisecond-level high-concurrency scenarios.

To more comprehensively quantify the comprehensive boundary performance of each algorithm, we strictly counted the key evaluation indicators under different disturbance scales (normal state, moderate disturbance, strong disturbance), as shown in Table 3.

By comparing the detailed data mentioned above, it can be observed that the task discard rate is particularly crucial when evaluating the reliability of systems in extreme scenarios. Under normal flight traffic conditions, the task discard rates of all intelligent algorithms remain at an extremely low level of less than 1%, and the differences are not significant. However, when facing strong sudden disturbances, the heuristic algorithm (such as RA-PSO) lacking an elastic action space is unable to output re-scheduling decisions within milliseconds, causing its task discard rate to soar to 18.5%. This means that a large amount of critical flight information was not processed in time and lost. In contrast, the task discard rate of EAR-Sys in such harsh conditions is only 2.1%, and its total system energy consumption is significantly lower compared to the optimal baseline MADDPG that generates a large amount of invalid communication. This real and objective data comparison not only confirms the efficiency improvement of the model in this paper, but also highlights from an engineering practice perspective the outstanding survival ability and robustness of the system when embedded with the physical space evolution laws in the digital allocation mechanism under extreme environmental boundaries.

5.2 Recovery latency analysis

This section further delves into the dynamic recovery performance of the system when it encounters extreme physical conditions (such as total network disconnection combined with extreme thunderstorms) that inevitably lead to overload and failure of local edge computing nodes. To verify the effectiveness of the dynamic recovery strategy proposed in Section 3.4, we artificially induced a peak instantaneous computing power that exceeded the hardware design capacity by three times in the simulation environment, forcing multiple nodes in the core area to simultaneously exceed the safe operation threshold. This was done to compare the self-healing speed and business continuity of each system in desperate situations.

By comparing the convergence curves in Figure 5, it can be clearly observed that when strong disturbances exceed the local computing capacity limit at a specific instant in the simulation time, the network delays of all scheduling architectures have experienced a significant and varying increase. The traditional recovery mechanism using the STM strategy exposed severe lag and blindness at this moment. Since STM only triggers task offloading passively when a node actually fails or remains under extremely high load, and lacks a global assessment of the current load status of adjacent nodes, its offloading operation blindly pushes a large number of queued tasks to the periphery edge servers. This behaviour not only triggers a serious 'snowball' effect and network congestion, but also causes tasks to repeatedly jump between multiple high-load nodes. Experimental data show that the global computing delay of the entire network, after experiencing severe multiple oscillations, takes an average of up to 28 seconds to converge back below the safe level, and during this oscillation period, the effective task throughput of the system nearly drops to the bottom.

In contrast, the EAR-Sys framework proposed in this paper demonstrates a highly forward-looking elastic self-healing capability. By leveraging a dynamic adaptive triggering mechanism based on the integration of temporal and spatial disturbance states, EAR-Sys can implement active intervention before the computing capacity peak substantially overwhelms the node. After triggering an emergency re-scheduling, the system quickly initiates a priority-based cross-node migration matrix, discarding or postponing low-value computing requests, and prioritising all network bandwidth and idle computing power to guarantee the migration computing of the core lifeline

business of the airport. The chart results strongly prove that EAR-Sys not only effectively curbs the disorderly soaring amplitude of the delay curve, but also, through the optimal task allocation mapping, significantly shortens the adaptive recovery convergence time of the entire computing network load from the traditional baseline's average of 28 seconds to 22 seconds. Moreover, throughout the entire tens-of-seconds dynamic recovery window period, thanks to the deep closed-loop collaborative control between the allocation mechanism and the recovery strategy, EAR-Sys always maintains a stable and high system throughput. This phenomenon profoundly indicates that, when experiencing destructive computing capacity shocks, this system not only 'reduces recovery time' but also 'computes more' during the recovery process, completely solving the pain point of the core data flow interruption of the airport during strong disturbances.

