http://dx.doi.org/10.62762/TSSR.2026.164131

RESEARCH ARTICLE

Design of Low-Altitude Air Route Networks with Robustness Boundary via Reinforcement Learning

Bingyu Zhu[1], **Shanghan Li**[1] **and Yimeng Liu**[2,*]

[1] School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China
[2] Hangzhou International Innovation Institute, Beihang University, Hangzhou 311115, China

Abstract

The rapid expansion of the low-altitude economy is reshaping urban transportation systems, while large-scale operations of dynamic and high-density low-altitude unmanned aerial vehicles pose significant challenges to performance and robustness of airspace infrastructure. Traditional free-flight and point-to-point paradigms have revealed inherent limitations in conflict resolution and congestion mitigation, making the construction of adaptive, structured, and dynamic low-altitude air route networks a critical pathway to achieve both route controllability and efficient operation. However, existing design approaches may struggle to rapidly adapt to dynamic flight requirements while ensuring high system performance. Increasing demands for robustness further complicates design of air route networks, necessitating a trade-off between performance and robustness. To address this coupled challenge, a two-stage design framework is proposed based on safe reinforcement learning (safe RL), enabling the automated construction of low-altitude air route networks with high performance and permissible robustness. The framework first constructs an initial backbone network using the shortest path sets derived from origin-destination (OD) demands to guarantee basic accessibility. Then, the initial backbone network is augmented by adding a given number of edges, which is formulated as a constrained Markov decision process (CMDP). By integrating the representation capability of graph neural networks (GNNs) with the constraint-handling mechanism of safe RL, the framework achieves adaptive network design that improves system travel performance under the robustness constraint. Experimental results in Washington demonstrate that the proposed method can effectively design air route networks across different OD scenarios. Embedding the robustness constraint into the reinforcement learning (RL)-based design paradigm, this approach provides a potential pathway for the automated design of next-generation critical infrastructure for low-altitude transportation with high performance and permissible robustness.

Keywords: low-altitude air route network, network design, network robustness, safe reinforcement learning.

1 Introduction

With the acceleration of urbanization and the rapid growth of flight demands, the low-altitude economy is emerging as an important sector in urban and

Submitted: 23 February 2026
Accepted: 04 March 2026
Published: 23 March 2026

Vol. 2, **No.** 2, 2026.
10.62762/TSSR.2026.164131

***Corresponding author:**
✉ Yimeng Liu
liuyimeng94@163.com

Citation
Zhu, B., Li, S., & Liu, Y. (2026). Design of Low-Altitude Air Route Networks with Robustness Boundary via Reinforcement Learning. *ICCK Transactions on Systems Safety and Reliability*, 2(2), 54–81.

regional development. The maturity of unmanned aerial vehicle (UAV) technology is driving low-altitude UAV logistics and urban low-altitude transportation toward large-scale operations [1]. Under the trend of diversified operations, the characteristics of high density, dynamic traffic patterns, and strong constraints in low-altitude airspace are becoming increasingly prominent. On the one hand, flight activities may operate under multiple constraints such as building clusters, no-fly zones, communication coverage, and meteorological disturbances. On the other hand, a large number of concurrent missions cause free-flight and point-to-point modes to face bottlenecks in conflict resolution and congestion mitigation. In this context, constructing structured low-altitude air route networks would be a key pathway connecting route controllability and efficient operation, transforming continuous airspace into a planned network representation, which can provide infrastructure for subsequent traffic assignment and congestion evacuation [2]. Related studies further indicate that low-altitude public air routes are regarded as a comprehensive foundation for future low-altitude UAV operation and services [3].

Existing studies on low-altitude air route network design have explored a wide range of approaches, including airspace discretization, network generation, and multi-objective design trade-offs [4, 5]. One class of methods constructs multi-level air route systems through iterative design and hierarchical organization, which emphasizes using explicit rules and constraints to represent complex airspace [3]. Another class of methods discretizes continuous airspace into hierarchical grid-based networks, enabling air route encoding, querying, and conflict detection [6]. Research targeting future Urban Air Mobility (UAM) operations further leverages multi-altitude grid structures and multi-dimensional edge cost models to generate diverse sets of alternative paths, providing a foundation for choice-based networks that support subsequent traffic assignment [7, 8]. Under risk mitigation requirements, some studies construct backbone networks using obstacle information and apply heuristic methods to modify edges to ensure obstacle avoidance under route safety interval constraints [9]. To address congestion in large-scale operations, several studies explicitly integrate airspace network design with congestion modeling [10]. These works highlight that point-to-point UAV deliveries in shared airspace are prone to induce substantial conflicts and potential congestion, and congestion-aware network design can be achieved by adopting congestion functions to characterize how edge travel times increase with traffic flow. Other studies propose dual-layer network architectures, consisting of a transportation layer and a delivery layer, to mitigate structural conflict risks [11]. The above studies have established a methodological foundation for low-altitude air route network planning.

However, significant gaps remain in low-altitude air route network design with respect to the operational demands of large-scale, adversarial, and dynamic environments. On the one hand, the coupled optimization of performance and demand adaptability has not been sufficiently addressed. Many existing approaches primarily emphasize geometric feasibility for obstacle avoidance or static metrics, like path lengths, the number of crossings and coverage areas, which provide limited characterization of demand-driven network performance [6]. In particular, under concurrent multi-OD demands where route choices of users may induce congestion, system performance should be evaluated within a traffic assignment framework. One of the most notable behavioral consistency models, Wardrop user equilibrium (UE), provides a well-established theoretical basis for characterizing network states arising from selfish route choice behavior [12]. However, incorporating UE into network design results in highly nonlinear and combinatorial optimization problems, which may significantly increase computational cost of conventional heuristic or static planning methods under heterogeneous demand patterns. On the other hand, robustness considerations introduce additional requirements. Low-altitude networks may be exposed to node and edge failures, degradation of communication coverage, or even intentional attacks that disrupt critical components [13]. Robustness boundary should be incorporated into network design and evaluated through dismantling processes [14]. Selecting a limited set of edges from a candidate topology for the design of an air route network constitutes a challenging problem involving combinatorial optimization, UE, and the robustness constraint. The designed network is required to maintain robustness above a prescribed threshold while minimizing the total travel time under UE.

The integration of RL and GNNs offers a promising approach to addressing the aforementioned challenge [15]. Low-altitude air route network design can be viewed as a combinatorial optimization

process of selecting a subset from a candidate edge set, which can be formulated as a sequential decision-making problem. In this formulation, an agent incrementally adds edges and iteratively refines the network based on environmental feedback of robustness and traffic performance. Meanwhile, GNNs are capable of learning transferable graph representations across various network structures and OD demand patterns, thereby enabling generalization across different OD demand scenarios. Existing surveys indicate that GNNs have become essential tools for combinatorial reasoning and optimization on graph-structured data, and have been integrated with learning-based strategies to address a wide range of combinatorial optimization problems [16, 17]. However, directly applying unconstrained RL may fail to reliably satisfy robustness requirements during both training and deployment, and may be inadequate for handling hard constraints such as robustness boundaries. Safe RL explicitly incorporates the robustness constraint into the learning objective while pursuing long-term rewards, thereby preventing catastrophic outcomes or constraint violations [18, 19]. Safe RL has been successfully applied in various constrained and high-risk scenarios. For example, in the field of robotics, safe RL is used to prevent collisions, impacts, or other behaviors that may damage equipment or endanger people during policy learning [20, 21]. In the field of autonomous driving, safe RL is applied to lane keeping, trajectory tracking, and risk-constrained control [22, 23]. In the field of power systems, safe RL is used for safety-critical tasks such as grid voltage control and emergency control [24, 25]. This RL paradigm with robustness boundaries provides direct inspiration for hard-constrained structural design problems.

Therefore, we propose a RL-based two-stage design method for low-altitude air route networks with robustness boundaries. Given a basic topology (candidate edge set) and multiple OD demand matrices, the proposed approach first constructs an initial backbone network using the shortest path sets of OD demands. It then leverages GNN representations together with safe RL to select a given number of augmented edges from the candidate edge set. Under the constraint that the network robustness remains above a specified threshold, the method seeks to reduce the total travel time under UE for the given OD demand.

The main contributions of this paper can be summarized as follows:

1. A robustness-constrained design model for low-altitude air route networks is formulated, integrating UE-based system performance and the robustness boundary into a unified constrained optimization framework.
2. A two-stage network design approach is developed, which combines a shortest-path-based backbone generation with a safe RL–based network augmentation, enhanced by GNN-based embeddings and reward shaping. This approach improves search efficiency within the feasible region and facilitates generalization across various OD demand scenarios.
3. Experimental results under diverse OD demand patterns demonstrate that the proposed approach consistently reduces the total system travel time (TSTT) while maintaining robustness above the prescribed threshold. Experimental analysis shows that the learned augmentation strategies redistribute flows away from heavily loaded edges, alleviate congestion on critical corridors, and introduce redundant bypasses that enhance robustness margins.

The remainder of this paper is organized as follows. Section 2 introduces the problem description and preliminaries. Section 3 presents the proposed two-stage design framework and the RL-based augmentation strategy. Section 4 describes the experimental setup and analyzes the results. Section 5 concludes the paper with discussion and future directions.

2 Problem Description and Preliminaries

This section introduces the modeling foundation required for robustness-constrained low-altitude air route network design. We first define the operational scenario and UE-based performance evaluation along with multi-OD demand representation. Next, a network dismantling metric is presented to evaluate structural robustness. Then, preliminaries of safe RL are introduced. Finally, integrating the above elements, the air route network design problem is formulated as a combinatorial optimization problem with explicit robustness requirements.

2.1 Scenario and Air Route Network Modeling

2.1.1 Design Model of Low-Altitude Air Route Network

This study formulates the low-altitude air route network design as a combinatorial optimization problem, in which a limited number of edges

are selected from a given candidate edge set to construct an air route network for specified OD demands. The objective is to optimize system performance under a predefined robustness constraint. Regarding the candidate edge set, road corridors provide natural geometric and semantic boundaries that facilitate delineation, planning, and dynamic adjustment of air routes. As a core component of urban infrastructure, road networks are closely aligned with functional zoning and risk exposure patterns, enabling the integration of ground-level risk and externality considerations during airspace planning [9]. Accordingly, several studies have advocated constructing low-altitude public air routes based on ground transportation networks and selecting subsets of road corridors to support demand accommodation and congestion alleviation [10, 26]. In this study, the urban ground road network is adopted as the source of candidate edges to investigate the trade-off between system performance and structural robustness in air route network design. Specifically, the ground road network G^{road}=(V^{road},E^{road}) is mapped to the candidate topology G^0=(V,E^0), in which road intersections are treated as candidate nodes of the air route network, and road segments are represented as candidate edges.

2.1.2 OD Demand on the Air Route Network

To characterize the spatial distribution of low-altitude operations, multiple OD demands are considered to represent transportation demand on the low-altitude air route network. The origins and destinations of OD pairs are drawn from nodes of the candidate topology $G^0 = (V, E^0)$ and form the OD pair set W:

$$W \subseteq V \times V, \tag{1}$$

where each $w = (o, d) \in W$ denotes a group of flight tasks between the origin o and the destination d. Its demand is represented by $q_{od} > 0$, indicating the traffic flow from o to d. Accordingly, the OD demand is formulated as:

$$Q = \{q_{od}\}_{(o,d)\in W}. \tag{2}$$

In actual operations, OD demands exhibit heterogeneity and dynamic characteristics across different task types and spatiotemporal demand patterns. To capture representative task scenarios of the low-altitude transportation and systematically evaluate network performance across connectivity, coverage, and bottleneck congestion, this study considers three types of OD distribution patterns, including random-node-pair (RNP) [10], warehouse-customer (WC) and space-uniform (SU) [27](see Supplementary Note 1 for details of these patterns).

2.1.3 Performance Evaluation of Air Route Networks

Given the OD demands, the performance of an air route network can be evaluated via UE traffic assignment, which can describe the stable traffic state under selfish route choice of multiple agents. TSTT under UE is adopted as the primary performance metric of route networks [12, 28]. Under UE, for each OD pair, all utilized paths have equal and minimal travel times, and no user can reduce their travel cost by unilateral deviation [29, 30]. The Bureau of Public Roads (BPR) function is employed to model the nonlinear relationship between edge travel time and traffic flow [31]. For each edge $e \in E$ in a given network $G = (V, E)$ with the OD demand Q, we use f_e^{UE} to denote the flow on edge e under UE, and use $t_e(f_e^{\mathrm{UE}})$ to denote the congestion travel time of edge e based on the BPR function (see Supplementary Note 2 for details of UE and BPR). Accordingly, TSTT is defined as [32]:

$$\mathrm{TSTT}(G, Q) = \sum_{e\in E} f_e^{\mathrm{UE}} t_e(f_e^{\mathrm{UE}}). \tag{3}$$

TSTT is a standard system-level performance metric in transportation network design, which can simultaneously reflect free-flow and congestion-induced travel time [33].

2.2 Network Robustness Preliminaries

As critical infrastructure in low-altitude airspace, air route networks are required to maintain a certain level of robustness to preserve connectivity under attacks. The dismantling process is adopted to evaluate network robustness, in which nodes are progressively removed from the network G by a targeted attack, yielding a connectivity degradation sequence $\{s_m\}$ [13, 34]. s_m denotes the size of the largest connected component after the removal of the m-th node.

Robustness $R(G)$ of the network with N nodes is quantified as the relative cumulative connectivity of the sequence [34]:

$$R(G) = \frac{1}{N}\sum_{m=0}^{M} s_m, \tag{4}$$

where M denotes the total number of dismantling steps. The intentional attack strategy considered in

this study is the classical iterative highest-degree attack (HDA), which iteratively removes the node with the highest degree in the current network until the network collapses into isolated components [35].

2.3 Fundamentals of Safe RL

2.3.1 Constrained Markov Decision Process

Safe RL has been proposed for decision-making with constraints like safety, reliability, and robustness, to minimize unacceptable constraint violations while preserving favorable long-term returns [36, 37], modeled by CMDP [38]. CMDP extends the standard Markov decision process (MDP) (S, A, P, r, γ) by incorporating one or more cost functions $c_i : S \times A \to \mathbb{R}$ with their associated constraint bounds d_i, where S is the state space, A is the action space, P is the state transition probability kernel, r is the reward function, γ is the discounting factor of reward, and i is the index of the cost function. For a given policy π used to select action a_t in state s_t at step t of a decision process, the expected discounted return $J_r(\pi)$ and the expected discounted cost $J_{c_i}(\pi)$ are defined as:

$$\begin{aligned} J_r(\pi) &= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \\ J_{c_i}(\pi) &= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma_c^i c_i(s_t, a_t) \right], \end{aligned} \tag{5}$$

where γ_c^i is the discounting factor of cost. The objective of CMDP is to maximize the expected return $J_r(\pi)$ subject to $J_{c_i} \leq d_i$ for all i, that is:

$$\max_\pi J_r(\pi), \quad \text{s.t. } J_{c_i}(\pi) \leq d_i, \forall i. \tag{6}$$

2.3.2 Safe RL Algorithms

Safe RL methods for CMDPs typically handle constraints via Lagrangian relaxation, trust-region constraint, or penalty formulations. Lagrangian approaches incorporate constraints into the objective function as a soft-constrained formulation, which may suffer from repeated constraint violations due to multiplier oscillations and constraint overshooting [21, 39, 40]. Trust-region methods such as Constrained Policy Optimization (CPO) [41], Constrained Proximal Policy Optimization (CPPO) [42] and Augmented PPO [43], restrict the update step of policy to stabilize the optimization process [44, 45]. Penalty-based approaches transform constraints into penalty terms, reducing the constrained problem to a single-objective form [46, 47]. In deep safe RL, Penalized Proximal Policy Optimization (P3O) proposes a unified formulation based on exact penalty functions and the clipping objective of Proximal Policy Optimization (PPO) [45], which transforms constraints into an unconstrained problem, avoiding explicit second-order optimization [47]. Due to its empirical training stability and compatibility, P3O framework is adopted as the foundation of this study (Details in Section 3.3).

2.4 Problem Formulation

Based on the above assumptions, the problem of low-altitude air route design is defined as: constructing a structured network by selecting a given number of edges from a given candidate topology to optimize system performance while satisfying the given robustness boundary. Specifically, the performance is evaluated by TSTT under UE. The robustness is measured by the dismantling-based metric, with requirements above a specified threshold. The design variable corresponds to discrete edge selection, and performance evaluation involves solving a UE-based traffic assignment and computing the dismantling-based robustness metric for each candidate network. On this basis, this section formulates the low-altitude air route network design as an optimization problem of network design with a robustness constraint. The inputs, design variables, optimization objectives and constraints are detailed below.

2.4.1 Input and Design Variable

Input

(1) Candidate topology: A directed graph $G^0 = (V, E^0)$ derived from the ground transportation network, where V denotes candidate nodes and E^0 denotes candidate edges;

(2) Features of each candidate edge $e \in E^0$: capacity ρ_e, free-flow speed v_e, and edge length l_e. The corresponding free-flow travel time t_e^0, calculated as $t_e^0 = \frac{l_e}{v_e}$;

(3) BPR parameters: α_{BPR} and β_{BPR};

(4) OD demand: $Q = \{q_{od}\}_{(o,d)\in W}$, where $W \subseteq V \times V$ is the OD pair set;

(5) Budget: the number of edges to be added B;

(6) Robustness constraint: robustness threshold R_{thr}.

Design Variable

The set of augmented edges ΔE selected from the

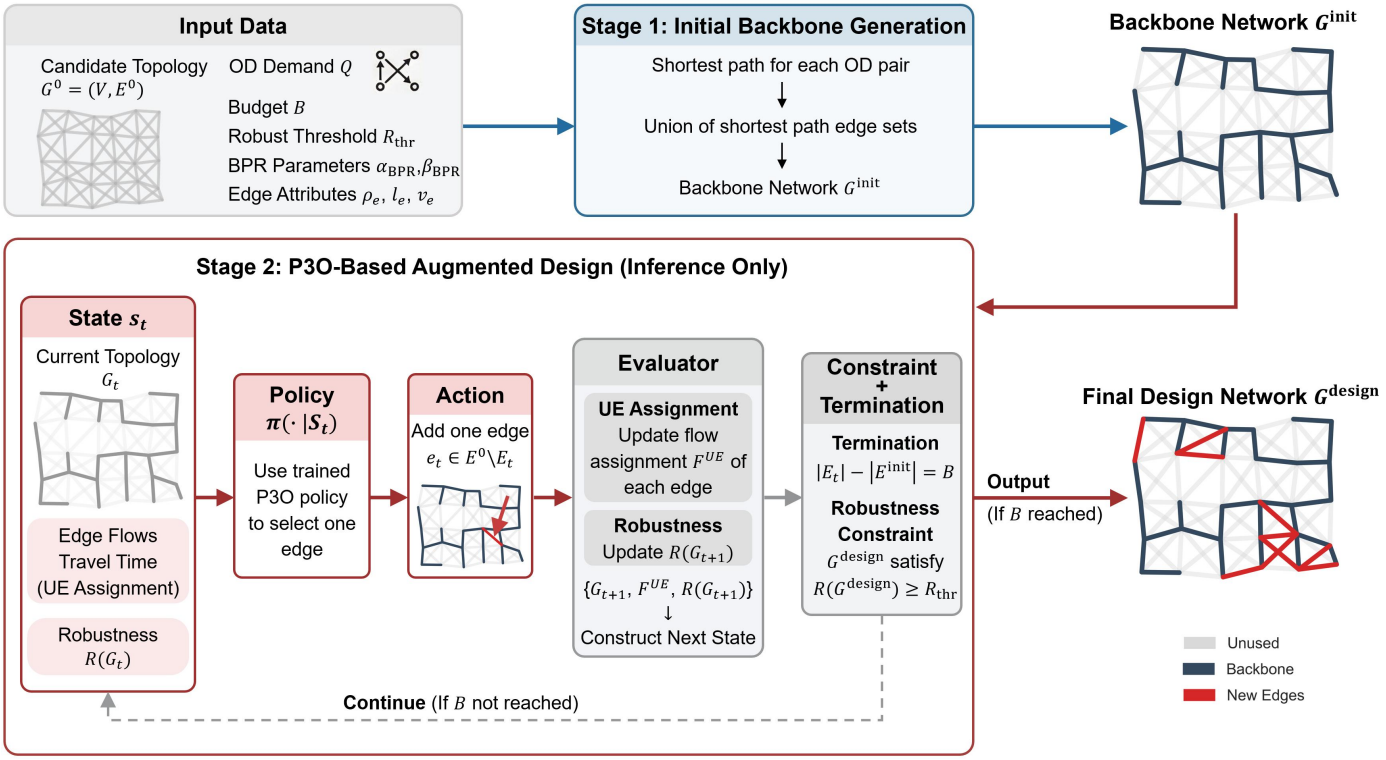


Figure 1. Overview of the proposed two-stage framework.

candidate edge set E^0 is denoted as:

$$\Delta E \subseteq E^0. \tag{7}$$

The designed network G^{design} is defined as:

$$G^{\text{design}} = (V, \Delta E). \tag{8}$$

2.4.2 *Optimization Objective and Constraints*

Optimization Objective

TSTT under UE is adopted to evaluate system performance TSTT^{UE}. Accordingly, for OD demand Q, the design objective is to select an augmented edge set ΔE such that TSTT of the designed network G^{design} is minimized:

$$\min_{\Delta E \subseteq E^0} \text{TSTT}^{\text{UE}}(G^{\text{design}}, Q), \quad G^{\text{design}} = (V, \Delta E). \tag{9}$$

For ease of notation, TSTT^{UE} is abbreviated as TSTT in the remainder of this paper, unless otherwise specified.

Constraints

The constraints consist of budget and robustness. For the budget constraint, the total number of selected augmented edges should not exceed the specified limit B:

$$|\Delta E| = B. \tag{10}$$

For the robustness constraint, the designed network should satisfy a minimum robustness requirement:

$$R(G^{\text{design}}) \geq R_{\text{thr}}, \tag{11}$$

where $R(\cdot)$ denotes the robustness metric (see Eq. (4)). The overall optimization problem is formulated as:

$$\begin{aligned} \min_{\Delta E \subseteq E^0} \quad & \text{TSTT}(G^{\text{design}}, Q) \\ \text{s.t.} \quad & R(G^{\text{design}}) \geq R_{\text{thr}}, \\ & |\Delta E| = B, \\ & G^{\text{design}} = (V, \Delta E). \end{aligned} \tag{12}$$

3 Methodology

To address the above problem, a two-stage learning-based design method is proposed, combining reachability backbone and constrained augmentation. Specifically, an initial backbone network that guarantees basic reachability and reasonable performance is first constructed. Subsequently, under the robustness constraint, the network structure is progressively augmented using GNNs and safe RL, with the objective of improving system performance. The overall framework comprises two components: (1) initial backbone generation based on shortest paths; and (2) P3O-based augmentation design. An overview of the proposed framework is shown in Figure 1.

3.1 Two-Stage Design Framework

3.1.1 Design Motivation

A basic feasibility requirement is OD reachability: for each OD pair (o, d), there must exist at least one feasible path from o to d in G^{design}. This implies that TSTT of networks with unreachable OD pairs cannot be evaluated during the early design process, which creates two challenges: (1) feasibility issue: random or greedy selection strategies in the early training may lead to infeasible network designs; (2) learning difficulty: in the early design, evaluating the quality of early actions is hard, leading to unstable training signals. Therefore, this paper adopts a two-stage design paradigm: the first stage provides a feasible starting point; the second stage performs robustness-constrained performance improvement.

3.1.2 Overview of the Two-Stage Framework

Given a candidate topology $G^0 = (V, E^0)$ and an OD demand $Q = \{q_{od}\}$, the air route network design problem is formulated as a two-stage process as follows.

Stage I Initial backbone generation: On the candidate topology G^0, for the given OD demand, the union E^{init} of edges in shortest paths covering all OD pairs is constructed to form an initial backbone network $G^{\text{init}} = (V, E^{\text{init}})$. This network guarantees reachability for all OD pairs and yields a reasonable baseline under free-flow conditions.

Stage II P3O-based augmented design: Starting from G^{init}, additional edges are progressively selected from the remaining candidate set $E^0 \setminus E^{\text{init}}$ through a sequential decision-making process. After each augmentation step, the improvements of the system performance and its robustness are evaluated. This stage is formulated as a CMDP with robustness boundary, and the P3O algorithm is employed to learn the design policy.