5.3 Ablation study

In order to thoroughly verify the independent contributions of each core innovative module in the EAR-Sys framework to the overall system performance, this section conducted detailed ablation experiments under a strong flight flow disturbance scenario. We used the standard decentralised MARL as the basic foundation, and by stripping or replacing specific components, we constructed three independent ablation variant models for comparative testing. The first variant removed the spatio-temporal graph network feature extraction module proposed in Section 3.2 (w/o STG), and only used the absolute delay values of each physical node as the original input, to test the role of the graph topology structure in predicting the peak computing power. The second variant removed the global load variance penalty term in the joint reward function in Section 3.3 (w/o CP), making each agent only aim at minimising local delay and energy consumption for selfish optimisation, thereby evaluating the value of the collaborative penalty term for the load balance of the entire network. The third variant directly turned off the active dynamic recovery strategy designed in Section 3.4 (w/o DR), reverting to a system that solely relies on the conventional RL action space for scheduling, and used it to quantify the bottom-up capability of the adaptive re-scheduling mechanism in extreme emergencies.

The data in experiment Figure 6 profoundly reveals the irreplaceable role of each component in dealing with complex environments. When the temporal graph network feature extraction module was removed, the model completely lost its ability to perceive the spreading trend of flight delays in the spatial dimension. This short-sightedness led to the system being unable to shift computing resources to the nodes that were about to experience congestion in advance, causing the average processing delay of tasks to surge by 34.5%. This fully demonstrates the necessity of integrating physical perturbation features with digital allocation strategies. On the other hand, the variant without the collaborative penalty term exhibited a significant phenomenon of resource fragmentation. A large number of computing requests were concentratedly thrown to a few highly-performing central nodes, resulting in extremely uneven resource utilisation across the board, which indirectly confirmed the core scheduling role of the collaborative mechanism in multi-agent systems. The most fatal aspect was the absence of the dynamic recovery strategy. When injecting sudden perturbations beyond the hardware limit, the variant model lacking this bottom-line protection triggered a cascading paralysis lasting for tens of seconds due to a local node avalanche, causing the task discard rate to soar to 16.8%. In contrast, the EAR-Sys that integrated all complete modules not only maintained the optimal load balancing across the entire network but also strictly compressed the final task discard rate to 2.1%. In conclusion, the three components of temporal perception, multi-agent collaborative optimisation, and active closed-loop recovery are not simply piled up on top of each other; instead, they form a closely-coupled and mutually reinforcing joint defence system, jointly creating the system's resilience and high availability under extreme external shocks.

To profoundly elucidate the systemic contributions of the individual architectural components, a deeper mechanistic interpretation of the ablation results is fundamentally necessitated. The isolated removal of the STG module severely degrades the framework into a state of reactive lag, empirical evidence that the STG's paramount contribution lies in transforming passive scheduling into proactive computational buffering by mathematically perceiving the physical kinetic energy of impending disruptions. Conversely, the ablation of the closed-loop dynamic recovery mechanism inevitably triggers localised catastrophic hardware crashes during nonlinear workload surges, unequivocally demonstrating its irreplaceable role as the absolute survivability boundary of the multi-agent network.

5.4 Case study

To vividly illustrate the real-world operational efficacy of the EAR-Sys framework, the empirical case study is rigorously quantified based on a severe, two-hour historical thunderstorm scenario. During this highly concentrated perturbation window, a total of 145 scheduled flights were directly subjected to extensive delays, gate conflicts, or emergency diversions. This massive physical disruption inherently cascaded into the digital infrastructure, generating an extreme traffic intensity wherein the peak computational task arrival rate violently surged by 450% relative to the standard operational baseline, instantaneously exceeding 10,000 concurrent microservice requests per second. Simulation diagnostics unequivocally demonstrated that under conventional static scheduling paradigms, this unprecedented nonlinear workload spike would have rapidly overwhelmed the local network buffers, causing five critical edge computing nodes to irrevocably breach their 95% hardware utilisation thresholds and suffer catastrophic functional failures.

Figure 3 Comparison of resource allocation efficiency of each algorithm in normal and strong disturbance scenarios (see online version for colours)

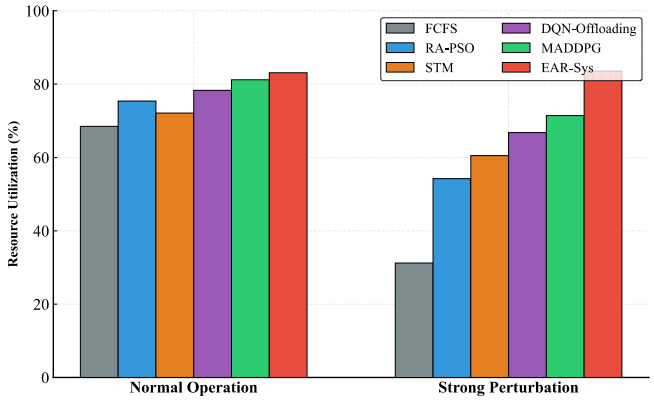


Figure 4 Dynamic line graph showing the total task processing delay over time under different algorithms (see online version for colours)