3.1.3 Initial Backbone Generation

The initial backbone network is constructed as the union of edges in the shortest paths for all OD pairs, which provides three advantages: (1) guaranteeing global OD reachability: ensuring that feasible paths exist for all demand pairs at the design starting point; (2) reducing the dimensionality of the search space: shrinking the design starting point from the complete candidate network to a demand-relevant subgraph; (3) providing a robustness baseline: the robustness of the backbone network can serve as a reference baseline of robustness thresholds.

The specific construction process of the initial backbone network is as follows. First, on the candidate topology $G^0 = (V, E^0)$, the shortest path problem is solved for each OD pair (o, d):

$$p_{od}^* = \arg\min_{p \in P_{od}} \sum_{e \in p} t_e^0, \tag{13}$$

where p_{od}^* denotes the shortest path from o to d, P_{od} denotes the set of all feasible paths from o to d, and t_e^0 denotes the free-flow travel time. Then, the edge sets of all OD shortest paths are taken as a union to obtain the initial backbone edge set:

$$E^{\text{init}} = \bigcup_{(o,d) \in W} p_{od}^*. \tag{14}$$

The initial backbone network is constructed as:

$$G^{\text{init}} = (V, E^{\text{init}}). \tag{15}$$

This backbone network ensures the shortest travel time for each OD pair under free-flow conditions. However, under actual traffic flows, G^{init} generally does not yield optimal performance. Under concurrent multiple OD demands, the aggregation of traffic along single shortest paths can lead to significant congestion. In addition, this structure may exhibit critical vulnerable components. Consequently, the backbone is employed solely as a feasible initial solution, and would be further improved through subsequent augmentation during the second-stage learning-based design.

3.2 P3O-based Augmented Design

After the backbone generation, the network augmentation process is modeled as a sequential decision-making problem with a robustness constraint and solved based on the P3O algorithm. Obtaining the initial backbone network, the design problem of augmentation process can be transformed into:

$$\begin{aligned} \min_{\Delta E^{\text{aug}} \subseteq E^0 \setminus E^{\text{init}}} \quad & \text{TSTT}(G^{\text{design}}, Q) \\ \text{s.t.} \quad & R(G^{\text{design}}) \geq R_{\text{thr}}, \\ & |\Delta E^{\text{aug}}| = B - |E^{\text{init}}|, \\ & G^{\text{design}} = (V, E^{\text{init}} \cup \Delta E^{\text{aug}}) \end{aligned} \tag{16}$$

where ΔE^{aug} denotes edges added during the augmentation process. For notational convenience, ΔE is used to represent ΔE^{aug}, and B' denotes the fixed augmentation budget $|\Delta E^{\text{aug}}|$. The P3O-based augmented design is introduced in the following four aspects: CMDP modeling, model training, model architecture, and network design.

3.2.1 CMDP Modeling

The network augmentation problem is modeled as a CMDP with a robustness constraint. In our design, the constraint corresponds to the robustness boundary requirement of the network. The detailed modeling process is described as follows:

1. **State space** S

At each decision step t, the state s_t is the current network $G_t = (V, E_t)$, where $E_t = E^{\text{init}} \cup \Delta E_t$ and ΔE_t denotes the augmented edge set at time t, with the UE flow distribution $F^{\text{UE}} = \{f_e^{\text{UE}}, \forall e \in E_t\}$ and robustness $R(G_t)$ under HDA attacks. Accordingly, the state space is defined as $S = \{G_t, F^{\text{UE}}, R(G_t)\}$.

2. **Action space** A

The action a_t represents an edge $e_{\text{new}} \in E_0 \setminus E_t$ added to the current network, transitioning the system to the next state s_{t+1}, with the augmented edge set updated as $\Delta E_{t+1} = \Delta E_t \cup \{a_t\}$. The action space A is the set of remaining candidate edges $A = E_0 \setminus E_t$.

3. **Reward function** r

The reward function r_t measures performance improvement of each action, which consists of two parts: (1) Performance reward, defined as the difference in TSTT before and after executing action:

$$r_t^{\text{task}} = \text{TSTT}(G_t, Q) - \text{TSTT}(G_{t+1}, Q). \tag{17}$$

where G_t denotes the network at state s_t. In the optimization progress of network design, TSTT is expected to decrease. (2) Robustness reward, used to evaluate the robustness change of the designed network after augmentation, which is defined as:

$$r_t^{\text{robust}} = R(G_{t+1}) - R(G_t). \tag{18}$$

This reward function encourages the agent to select edges with robustness improvement. The robustness reward is activated only when the network does not satisfy the robustness constraint. When the network robustness exceeds the robustness constraint, the reward is clipped:

$$r_t^{\text{robust}} = \min(R(G_{t+1}), R_{\text{thr}}) - \min(R(G_t), R_{\text{thr}}). \tag{19}$$

In summary, the overall reward function r_t is defined as:

$$r_t = r_t^{\text{task}} + w_{\text{shape}} \cdot r_t^{\text{robust}}, \tag{20}$$

where w_{shape} is a balancing factor that controls the trade-off between the performance reward and the robustness reward.

4. **Cost function** c

The cost function c_t quantifies whether the robustness constraint is violated during the design process. In this problem, the robustness constraint is a hard constraint, which requires that the robustness of the final designed network is not lower than the threshold R_{thr}. Because this constraint is only evaluated at the terminal state of each episode, the cost signal is sparse. To address this issue, we reformulate the robustness threshold constraint as an incremental robustness constraint, thereby decoupling the sparse terminal cost into a process-level formulation:

$$c_t = -(R(G_{t+1}) - R(G_t)). \tag{21}$$

Then, the cumulative cost of a complete episode is:

$$C = \sum_{t=0}^{T-1} -(R(G_{t+1}) - R(G_t)) = -(R(G_T) - R(G_0)), \tag{22}$$

where T is the episode length and G_0 corresponds the initial network G^{init}. Given $R(G_0)$, the terminal robustness threshold constraint in this design process is equivalent to a terminal robustness increment constraint:

$$\begin{aligned} R(G_T) \geq R_{\text{thr}} &\iff R(G_T) - R(G_0) \geq R_{\text{thr}} - R(G_0) \\ &\iff C \leq -(R_{\text{thr}} - R(G_0)). \end{aligned} \tag{23}$$

This cost function design encourages decisions with high robustness improvement. As an episode terminates, the cumulative cost can evaluate whether the design result satisfies the robustness constraint.

3.2.2 Model Architecture

The Actor–Critic architecture and dual value estimation are adopted in the framework. The Actor network samples an action a_t from the candidate edge set $E^0 \setminus E_t$ based on the state s_t. Two Critic networks separately estimate the reward value function $V^r(s_t)$ and the cost value function $V^c(s_t)$. In implementation, the reward Critic $V^r(\phi^r)$ and the cost Critic $V^c(\phi^c)$ adopt two isomorphic but parameter-independent networks. The detailed model architecture is illustrated below.

1. Graph Encoder

The encoder modules of both the Actor and the Critic share the same architecture, which consists of a linear projection followed by multi-layer graph attention-based message passing. Specifically, the input node feature matrix X is first mapped to a hidden representation via a linear transformation. The resulting node embeddings are then normalized using node-wise l_2 normalization to improve training stability [48]:

$$H^{(0)} = \mathrm{Norm}(XW_0). \tag{24}$$

Subsequently, L layers of GATv2 are stacked using a multi-head attention mechanism. The first $L-1$ layers employ 8 attention heads, while the final output layer uses a single head. For all layers except the last, a nonlinear activation function is applied after message propagation, followed by an additional normalization step. The propagation operation of each GATv2 layer is defined as follows [49]:

$$\mathbf{x}_i' = \sum_{j \in \mathcal{N}(i)\cup\{i\}} \alpha_{i,j}\Theta_i\mathbf{x}_j + \Theta_a\mathbf{x}_i \tag{25}$$

$$\alpha_{i,j} = \frac{\exp\left(\mathbf{a}^\top \mathrm{LeakyReLU}(\Theta_s\mathbf{x}_i+\Theta_t\mathbf{x}_j+\Theta_c\mathbf{x}_{e_{i,j}})\right)}{\sum_{k\in\mathcal{N}(i)\cup\{i\}} \exp\left(\mathbf{a}^\top \mathrm{LeakyReLU}(\Theta_s\mathbf{x}_i+\Theta_t\mathbf{x}_k+\Theta_c\mathbf{x}_{e_{i,k}})\right)}. \tag{26}$$

where $\mathbf{x}_i'$ denotes the output of node i in one GATv2 layer, $\mathbf{x}_i$ denotes the input feature of node i at the current layer, $\mathbf{x}_{e_{i,j}}$ denotes the feature of edge (i,j), $\mathcal{N}(i)$ represents the set of neighbors of node i, and $\mathbf{a}$ and $\mathbf{\Theta}$ are learnable parameters. The final node embedding is obtained as $\mathbf{h}_i \in \mathbb{R}^d$, where the embedding of the virtual node v^* is taken as the graph-level embedding $\mathbf{g}$ (see details of input features and virtual node in Supplementary Note 3).

2. Actor Network

The task of the Actor is to map the current network state to a score for each feasible candidate edge $e = (u, v) \in E^0 \setminus E_t$, and form a discrete action selection probability distribution $\pi_\theta(a_t|s_t)$. After graph encoding, the Actor takes the embedding of the virtual node as the graph-level vector $\mathbf{g}$, and applies an additional linear transformation followed by an activation layer to the node embeddings to enhance representation capacity. Meanwhile, the robustness increment constraint value d is mapped to a vector $\mathbf{c}$ through an MLP-Tanh-MLP architecture and broadcast to each edge. Therefore, for each candidate edge $e = (u, v)$, its input vector $\mathbf{z}_e$ is given as:

$$\mathbf{z}_e = [(\mathbf{h}_u - \mathbf{h}_v), (\mathbf{h}_u + \mathbf{h}_v), \Theta\mathbf{x}_e, \mathbf{g}, \mathbf{c}], \tag{27}$$

where Θ is a learnable parameter and $\mathbf{x}_e$ denotes the feature of edge e. Then, the edge representations are processed by a three-layer MLP f_{actor} (with ReLU activation function [50] applied between layers), to output the logit of each edge logit_e:

$$\text{logit}_e = f_{\text{actor}}(\mathbf{z}_e). \tag{28}$$

Finally, a softmax operation is applied to the logits of all candidate edges to obtain the action distribution $\pi_\theta(\cdot|s_t)$.

3. Critic Network

The Critic adopts the same encoder structure as the Actor, but its output is a graph-level scalar value and does not score candidate edges. Similarly, the embedding of the virtual node is taken as $\mathbf{g}$, and the robustness increment constraint value d is mapped to a vector $\mathbf{c}$. Then they are concatenated as $[\mathbf{g}, \mathbf{c}]$. The concatenated vector is subsequently fed into a decoder composed of a two-layer MLP (with Leaky ReLU activation functions [51] between layers) of the Critic to output the scalar value $V(s_t)$. In the P3O architecture, two Critic networks are constructed separately to output $V^r(s)$ and $V^c(s)$.

3.2.3 Model Training

To achieve a stable balance between performance optimization and robustness constraint satisfaction, P3O is adopted for training during the network augmentation stage. The key idea of P3O is to convert the hard constraints of the CMDP into unconstrained optimization objectives via exact ReLU-based penalty terms. In addition, the clipped surrogate objective of PPO [45] is employed to simultaneously limit the update magnitudes of both the reward-related and constraint-related policy. This improves constraint satisfaction in practice and stabilizes policy updates within a first-order optimization framework [47]. The overall training procedure follows a structure of an outer-loop sampling, inner-loop proximal policy updates and value function regression.

During training, trajectories are sampled by iteratively selecting augmentation edges until the predefined budget is reached. For all state-action pairs, rewards and cost signals are recorded and used to compute policy advantages. We adopt the generalized advantage estimation (GAE) to reduce variance in both reward and cost learning. Policy parameters are updated via a clipped surrogate objective combined with an exact penalty formulation to enforce the robustness constraint [47]. Separate value networks

are trained for reward and cost estimation to stabilize constrained policy optimization. Detailed trajectory sampling, advantage computation, and parameter update procedures are provided in the Supplementary Note 4.

3.2.4 *Inference Design*

After training, the learned P3O policy is used to generate an augmented air route network under new OD demands. In the inference stage, starting from the initial backbone network G^{init}, B' candidate edges are gradually selected and added to the candidate topology G^0 to obtain the final designed network G^{design}. After each edge addition, a UE traffic assignment is performed on the current network to update link flows and travel times. The updated network is then used to construct the graph state for the next decision step, yielding an iterative loop of edge selection, traffic assignment, and state update until the edge-budget constraint is met.