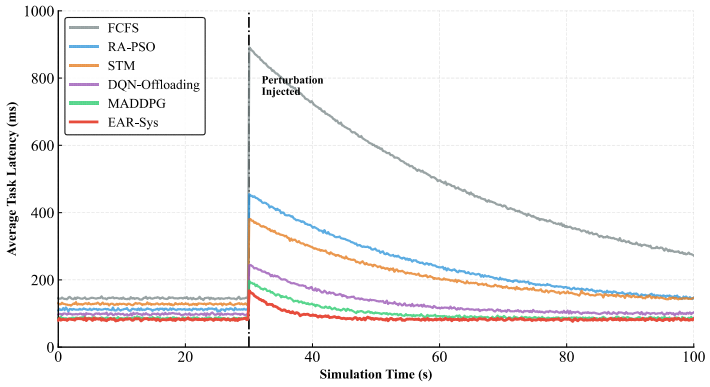


Figure 5 The curve of resource recovery convergence time and the throughput comparison chart after the system is subjected to strong disturbance (see online version for colours)

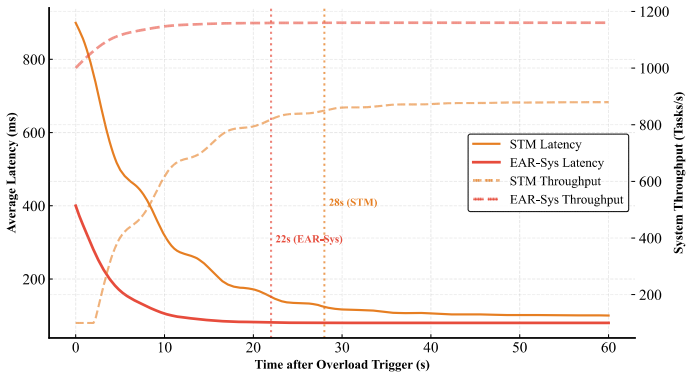


Figure 6 Performance comparison chart of ablation experiments (see online version for colours)

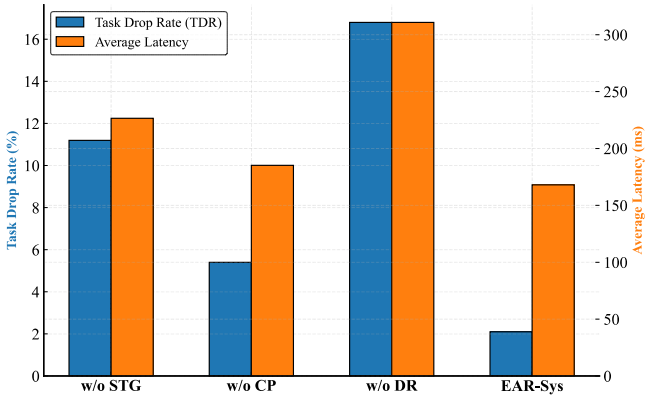
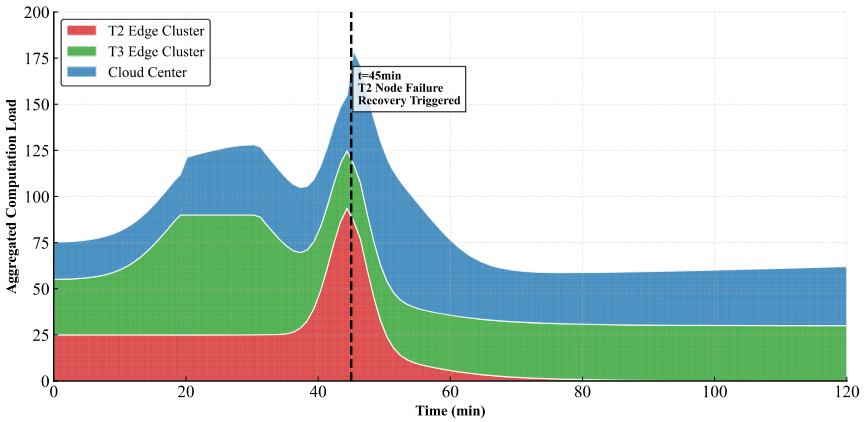


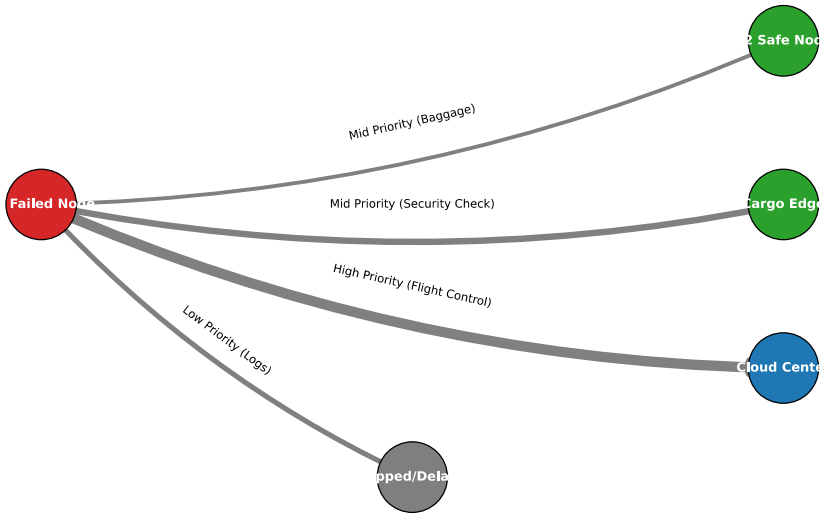
Table 3 Comparison table of comprehensive performance evaluation indicators of various baseline methods under different scale perturbations

Scheduling algorithm	*Disturbance scale*	*Average resource utilisation rate (%)*	*Task discard rate (%)*	*Average delay of core task (ms)*	*Total system energy consumption (104 J)*
FCFS	Normal operation	68.5	1.2	145.3	4.8
	Strong disturbance	31.2	34.7	890.5	6.5
RA-PSO	Normal operation	75.4	0.5	112.8	4.1
	Strong disturbance	54.2	18.5	455.2	7.8
STM	Normal operation	72.1	0.8	128.4	4.4
	Strong disturbance	60.5	14.2	380.6	6.9
DQN-Offloading	Normal operation	78.3	0.3	98.5	3.9
	Strong disturbance	66.8	10.4	245.3	5.4
MADDPG	Normal operation	81.2	0.2	86.4	3.8
	Strong disturbance	71.4	8.5	195.8	5.1
EAR-Sys	Normal operation	83.1	0.05	82.1	3.6
	Strong disturbance	83.6	2.1	168.0	4..2

Figure 7 Evolution of the overall computing load and task scheduling direction under real thunderstorm weather scenarios, (a) system load evolution under thunderstorm disturbance (b) priority-based task migration matrix at t = 45 min (see online version for colours)



(a)



(b)

In order to visually demonstrate the dynamic decision-making process and micro-scheduling behaviour of the EAR-Sys framework in real extreme scenarios, this section extracts a real case of extensive flight delays caused by extremely harsh weather conditions from the test set for in-depth analysis. This case is taken from a severe convective thunderstorm weather event that lasted for approximately two hours in a certain summer year within the dataset. During this period, the two main runways of the airport were intermittently closed, resulting in severe congestion of over 150 incoming and outgoing flights within a short period of time. As a large number of flights suddenly changed from normal status to long-term delays or diversion, the concurrent computing request volume in multiple dimensions such as the flight information display system in the terminal building, security verification, luggage sorting, and coordinated vehicle scheduling on the air side instantly increased by more than 4.5 times the normal peak, causing a devastating local impact on the underlying distributed edge computing network.