For action selection, we employ an inference strategy in which the Actor proposes candidates and the Critic guides the search, mitigating the myopic behavior of purely greedy selection. Specifically, at each inference step, the Actor outputs edge-selection probabilities over the feasible action set and selects the top-K edges as candidate expansions. For each expanded candidate state, the reward Critic value estimate $V^r(s)$ is used for scoring, and the top K branches with the highest scores are retained for the next step [52]. To ensure inference robustness, the algorithm always retains the pure Actor greedy sequence as a fallback branch, avoiding result degradation due to value estimation errors. After the prefix search reaches the preset depth, the remaining augmented edges are completed using the Actor greedy strategy to control inference overhead. After inference, TSTT under UE is calculated for the final network and the robustness $R(G^{\text{design}})$ is evaluated. If $R(G^{\text{design}}) \geq R_{\text{thr}}$, the design satisfies the robustness boundary constraint; otherwise, it is judged infeasible. When multiple candidate networks satisfy the robustness constraint, we select the one with the minimum TSTT, thereby obtaining a better operational performance result under the robustness boundary.

4 Experimental Study

4.1 Experimental Setup

This study evaluates the proposed approach through experiments based on a future low-altitude air route network design scenario over the city of Washington. The candidate topology $G^0 = (V, E^0)$ with $|V| = 192$ and $|E^0| = 604$ is constructed by vertically projecting the ground transportation network of Washington into the airspace as illustrated in Figure 2. The detailed experimental settings are described below.

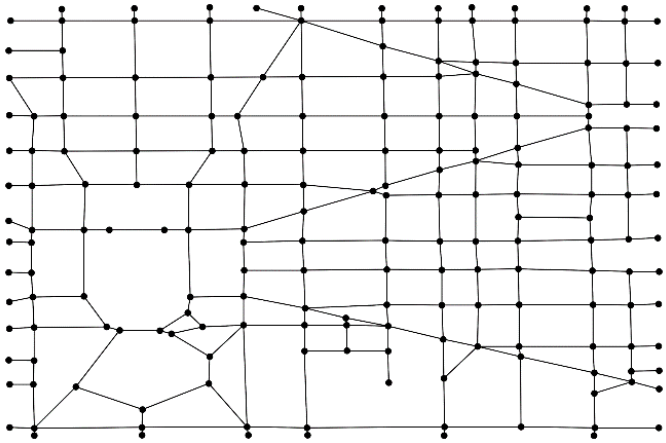


Figure 2. Candidate topology mapped from the Washington road network. Each edge is a bidirectional edge.

4.1.1 *Parameters of Candidate Topology*

Capacity is set to $\rho_e = 500$, $\forall e \in E^0$;

The free-flow speed is $v_e = 11\,\text{m/s}$ for all $\forall e \in E^0$;

The mean edge length is $\mathbb{E}[l_e] = 119.94\,\text{m}$, with range $l_e \in [26.43, 371.67]\,\text{m}$;

The BPR congestion parameters are set to $\alpha_{\text{BPR}} = 0.15$ and $\beta_{\text{BPR}} = 4$.

4.1.2 *OD Demand* $Q = \{qod\}_{(o,d)\in W}$

For each demand scenario, the number of OD pairs is set to $|Q| = 100$, and the average demand per OD pair is set to $\mathbb{E}[q_{od}] = 150$, resulting in a total demand of $\sum_{(o,d)\in W} q_{od} = 15000$. Three types of OD patterns are considered in the experiments, and their generation mechanisms are described as follows:

1. **RNP**: OD pairs are uniformly sampled from the entire set of nodes. The total demand is then evenly allocated among all selected OD pairs, forming a multi-OD demand [53].

2. **WC**: The top 3% of nodes with the highest degrees in the network are selected as the warehouse set H, and the remaining nodes constitute the customer set U. Origins of OD pairs are sampled from the warehouse set, and destinations are sampled from the customer set, in order to simulate a logistics demand structure from trunk supply points to end customers [54]. Specifically, for each warehouse $h \in H$, the shortest path cost c_{hu} to all customers u

is calculated, and distance-decay-based weights are constructed as follows:

$$\omega_{hu} = O_h D_u \exp(-\beta c_{hu}), \tag{29}$$

where O_h and D_u are randomly generated node weights, and β is the reciprocal of the average shortest path cost [55, 56]. We sample 100 OD pairs $h \to c$ according to the weight distribution, and the demands of the selected OD pairs are scaled proportionally according to the weights to match the total demand.

3. **SU**: Distance-decay-based weights are first constructed based on the shortest path costs between all node pairs in the network [54]:

$$\omega_{od} = \exp(-\beta c_{od}), \tag{30}$$

where β is the reciprocal of the average shortest path cost. A given number of OD pairs are then sampled according to this weight distribution, and the demands of the selected OD pairs are scaled proportionally according to the weights to match the total demand.

For each demand scenario, 100 OD demand matrices are generated for training a RL agent. Inference-based network design is then evaluated on an additional 30 set of OD demands that are distinct from those used during training.

4.1.3 Design Cost and Robustness Threshold

In our experiments, an initial backbone network $G^{\text{init}} = (V, E^{\text{init}})$ is first generated using the shortest path method. Based on this backbone network, the cost of network augmentation is set to $B' = 50$, meaning that in each episode the agent can select at most 50 edges from the candidate edge set to be added to the initial backbone network. Similarly, the robustness constraint threshold R_{thr} is calculated based on the number of edges in the backbone network $|E^{\text{init}}|$, the design cost B', and the robustness ratio coefficient κ as:

$$R_{\text{thr}} = \kappa \cdot R(G^{\text{init}}) \cdot \frac{|E^{\text{init}}| + B'}{|E^{\text{init}}|}. \tag{31}$$

In the experiments, the ratio coefficient is set to $\kappa = 1$.

4.1.4 Learning algorithm configuration

1. Hyperparameters of P3O Training

Reward discount factor $\gamma = 0.9$; Cost discount factor $\gamma_c = 0.9$; GAE parameter $\lambda = 0.95$; Learning rates $\eta_\pi = 0.0005$, $\eta_{vr} = 0.0001$, and $\eta_{vc} = 0.0001$; Clip coefficient $\varepsilon = 0.2$; Number of policy update iterations per round iter = 10; Buffer capacity 2500.

2. Parameters of Model Architecture

(1) Graph Encoding Layers

The dimension of node features is 9. The dimension of edge features is 4. The dimension of node linear embedding layer is 64. The number of GATv2 layers is 3. The numbers of attention heads is 8, 8, and 1. The output dimension of each head is 64. The activation function between GATv2 layers is ReLU [50].

(2) Conditional Signal Encoder

The dimension of robustness signal is 1. The dimension of linear layer output is 8. The activation function between layers is Tanh [57].

(3) Actor Regression Head

Post-processing of node embeddings: the dimension of linear layer output is 64, followed by a ReLU activation layer; Edge attribute projection: the dimension of linear layer output is 64; Edge scoring layers is 3-layer MLP with output dimensions of 16, 16 and 1, and the activation function ReLU between layers.

(4) Critic Regression Head

It is a 2-layer MLP with dimensions of 16 and 1, and the activation function between layers is Leaky ReLU [51].

4.1.5 Baseline Method

To provide a non-learning-based combinatorial optimization baseline, this paper implements a simulated annealing (SA) method for comparison in air route network augmentation design [58, 59]. SA shares the same candidate topology and evaluation modules with the proposed method, ensuring fairness of comparison. The detailed configuration of the SA baseline is summarized as follows.

Input: the candidate network and initial backbone network, OD demands, and augmentation budget.

Output: an augmented network design G_{SA}^{design} satisfying the robustness threshold constraint, and its corresponding TSTT(G_{SA}^{design},Q).

In practical experiments, to improve algorithm efficiency and stability, the initial backbone generation is also incorporated into the SA algorithm.

4.2 Evaluation Metrics

In these experiments, to comprehensively evaluate the performance of the low-altitude air route network design method, we adopted the following three main evaluation metrics: TSTT, robustness constraint satisfaction, and design process runtime.

1. TSTT is the core metric for evaluating the effectiveness of the network design, representing the total travel time of all route tasks under UE conditions (Eq. (3)). In each OD demand scenario, we evaluate the performance of the design method by the average TSTT across multiple OD demands.

2. Robustness constraint satisfaction is used to evaluate whether the designed network can maintain a certain level of robustness when facing node attacks. The network robustness is evaluated based on its connectivity, with the robustness constraint requiring that the robustness of the final designed network exceeds a specified threshold. Specifically, we report the robustness values from multiple design results.

3. Design process runtime is a key metric for evaluating design methods, especially in modern systems with rapidly changing demand patterns. We evaluate the time required for the algorithm to complete the augmentation process from the initial backbone network to the final design network, given the candidate edge set and design budget. The design time here reflects the runtime for a single-instance design (i.e., generating an augmentation plan), excluding the offline training cost of the RL model. All runtimes are measured on CPU with AMD EPYC 9654 96-Core Processor.

4.3 Design Results Under Different Scenarios

This section presents the design results of the proposed method under three typical demand scenarios (RNP, WC, and SU). We analyze the RNP scenario in depth (topology evolution, flow distribution, and route changes) and then discuss adaptive behaviors in WC and SU, focusing on congestion hotspots and wide-area coverage, respectively.

4.3.1 Design Results under the RNP Scenario

The RNP scenario simulates decentralized, on-demand point-to-point traffic demands in urban environments, characterized by random flow directions and widespread distribution. Figure 3 illustrates the OD distribution in this scenario, with red points representing origins and blue points representing destinations, scattered across the network. The gray lines indicate potential traffic flows, showing a complex and decentralized pattern. Under this demand scenario, we compare the performance differences between the initial backbone network and the RL-based augmented design network.

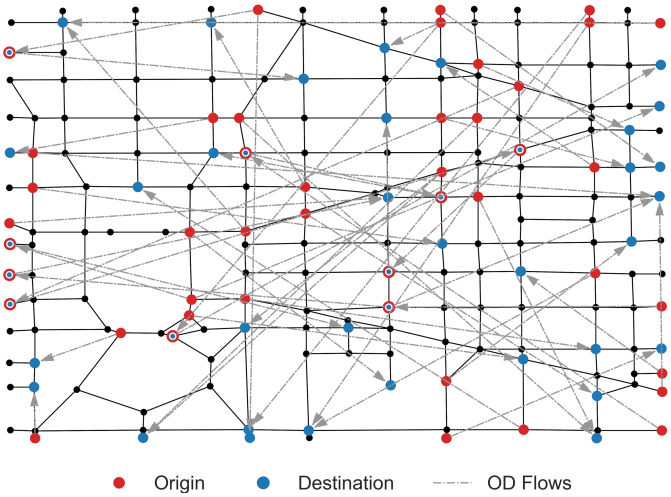


Figure 3. OD distribution of the RNP scenario. Only 50 out of the 100 OD pairs are shown.

1. Initial Backbone Network

Under this OD demand, the backbone network constructed by the union of shortest paths (Figure 4(a)) guarantees the reachability of all OD pairs, with TSTT of 1.90×10^6s. However, under UE assignment, evident corridors of traffic loads emerge: high traffic flows concentrate on a small number of diagonal corridors, and the top-5 highest-flow edges form potential bottlenecks (Figure 4(b)). This structure, where a limited number of critical edges carry a substantial share of cross-regional demand, exposes the system to two coupled risks. On the one hand, persistent congestion along these corridors increases the overall travel time. On the other hand, strong dependence on critical edges renders the network vulnerable to attacks or failures, potentially leading to a rapid degradation of network connectivity.