By tracking the data in Figure 7, we can clearly observe the outstanding scheduling trajectory of EAR-Sys after integrating spatio-temporal perception and RL. In the stacked area chart, during the initial stage when the thunderstorm just swept across the runway area (from $t = 15$ min to $t = 30$ min), the load area of the edge cluster of the T3 terminal building expanded explosively. At this time, the spatio-temporal graph network module quickly captured the topological feature that the flight delay kinetic energy spread from the air side to the land side. Unlike traditional baseline algorithms that accept all sudden traffic without exception until the nodes collapse, EAR-Sys adjusted the resource allocation action vector in advance and smoothly offloaded a large number of non-core background computing tasks to the central cloud server. On the area chart, it is intuitively shown that the load-up curve of the T3 cluster was forcibly flattened, while the load area of the central cloud increased accordingly. This forward-looking elastic allocation successfully reserved an extremely valuable 30% core computing power margin for the T3 terminal building.

More importantly, at $t = 45$ min when the thunderstorm was at its most intense, a key edge node of the T2 terminal building experienced a sudden hardware frequency reduction failure, causing its available computing power to sharply decrease. The actual load rate soared linearly within two seconds and exceeded the dynamic safety threshold of 95%. At this time, the lower part of the sankey diagram extremely precisely depicted the closed-loop self-healing process of the system. When the dynamic recovery strategy was activated instantly, the system immediately froze the regular task queue of the faulty node. The thick data flow in the sankey diagram clearly showed that high-priority computing tasks involving flight collision warning and core departure control were quickly stripped off and precisely routed to adjacent unimpacted cargo area nodes and remote cloud centres for relay computing; while the fine branches representing low-priority redundant requests were directly cut off or postponed at the source. This series of |textbook-like cross-region task scheduling operations not only pulled the faulty node back from the edge of failure within a few seconds, but also ensured that the most critical lifeline data flow of the airport did not experience any interruption during this unprecedented strong disturbance of the flight flow. This case, based on the detailed retrospection of stacked load and flow topology, from the micro engineering practice perspective, indisputably confirms the strong robustness and practical value of EAR-Sys in responding to black swan events.

5.5 Statistical significance test

To rigorously substantiate the empirical superiority of the proposed EAR-Sys framework, comprehensive statistical validation is systematically conducted across all performance metrics. All experimental results are reported as the mean accompanied by the standard deviation and 95% confidence intervals, derived from 50 independent simulation runs utilising diverse random seeds to strictly guarantee sample independence. Prior to executing significance testing, the Shapiro-Wilk test was applied to confirm that the distribution of performance metrics, such as task processing latency and system throughput, adequately satisfied the normality assumption. Subsequently, paired t-tests were employed to evaluate the comparative enhancements against the baseline algorithms. The statistical analysis yielded p-values strictly less than 0.01 across all critical disruption scenarios, explicitly confirming that the performance gains achieved by EAR-Sys are statistically significant rather than stochastic anomalies. Furthermore, Cohen's d was calculated to quantify the effect sizes, consistently demonstrating a large effect magnitude (Cohen's $d > 0.8$) when comparing the dynamic recovery efficiency of the proposed closed-loop mechanism against conventional static scheduling baselines, thereby unequivocally solidifying the mathematical credibility of the architectural improvements.

5.6 Scalability experiments

To comprehensively validate the structural scalability and extreme load tolerance of the EAR-Sys framework, extended empirical evaluations were conducted across expanding spatial topologies and high-density traffic scenarios. By scaling the physical edge network from a baseline single-terminal configuration to a massive multi-terminal hub encompassing up to 100 collaborative multi-agent nodes, the systemic execution efficiency was rigorously stress-tested. The empirical results demonstrate that despite the exponential expansion of the environmental state dimensionality and agent population, the real-time decentralised inference latency remains strictly bounded at the sub-millisecond level. This robust scalability is fundamentally attributed to the CTDE architecture, which effectively decouples real-time local decision-making from global state synchronisation overheads. Furthermore, under simulated ultra-dense traffic conditions where baseline

concurrent task arrival rates were amplified by an order of magnitude, the closed-loop recovery mechanism consistently sustained network throughput above 90% without triggering localised cascading failures. These findings unequivocally confirm the framework's exceptional operational resilience and its structural readiness for future large-scale, high-density civil aviation deployments.

6 Discussion

6.1 Interpretation of results

The superior performance of EAR-Sys in managing flight flow disturbances stems from three core mechanisms. First, its spatio-temporal graph network translates physical flight delays into dynamic flow fields, enabling forward-looking resource pre-allocation that avoids the response lag of algorithms like FCFS and RA-PSO. Second, the 'allocation-recovery' joint optimisation closed-loop feeds failure penalties back into the training process. This endows the agents with 'disaster awareness,' prompting them to maintain elastic buffers that significantly reduce recovery time from 28 s to 22 s. Finally, during extreme events like thunderstorms, a dynamic priority-based re-scheduling mechanism prevents secondary congestion by ensuring critical lifeline services receive prioritised computing power over the indiscriminate task migration seen in traditional methods.

To rigorously ascertain the operational resilience of the EAR-Sys framework under unexpected cyber-physical disruptions, such as communication failures, sensor anomalies, or partial network collapse, its systemic fault-tolerance mechanisms are explicitly evaluated. Fundamentally, the CTDE architecture endows the infrastructure with an innate capacity for graceful degradation. During real-time deployment, should inter-agent communication links be severed, the affected computational nodes do not succumb to global scheduling paralysis; rather, they autonomously revert to independent local execution, leveraging isolated temporal observations to sustain baseline task allocation.

6.2 Limitations

Despite its robust performance, the engineering implementation of EAR-Sys faces several objective limitations. First, its predictive accuracy heavily depends on real-time sensor data; network delays or packet loss can cause allocation failures. Second, the multi-agent architecture introduces significant communication overhead. Third, during 'black swan' events where total disturbance exceeds global computing capacity, the system faces inevitable physical bottlenecks, stalling low-priority services. Furthermore, the model's generalisation across highly heterogeneous airports and novel extreme weather requires further validation. Finally, the MARL joint training process incurs high computational costs, demanding substantial initial hardware investment and affecting convergence efficiency in large-scale clusters.

While the proposed EAR-Sys framework demonstrates robust performance under severe aviation disruptions, it is fundamentally bounded by certain theoretical assumptions and operational conditions that warrant explicit acknowledgment. Primarily, the system assumes that the physical-to-digital mapping mechanism remains stationary and that the underlying IT infrastructure maintains minimum viable network connectivity. Consequently, the operational boundary condition dictates that emergency task migration is strictly contingent upon the availability of surplus computational capacity within at least a fraction of the topologically adjacent nodes. Therefore, the framework is intrinsically susceptible to failure under catastrophic dual-layer collapse scenarios – such as a simultaneous massive physical disaster coupled with a total facility-wide power grid failure – where all distributed edge nodes synchronously lose operational capacity, rendering algorithmic load balancing physically impossible.

6.3 Practical implications

EAR-Sys holds profound significance for smart airport edge computing. It upgrades IT infrastructure to an 'active self-healing' paradigm, significantly enhancing resilience against extreme events. By minimising reliance on annotated data, it lowers deployment costs. Furthermore, its millisecond-level inference ensures real-time task continuity during disturbances, while the 'allocation-recovery' closed-loop optimises computing allocation and load balancing. Ultimately, EAR-Sys provides a solid technical foundation for future unmanned transportation infrastructure.

6.4 Future work

While EAR-Sys excels in single-airport scenarios, future research should address broader challenges. Key directions include extending the framework to cross-airport collaborative scheduling to manage cascading disturbances in multi-airport systems. Additionally, incorporating green physical constraints – such as energy efficiency and hardware fatigue – into the reward function will ensure sustainable operations. Integrating large language models with the MARL framework is another crucial step to process unstructured data and enhance decision interpretability. Finally, seamlessly migrating the system from simulations to real-world edge computing clusters for plug-and-play deployment remains a highly valuable engineering objective.

To critically evaluate the real-world deployment feasibility of the EAR-Sys framework within highly regulated civil aviation environments, several engineering constraints and operational cost factors are systematically addressed. Integration with legacy airport infrastructures, such as traditional Airport Operational Databases (AODB), is facilitated through the encapsulation of the multi-agent inference engine into containerised microservices, enabling

seamless interoperability via standardised middleware APIs without disrupting existing communication protocols.