2. Augmented Design Network

Based on the RL-based method, 50 edges are added to the backbone network, resulting in TSTT of 1.69×10^6s, which is a reduction of 11.24% compared to the initial backbone network. Rather than simply densifying existing main corridors, the added edges reinforce the network through distributed, multi-point enhancements. Specifically, they introduce lateral connections, local loops, and inter-corridor bypasses, thereby expanding the set of alternative paths available under UE conditions (Figure 5(a)). These structural modifications are associated with operational performance. Tracking the top-5 highest-flow edges in the backbone network reveals that all of them experience load reductions in the augmented design, with the most heavily loaded edge exhibiting a 45.71% decrease in flow

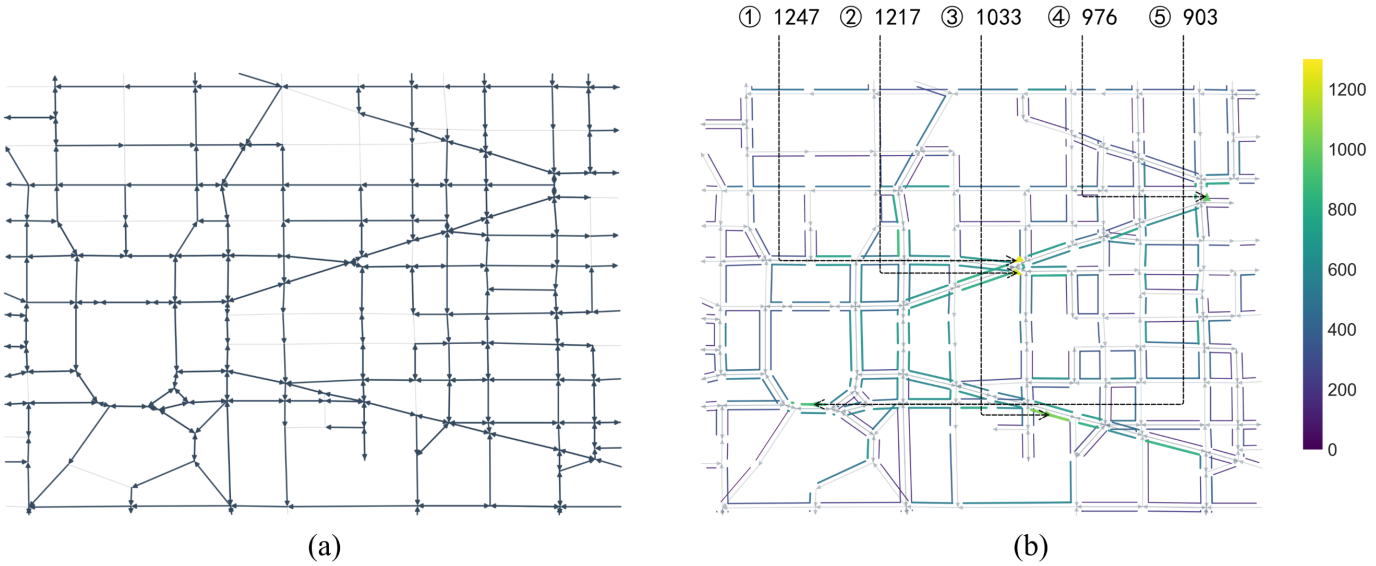


Figure 4. Backbone network design results under the RNP scenario. (a) Backbone network. $|E^{init}| = 381$. Light gray edges indicate candidate edges, whereas dark gray edges indicate backbone edges. Edges with a single arrow are one-way edges, while edges with arrows at both ends are bidirectional edges. (b) Flow distribution of the backbone network. The flow values of the top-5 highest-flow edges are annotated. Edges following the viridis colormap indicate the magnitude of traffic flow with layout following the right-hand traffic rule.

(Figure 5(b)). From a global perspective, the edge flow distribution shifts leftward compared with the backbone, with the mean flow decreasing from 3.64×10^2 to 3.19×10^2, representing an overall reduction of approximately 12.36% (Figure 5(c)). Moreover, the right tail of the high-flow region shows a contraction (Figure 5(c)), suggesting that both the number and intensity of highly loaded edges are reduced. We further compare routes for the six OD pairs that experience the largest reductions in travel time (Figure 6). All six OD pairs achieve travel time improvements, with a maximum absolute reduction of 60 s (OD Pair 1) and a maximum relative reduction of 38.66% (OD Pair 5). Notably, for OD Pairs 1, 2, and 6, the actual path length increases while the travel time still decreases, indicating that the time savings mainly arise from avoiding congested segments and redistributing traffic flows rather than from purely geometric shortening. This finding is consistent with the system-level observations of load reduction on critical edges and overall load dispersion.

3. Changes in Structural Characteristics and Robustness

From a structural perspective, the augmented network exhibits a more decentralized structure. Most newly added edges are located in low-betweenness regions, and some of these added edges carry a moderate level of flow (Figure 7(a)). This indicates that the augmented edges weaken the dependence on a small number of critical edges by providing several bypasses and redundant connections, effectively reducing the flow on high-betweenness edges.

Moreover, this low-betweenness addition strategy also shifts the overall distribution of edge betweenness centrality leftward, with the mean decreasing from 2.03×10^{-2} to 1.69×10^{-2}, corresponding to a reduction of about 16.75% (Figure 7(b)). Furthermore, we provide a comparison of the network robustness collapse curves (Figure 7(c)), where the robustness threshold is $R_{thr} = 12.30$. The robustness of the augmented design network is $R(G^{design}) = 15.76$, exceeding the threshold, thereby achieving a feasible design result under the constraint. The augmented network maintains a larger connected component in the early stages of node removal, delaying the connectivity degradation process. The decentralized structural adjustments with redundant bypasses learned by RL not only improve UE performance, but also enhance the overall robustness of the network against targeted attack.

4.3.2 *Design Results under the WC Scenario*

The WC scenario strengthens the concentration and adversarial nature of demand (Figure 8(a)), making the network more likely to exhibit an operating state in which a small number of critical edges carry most of the traffic flow. Under the same total demand, compared with the RNP scenario, the number of edges in the backbone network under the WC scenario is significantly reduced (Figure 8(b)), which suggests

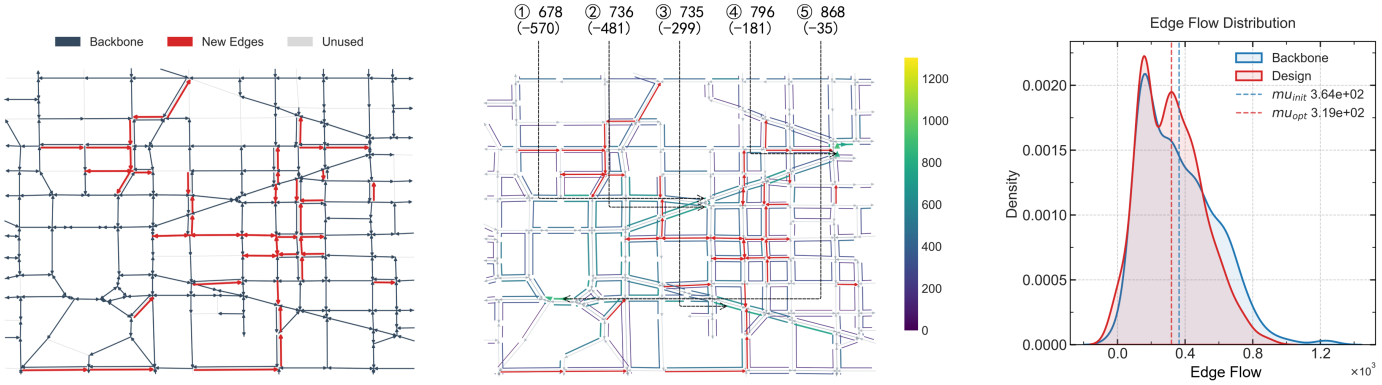


Figure 5. Augmented design results under the RNP scenario. (a) Augmented design network. Edges with a single arrow are one-way edges, while edges with arrows at both ends are bidirectional edges. (b) Traffic flow distribution of the augmented design network. The flow values and decreases of the top-5 highest-flow edges in the backbone network are annotated. Red edges indicate new edges. Edges following the viridis colormap indicate the magnitude of traffic flow with layout following the right-hand traffic rule. (c) Comparison of flow distributions between the backbone network and the augmented design network.

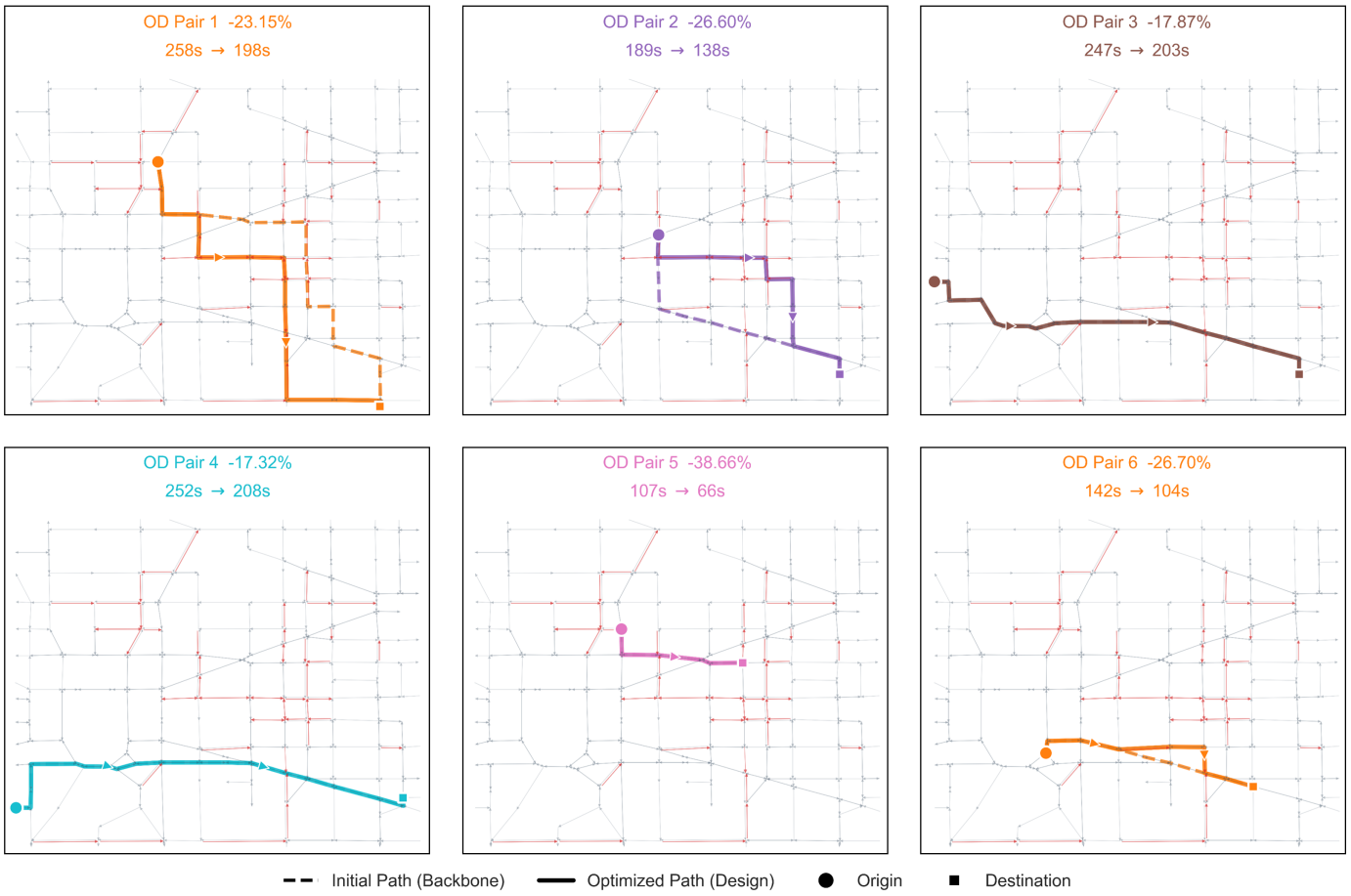
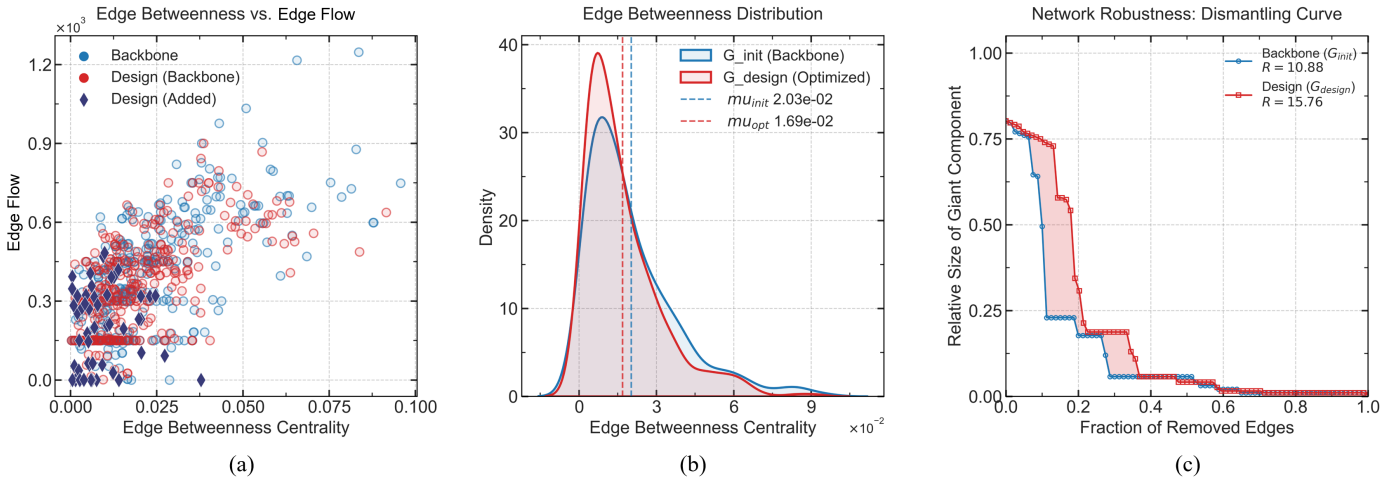