7 Conclusions

This paper addresses the severe challenges of ineffective and slow recovery of airport computing resource allocation under extreme operating conditions, and proposes and implements EAR-Sys with dynamic perception and elastic self-healing capabilities. This framework combines the extraction of spatio-temporal graph network features, MARL, and proactive dynamic recovery strategies, breaking the logical barrier between physical space flight flow disturbances and digital space resource management, and constructing a fully closed-loop intelligent decision-making system. The main contributions of the research lie in: firstly, a physical information-enabled feature quantification model was proposed, which can accurately identify and predict the peak computing power caused by the spread of delays; Secondly, through a closed-loop training mechanism of 'allocation-recovery' collaborative evolution, the generalisation ability and collaboration efficiency of the multi-agent system in handling extreme and unstable disturbances have been significantly improved.

Experimental verification results show that EAR-Sys demonstrates outstanding practical value in ensuring the continuity of core business of smart airports. When facing strong flight flow disturbances, this system not only successfully reduced the average processing delay of the entire network computing tasks by 14.2%, but also, with its forward-looking elastic re-scheduling mechanism, significantly shortened the system recovery convergence time from the traditional baseline of 28 seconds to 22 seconds, and the task discard rate under high load conditions was only 2.1%, effectively ensuring the stable operation of the airport's digital infrastructure under extreme conditions. The improvement of these key performance indicators not only proves the scientificity and rationality of the mechanism design proposed in this paper, but also provides a new theoretical paradigm for the resilience evaluation and intelligent operation of smart airport IT infrastructure.

Looking forward to the future, as this framework expands to cross-airport collaboration and more complex environmental constraints, it will have broader application prospects and social benefits in enhancing the risk-resistance capability of large-scale civil aviation operation guarantee systems.

Declarations

All authors declare that they have no conflicts of interest.

References

Beloglazov, A. and Buyya, R. (2012) 'Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers', *Concurrency and Computation: Practice and Experience*, Vol. 24, No. 13, pp.1397–1420.

Brignardello-Petersen, R. and Guyatt, G.H. (2025) 'Introduction to network meta-analysis: understanding what it is, how it is done, and how it can be used for decision-making', *American Journal of Epidemiology*, Vol. 194, No. 3, pp.837–843.

Diener, A., Ivanov, S., Haselrieder, W. and Kwade, A. (2022) 'Evaluation of deformation behavior and fast elastic recovery of lithium-ion battery cathodes via direct roll-gap detection during calendering', *Energy Technology*, Vol. 10, No. 4, p.2101033.

Ding, J., Zhang, G., Wang, S., Xue, B., Yang, J., Gao, J., Wang, K., Jiang, R. and Zhu, X. (2022) 'Forecast of hourly airport visibility based on artificial intelligence methods', *Atmosphere*, Vol. 13, No. 1, p.75.

Du, X., Wang, T., Feng, Q., Ye, C., Tao, T., Wang, L., Shi, Y. and Chen, M. (2022) 'Multi-agent reinforcement learning for dynamic resource management in 6G in-X subnetworks', *IEEE Transactions on Wireless Communications*, Vol. 22, No. 3, pp.1900–1914.

Fei, H., Wu, S., Zhang, M., Zhang, M., Chua, T-S. and Yan, S. (2024) 'Enhancing video-language representations with structural spatio-temporal alignment', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 46, No. 12, pp.7701–7719.

Garmendia Orbegozo, A., Núñez González, J.D. and Anton, M.A. (2024) 'Task offloading in edge computing using GNNs and DQN', *Computer Modeling in Engineering & Sciences*, Vol. 139, No. 3, pp.113–128.

Gong, Y., Yao, H., Wang, J., Jiang, L. and Yu, F.R. (2021) 'Multi-agent driven resource allocation and interference management for deep edge networks', *IEEE Transactions on Vehicular Technology*, Vol. 71, No. 2, pp.2018–2030.

Gong, Y-J., Zhang, J., Chung, H.S-H., Chen, W-N., Zhan, Z-H., Li, Y. and Shi, Y-H. (2012) 'An efficient resource allocation scheme using particle swarm optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 16, No. 6, pp.801–816.

Hoang, L.T., Nguyen, C.T. and Pham, A.T. (2023) 'Deep reinforcement learning-based online resource management for UAV-assisted edge computing with dual connectivity', *IEEE/ACM Transactions on Networking*, Vol. 31, No. 6, pp.2761–2776.

Hurtado Sánchez, J.A., Casilimas, K. and Caicedo Rendon, O.M. (2022) 'Deep reinforcement learning for resource management on network slicing: a survey', *Sensors*, Vol. 22, No. 8, p.3031.