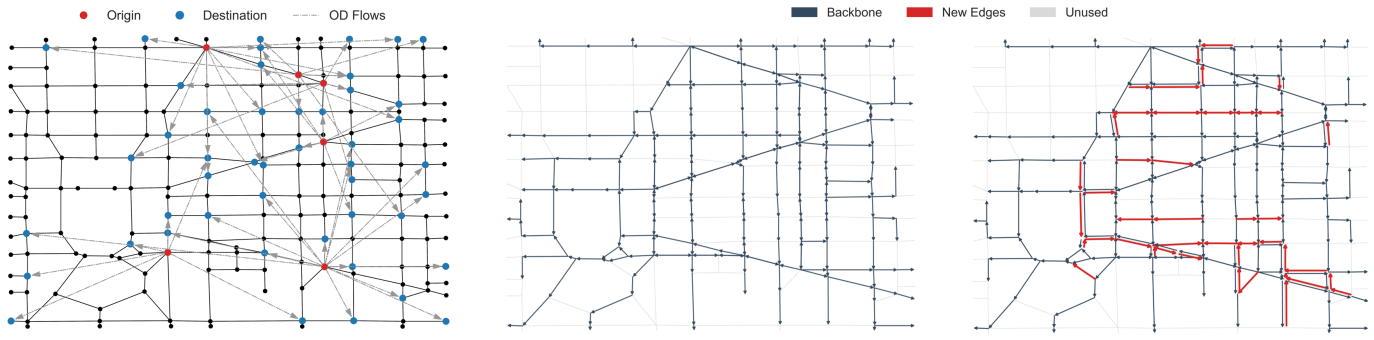
Figure 6. Paths of the six OD pairs with the largest reductions in travel time. The top of each subplot shows the time reduction percentage and the travel time of paths in the backbone network and the augmented design network.

that under the concentrated demand of WC, the backbone network may bear higher pressure from flow carrying. The augmented design network establishes multiple groups of lateral connections around the OD demand center (Figure 8(c)). In terms of performance under UE conditions, the TSTT of the augmented network decreases from 2.37×10^6s to 1.32×10^6s, a reduction of 44.52%. Meanwhile, the top-5 highest-flow edges in the backbone network exhibit consistent load reductions after design,



(a) (b) (c)

Figure 7. Structure and robustness under the RNP scenario. (a) Scatter plot of edge betweenness centrality and flow. (b) Edge betweenness distribution of the backbone network and the augmented design network. (c) Robustness curve.



(a) (b) (c)

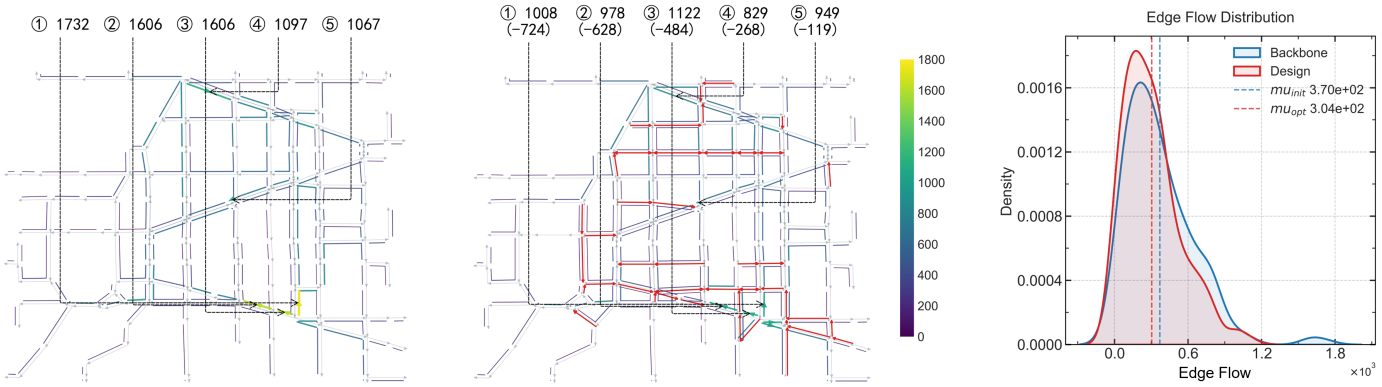
Figure 8. Design results under the WC scenario. (a) OD distribution in the WC scenario. Only 50 out of the 100 OD pairs are shown. (b) Backbone network. $|E^{init}| = 236$. (c) Augmented design network. Edges with a single arrow are one-way edges, while edges with arrows at both ends are bidirectional edges.

with the top-1 highest-flow edge decreasing by 41.80%, indicating that the augmentation effectively weakens the dependence on the main trunk corridors (Figure 9(a, b)). At the distribution level, the edge-flow distribution shows a more pronounced contraction in the high-flow tail, with the mean decreasing from 3.70×10^2 to 3.04×10^2, a reduction of 17.84% (Figure 9(c)).

From a structural perspective, the augmented design supplements the backbone network with two high-betweenness edges to alleviate their flow pressure, and additionally introduces several high-betweenness edges that carry low and moderate flows to enhance overall connectivity (Figure 10(a)). Although this design increases the average betweenness of the network, it reduces the extreme betweenness values. Robustness results show that the augmented design network improves the robustness of the backbone network, increasing it from 1.44 to 4.37, which exceeds the robustness threshold of 1.75 (Figure 10(c)). This indicates that, under the WC scenario, the proposed method can balance robustness requirements and performance under concentrated demand. The learned design strategy not only improves system operational performance through systematic flow redistribution, but also identifies high-betweenness edges that can enhance robustness while remaining non-critical in terms of traffic load under the given OD demand.

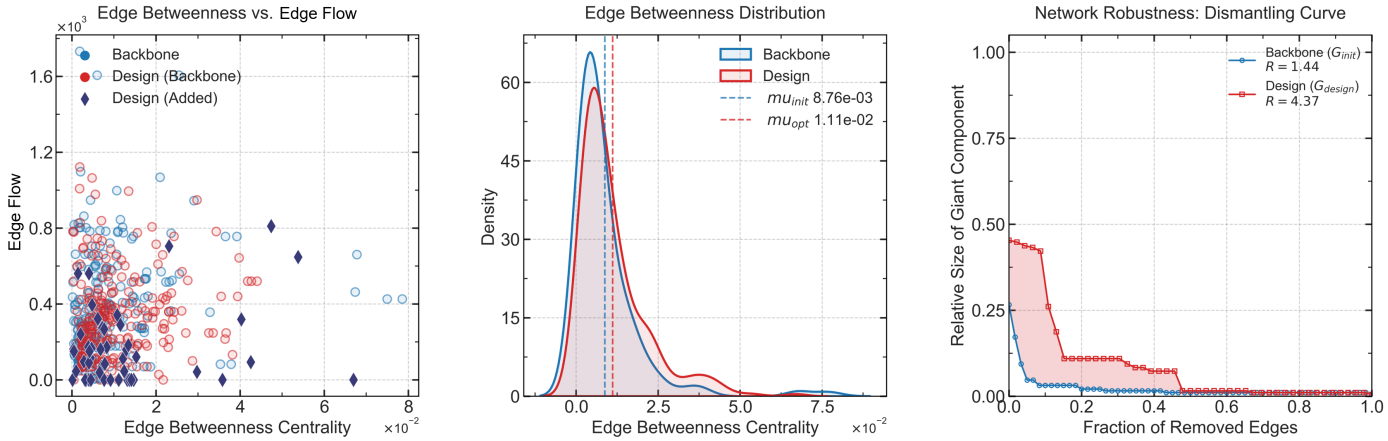
4.3.3 Design Results under the SU Scenario

The SU scenario represents an evenly distributed, wide-area demand pattern (Figure 11(a)). In our experiments, backbone networks under SU typically do not exhibit the extreme bottlenecks observed in the WC scenario, but may still contain localized critical concentrations. Under the same total demand, the number of edges in the backbone network under the SU scenario (374) is close to that under the RNP scenario (381) (Figure 11(b)), but the peak flow carried by edges is significantly lower than that in the RNP scenario (Figure 12(a)). This implies that under the evenly distributed demand of the SU



(a) (b) (c)

Figure 9. Flow distributions under the WC scenario. (a) Flow distribution of the backbone network. (b) Flow distribution of the augmented design network. (c) Comparison of flow distributions between the backbone network and the augmented design network. In (a) and (b), the flow values of the top-5 highest-flow edges in the backbone network and their decreases after augmentation are annotated. Red edges indicate new edges. Edges following the viridis colormap indicate the magnitude of traffic flow with layout following the right-hand traffic rule.



(a) (b) (c)

Figure 10. Structure and robustness analysis under the WC scenario. (a) Scatter plot of edge betweenness centrality and flow. (b) Edge-betweenness distributions of the backbone network and the augmented design network. (c) Robustness curve.

scenario, although the overall traffic load pressure is reduced, the spatially extensive OD demands still require a relatively dense backbone to maintain OD reachability across the entire region. The augmented design network also constructs multiple sets of paths across different regions of the network around the widely distributed OD demands (Figure 11(c)). For the performance under UE, the augmented network reduces TSTT from 1.09×10^6s to 1.06×10^6s, a 2.66% reduction, smaller than those observed in the RNP and WC scenarios. This is consistent with the relatively low pressure of traffic load in the SU scenario. However, for several high-load edges, the augmented design is still able to reduce their flows, suggesting reduced reliance on the original bottlenecks (Figure 12(b)). Specifically, the top-5 highest-flow edges in the backbone network exhibit systematic load reductions after design, with the most heavily loaded edge decreasing by 35.80%. At the distribution level, the edge-flow distribution also exhibits a noticeable contraction in the high-flow tail, with the mean decreasing from 2.49×10^2 to 2.19×10^2, corresponding to a 12.05% reduction (Figure 12(c)).

From a structural perspective, the augmented design under the SU scenario exhibits most added edges have low betweenness centrality and carry relatively low flows (Figure 13(a)). This reduces the average edge betweenness of the network (Figure 13(b)), with a reduction of 13.16%. Robustness results show that the augmented design network improves the robustness of the backbone network, increasing it from 12.74 to 17.23, which exceeds the robustness threshold of 14.44 (Figure 13(c)).

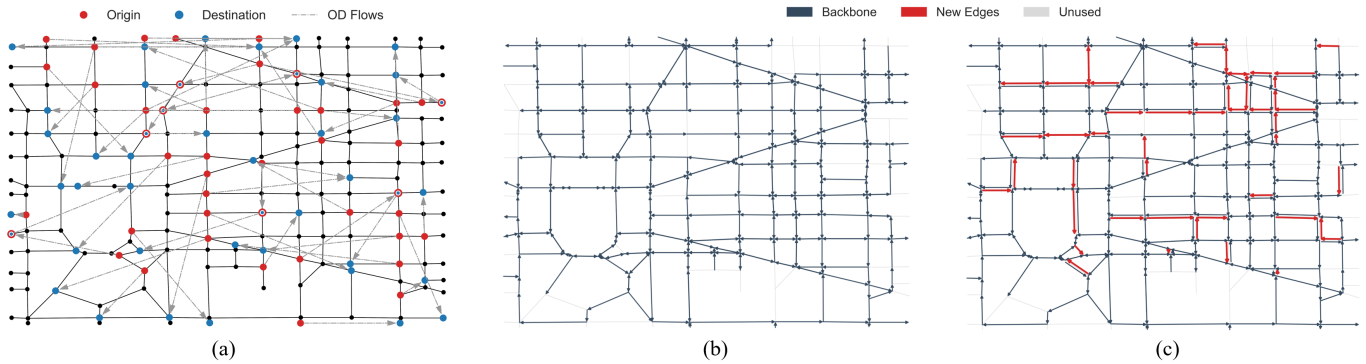


Figure 11. Design results under the SU scenario. (a) OD distribution in the SU scenario; for clarity, only 50 out of the 100 OD pairs are displayed. (b) Backbone network. $|E^{init}| = 374$. (c) Augmented design network. Edges with a single arrow are one-way edges, while edges with arrows at both ends are bidirectional edges.

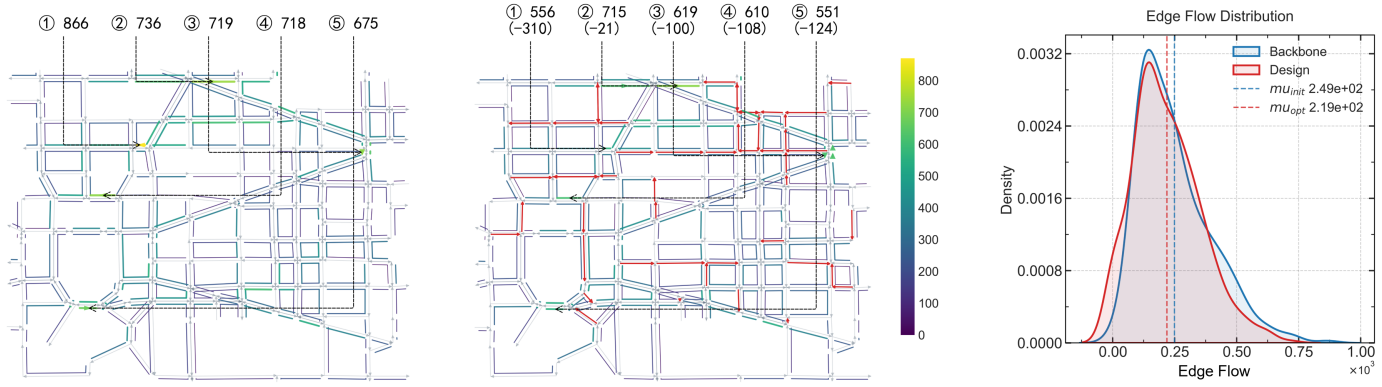


Figure 12. Flow distributions under the SU scenario. (a) Flow distribution of the backbone network. (b) Flow distribution of the augmented design network. (c) Comparison of flow distributions between the backbone and augmented design networks. In (a) and (b), the flow values of the top-5 highest-flow edges in the backbone network, along with their reductions after augmentation, are annotated. Red edges indicate new edges. Edges following the viridis colormap indicate the magnitude of traffic flow with layout following the right-hand traffic rule.

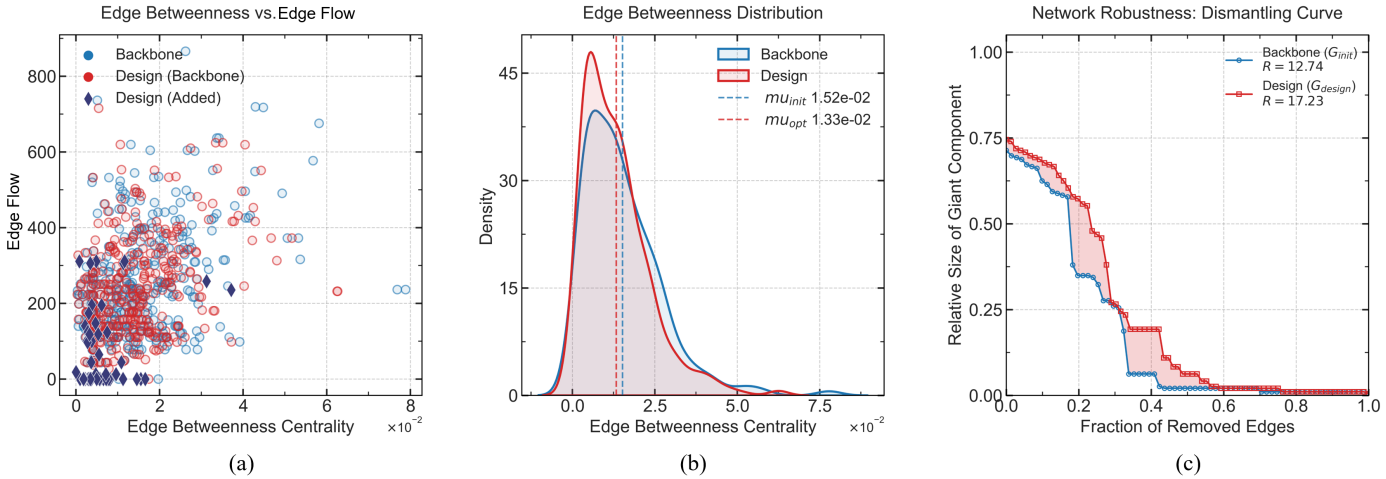


Figure 13. Structure and robustness analysis under the SU scenario. (a) Scatter plot of edge betweenness centrality and flow. (b) Edge-betweenness distributions of the backbone network and the augmented design network. (c) Robustness curve.

4.4 Performance Comparison

To further evaluate the effectiveness of the proposed method, we compare the P3O-based RL algorithm with SA algorithm. Under the same candidate topology and the same OD scenarios, the two methods adopt the same augmentation budget (the number

of added edges is B′) and an identical evaluation pipeline. For each output network, the TSTT under UE and the dismantling-based robustness metric are calculated. The objective and constraint evaluation for SA are consistent with those of the RL method, i.e., minimizing TSTT while satisfying the robustness threshold. When a candidate solution does not satisfy the threshold, the same infeasibility criterion is applied to ensure fair comparison. We conduct batch repeated experiments for the three demand scenarios (RNP, WC, and SU). Each test corresponds to an independently generated OD instance, yielding 30 network design outcomes per scenario.

4.5 Solution Quality Comparison

In terms of network performance (Figure 14(a-c)), the RL method achieves lower average TSTT than SA in both the WC and RNP scenarios. The advantage is most significant in the WC scenario, whereas in the SU scenario the performance of RL and SA is comparable. This suggests that the advantage of the RL-based design tends to be larger when the initial backbone exhibits higher load concentration. Regarding robustness (Figure 14(d-f)), RL also outperforms SA on average across all scenarios. The advantage is especially evident in the WC scenario, suggesting that under high-load and adversarial demand conditions, RL tends to learn augmentation strategies that promote decentralization of critical links and redundancy enhancement. In contrast, SA tends to produce designs with more concentrated critical edges in these instances, which is associated with lower robustness. For the SU scenario, the overall advantage of RL is smaller, with a lower minimum value and larger variability. This suggests that when demand is relatively stable and inherent network bottlenecks are weak, robustness improvements may exhibit significantly diminishing marginal returns while still satisfying the constraint. Regarding computational runtime (Figure 14(g-i)), the RL method demonstrates a clear advantage across all three scenarios.

4.6 Efficiency Comparison

To further evaluate algorithmic efficiency, we present TSTT-Runtime scatter plots for the three scenarios, with results from both methods annotated for each instance to enable direct instance-wise comparison (Figure 15). As indicated by the horizontal distribution of points, the RL solutions are generally shifted toward shorter runtimes, demonstrating a clear advantage in per-instance computational cost. This efficiency gain aligns with the fundamental differences in search mechanisms: RL leverages a learned policy to directly generate sequences of augmentation decisions, avoiding extensive neighborhood perturbations and iterative temperature annealing. In contrast, SA relies on repeated candidate generation and evaluation, with computational cost scaling approximately with the number of search iterations, making it increasingly time-consuming for more difficult instances or under tighter feasibility constraints.

In instance-level comparisons under identical conditions, most paired points in the RNP and WC scenarios exhibit a clear shift from SA (longer runtime and higher TSTT) to RL (shorter runtime and lower TSTT). This indicates that RL achieves simultaneous improvements in both solution quality and efficiency for the majority of instances. In the SU scenario, although the advantage of RL in terms of TSTT is less pronounced, its runtime remains substantially lower than that of SA. Overall, the distribution of results suggests that RL approaches a design frontier characterized by both low runtime and low TSTT, indicating better scalability for practical deployment. Under dynamic deployment requirements for low-altitude air route networks—such as multi-scenario, batch, and rolling updates—where OD demands change frequently or feasible alternative networks need to be generated rapidly, the inference-based design of RL can reduce online computational burden while maintaining improved solution quality in most instances.

5 Discussion and Conclusion

With the extensive deployment of UAV applications in urban environments, low-altitude operations increasingly exhibit a combination of high traffic density, stringent operational constraints, and substantial uncertainty. Under such conditions, conventional point-to-point and free-flight paradigms face limited scalability in conflict resolution, congestion mitigation, and emergency response. Structured air route networks therefore provide essential infrastructure for scalable low-altitude traffic operation. Their designs need to not only deliver demand-driven system performance, but also sustain acceptable connectivity and service levels in the presence of failures and adversarial disruptions. To address the coupled problem of performance and robustness, this paper proposes a low-altitude air route network augmentation design framework under robustness boundary constraints. Given an initial candidate topology and a multi-OD demand,

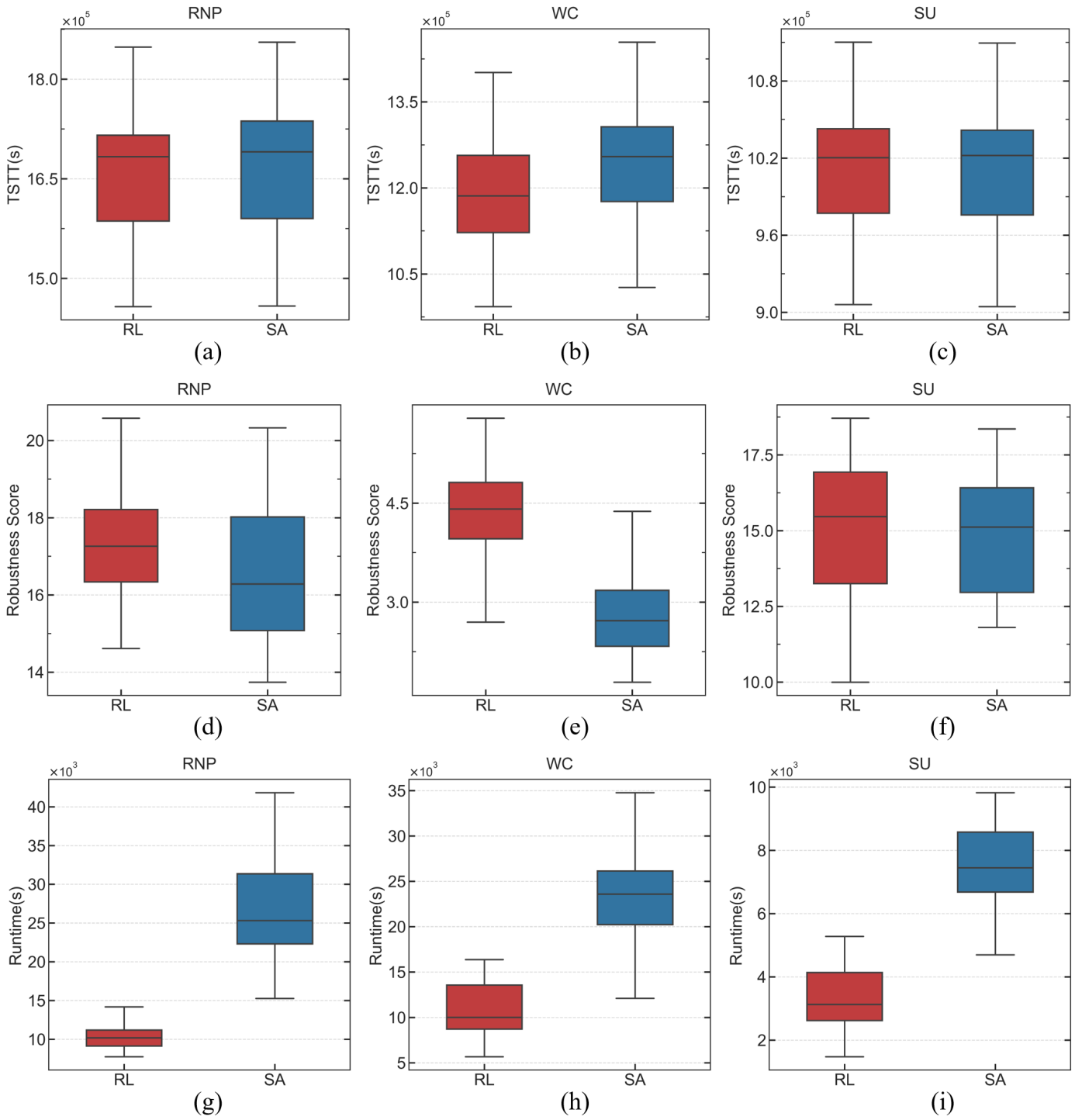


Figure 14. Batch comparison of design results between RL and SA. (a)-(c) Box plots of TSTT. (d)-(f) Box plots of robustness. (g)-(i) Box plots of runtime. The results shown correspond to 30 instances.

the framework first constructs an initial backbone network using the set of OD shortest paths. Then, the network design process of selecting a limited number of additional edges from the candidate edge set is modeled as a constrained Markov decision process. An augmentation strategy is learned under a safe RL framework, ensuring that the network robustness does not fall below a predefined threshold while minimizing TSTT under UE conditions. The framework utilizes graph neural networks to encode the structural information and state features of the current design network. Additionally, global signals related to the robustness constraint are provided as conditions to enhance the adaptability and generalization of the policy across different topologies and constraint levels.