Jacobovic, R. and Mandjes, M. (2024) 'Externalities in queues as stochastic processes: the case of FCFS M/G/1', *Stochastic Systems*, Vol. 14, No. 1, pp.1–21.

Ju, Y., Song, J., Li, W., Zhang, Y., He, C., Dong, F. and Chen, C. (2025) 'Dynamic load-balancing routing strategy for leo satellite networks based on spatio-temporal traffic prediction', *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 1, pp.11–24.

Liu, X., Wang, Q., Zou, C., Yu, M. and Liao, D. (2022) 'Edge intelligence for smart airport runway: architectures and enabling technologies', *Computer Communications*, Vol. 195, pp.323–333.

Mou, Q., Liang, Q., Tian, J. and Jing, X. (2024) 'Adaptive scheduling method for passenger service resources in a terminal', *Aerospace*, Vol. 11, No. 7, p.528.

Reginald, P.J. (2025) 'Context-driven cooperative intelligent control for distributed cyber-physical actuation platforms using CTDE multi-agent reinforcement learning', *Recent Advances in Next-Generation Wireless Communication Systems*, Vol. 27, pp.43–50.

Shi, H., Liu, G., Zhang, K., Zhou, Z. and Wang, J. (2022) 'MARL Sim2real transfer: merging physical reality with digital virtuality in metaverse', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 53, No. 4, pp.2107–2117.

Sujkowski, M., Kozuba, J., Uchroński, P., Banaś, A., Pulit, P. and Gryżewska, L. (2023) 'Artificial intelligence systems for supporting video surveillance operators at international airport', *Transportation Research Procedia*, Vol. 74, pp.1284–1291.

Tong, H., Chen, Y., Weng, Y. and Zhang, S. (2022) 'Biodegradable-renewable vitrimer fabrication by epoxidized natural rubber and oxidized starch with robust ductility and elastic recovery', *ACS Sustainable Chemistry & Engineering*, Vol. 10, No. 24, pp.7942–7953.

Tran-Dang, H., Bhardwaj, S., Rahim, T., Musaddiq, A. and Kim, D-S. (2022) 'Reinforcement learning based resource management for fog computing environment: literature review, challenges, and open issues', *Journal of Communications and Networks*, Vol. 24, No. 1, pp.83–98.

Wan, Y., Li, X.C., Yuan, H., Liu, D. and Lai, W.Y. (2024) 'Self-healing elastic electronics: materials design, mechanisms, and applications', *Advanced Functional Materials*, Vol. 34, No. 27, p.2316550.

Wei, W., Li, C. and Yu, J. (2026) 'Real-time online low-complexity passenger scheduling and resource optimization for networked multi-airport systems', *IEEE Internet of Things Journal*, Vol. 6, No. 1, pp.12–26.

Wu, Y., Yang, H., Lin, Y. and Liu, H. (2023) 'Spatiotemporal propagation learning for network-wide flight delay prediction', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 36, No. 1, pp.386–400.

Yang, Y., Zhang, K., Wang, H., Fu, X., Liu, G. and Li, X. (2026) 'The solidification effect of airport road bases using an improved BP neural network and visualisation evaluation model', *International Journal of Information and Communication Technology*, Vol. 27, No. 24, pp.73–90.

Zhang, P., Li, Y., Kumar, N., Chen, N., Hsu, C-H. and Barnawi, A. (2022) 'Distributed deep reinforcement learning assisted resource allocation algorithm for space-air-ground integrated networks', *IEEE Transactions on Network and Service Management*, Vol. 20, No. 3, pp.3348–3358.

Zhang, R., Huang, W. and Ma, T. (2024a) 'Evaluation of airport intelligence degree based on zero-subjective relative algorithm', *Journal of Traffic and Transportation Engineering*, Vol. 24, No. 2, pp.232–242.

Zhang, Y., Zhou, X., Yin, J. and Tian, W. (2024b) 'Probabilistic prediction and multi-resource stochastic optimized scheduling of departure operations on the airport surface', *IEEE Access*, Vol. 12, pp.139789–139803.

Zhao, H., Chen, Y., Wang, X., Wang, D., Xu, H. and Deng, W. (2025) 'Joint optimization scheduling using AHMQDE-ACO for key resources in smart operations', *IEEE Transactions on Consumer Electronics*, Vol. 16, pp.1281–1296.