The experimental results across the three OD scenarios

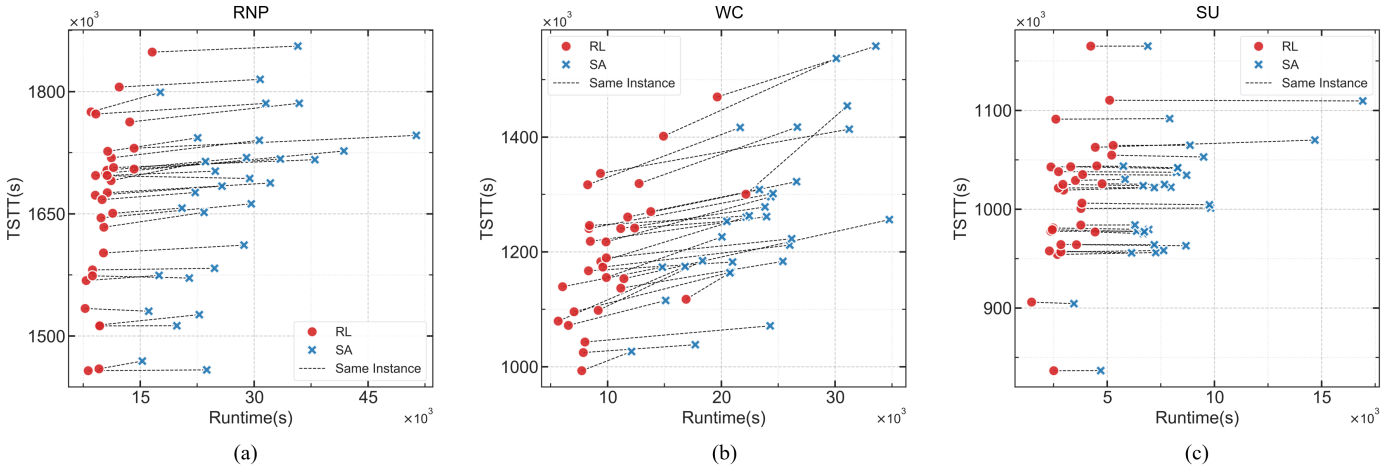


Figure 15. Efficiency and performance comparison between RL and SA. (a) TSTT-Runtime scatter plot for the RNP scenario (with guiding lines connecting RL and SA results). (b) WC scenario. (c) SU scenario. The dark gray dashed lines connect the results of the two algorithms for the same instance.

show that under the same augmentation budget and evaluation process, the proposed method consistently reduces TSTT under UE while meeting the robustness threshold. Taking the RNP scenario as an example, the initial backbone network exhibits significant traffic concentration and corridor load under UE distribution. In contrast, the augmented network introduces bypasses and alternative paths between critical corridors by adding new edges, thereby expanding the set of available paths for OD pairs, and reducing congestion pressure on high-load edges. This increases path diversity and shifts flow away from a few heavily loaded corridors, alleviating congestion hotspots. For several OD pairs, the augmented network achieves substantial reductions in travel time. Notably, for some OD pairs, the path lengths increase while travel times decrease, owing to the avoidance of highly congested edges. This suggests that the performance improvement primarily results from congestion avoidance and traffic redistribution, rather than simple geometric shortening. Structural analysis is consistent with this observation: after augmentation, the betweenness centrality distribution of the network shifts leftward, with the mean value decreasing. By weakening the coupling between high-betweenness and high-flow edges and adding redundant bypasses, the design reduces structural criticality and improves robustness under targeted dismantling. This is consistent with the robustness degradation curve, which shows that the augmented network maintains larger connected components during the intermediate stages of dismantling, with connectivity degradation occurring more gradually. As a result, the augmented network satisfies the robustness threshold constraint while achieving higher performance.

Through comparison with the classical heuristic simulated annealing approach, we illustrate the advantage of the proposed RL approach on both performance and runtime, supporting its practical viability. Overall, the proposed RL approach achieves lower TSTT values and stronger robustness across different scenarios. The proposed approach gains the highest advantage in the WC scenario, followed by the RNP scenario, while performing comparably in the SU scenario. This indicates that under highly concentrated demand structures, strategies with constraint-learning capability may be better suited to producing augmentation designs with dispersed critical links and enhanced redundancy. More importantly, the design runtime of the RL method is substantially lower than that of SA. In instance-wise comparisons across multiple scenarios, RL more frequently demonstrates a trend of faster runtimes and superior network performance. This property aligns well with the operational requirements of low-altitude air route networks, which demand multi-scenario, batch, and rolling updates.

This study supports the feasibility of combining graph-based representations with safe RL for robustness-constrained network design, but several limitations remain. First, the robustness metric adopted in this paper is an area-based indicator derived from a dismantling process with a specific attack strategy. Although this formulation can reflect connectivity degradation when critical bottlenecks are continuously destroyed, its ability to represent other failure modes, such as random failures, regional disruptions, link breakages, or multi-strategy adversarial scenarios, remains limited [60, 61].

Second, the evaluation of traffic performance in this study adopts the UE model. Although it is classical and interpretable, it cannot fully capture real-world operational complexities, such as time-varying capacities, priority rules, cooperative conflict avoidance, and strategic interactions among multiple operators [62]. Third, multiple engineering-level constraints such as length-based costs, noise and risk constraints, and geometric constraints of no-fly zones have not yet been comprehensively incorporated into a unified framework [63, 64]. This limitation may reduce the direct applicability of the proposed approach in more complex and realistic operational settings.

Based on the above limitations, future research can be carried out in three directions. First, the robustness constraint can be extended to multiple failure modes and multi-strategy adversarial settings, forming robustness boundaries under multi-dimensional robustness evaluation [65]. Second, time-varying OD, stochastic disturbances, and online transfer mechanisms can be introduced, to strengthen the adaptability of the policy to demand fluctuations [66, 67]. Third, while maintaining interpretability and computational tractability, engineering indicators such as cost, risk, and fairness together with the robustness constraint can be embedded into the learning framework [68]. And the implementation of additional baseline methods can be considered, such as large-neighborhood or evolutionary search, as well as alternative safe RL algorithms [69, 70], to further clarify the advantage boundaries and applicable conditions of the method. Overall, this study suggests that, for complex network design problems where congestion-equilibrium mechanisms and robustness boundary constraints coexist, safe RL can provide a promising engineering approach for obtaining structural solutions that balance performance and robustness.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported by the Civil Aviation Safety Capacity Building Project of Civil Aviation Administration of China under Grant HA202511, and by the National Natural Science Foundation of China under Grant 72501083.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Hamissi, A., & Dhraief, A. (2023). A survey on the unmanned aircraft system traffic management. *ACM Computing Surveys, 56*(3), 1–37. [CrossRef]

[2] Bauranov, A., & Rakas, J. (2021). Designing airspace for urban air mobility: A review of concepts and approaches. *Progress in Aerospace Sciences, 125*, 100726. [CrossRef]

[3] Xu, C., Liao, X., Ye, H., & Yue, H. (2020). Iterative construction of low-altitude UAV air route network in urban areas: Case planning and assessment. *Journal of Geographical Sciences, 30*(9), 1534–1552. [CrossRef]

[4] Lee, U. J., Ahn, S. J., Choi, D. Y., Chin, S. M., & Jang, D. S. (2023). Airspace designs and operations for uas traffic management at low altitude. *Aerospace, 10*(9), 737. [CrossRef]

[5] Zhang, Z., Zheng, Y., Li, C., Jiang, B., & Li, Y. (2025). Designing an Urban Air Mobility Corridor Network: A Multi-Objective Optimization Approach Using U-NSGA-III. *Aerospace, 12*(3), 229. [CrossRef]

[6] Zhai, W., Han, B., Li, D., Duan, J., & Cheng, C. (2021). A low-altitude public air route network for UAV management constructed by global subdivision grids. *PLoS One, 16*(4), e0249680. [CrossRef]

[7] Wang, Z., Delahaye, D., Farges, J. L., & Alam, S. (2022, June). Route network design in low-altitude airspace for future urban air mobility operations: A case study of urban airspace of Singapore. In *International Conference on Research in Air Transportation (ICRAT 2020)*.

[8] Lozano Tafur, C., Orduy Rodríguez, J., Aldana Rodríguez, D., Traslaviña, D. S., Fernández Valencia, S., & Celis Ardila, F. H. (2025). Risk-Based Design of Urban UAS Corridors. *Drones, 9*(12), 815. [CrossRef]

[9] Zhang, H., Tian, T., Feng, O., Wu, S., & Zhong, G. (2023). Research on public air route network planning of urban low-altitude logistics unmanned aerial vehicles. *Sustainability, 15*(15), 12021. [CrossRef]

[10] Stuive, L., & Gzara, F. (2024). Airspace network design for urban UAV traffic management with congestion. *Transportation Research Part C: Emerging Technologies, 169*, 104882. [CrossRef]

[11] Li, Z., Li, S., Lu, J., & Wang, S. (2025). Air Route Network Planning Method of Urban Low-Altitude Logistics UAV with Double-Layer Structure. *Drones, 9*(3), 193. [CrossRef]

[12] Patriksson, M. (2015). *The traffic assignment problem: models and methods*. Courier Dover Publications.

[13] Albert, R., Jeong, H., & Barabási, A. L. (2000). Error and attack tolerance of complex networks. *nature, 406*(6794), 378-382. [CrossRef]

[14] Wandelt, S., Sun, X., Feng, D., Zanin, M., & Havlin, S. (2018). A comparative analysis of approaches to network-dismantling. *Scientific Reports, 8*(1), 13513. [CrossRef]

[15] Darvariu, V. A., Hailes, S., & Musolesi, M. (2024). Graph reinforcement learning for combinatorial optimization: A survey and unifying perspective. *arXiv preprint arXiv:2404.06492*.

[16] Cappart, Q., Chételat, D., Khalil, E. B., Lodi, A., Morris, C., & Veličković, P. (2023). Combinatorial optimization and reasoning with graph neural networks. *Journal of Machine Learning Research, 24*(130), 1–61.

[17] Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research, 134*, 105400. [CrossRef]

[18] Zhao, W., He, T., Chen, R., Wei, T., & Liu, C. (2023). State-wise safe reinforcement learning: A survey. *arXiv preprint arXiv:2302.03122*.

[19] Liu, Y., Halev, A., & Liu, X. (2021, August). Policy learning with constraints in model-free reinforcement learning: A survey. In *The 30th international joint conference on artificial intelligence (ijcai)*. [CrossRef]

[20] Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017, July). Constrained policy optimization. In *International conference on machine learning* (pp. 22-31). Pmlr.

[21] Stooke, A., Achiam, J., & Abbeel, P. (2020, November). Responsive safety in reinforcement learning by pid lagrangian methods. In *International conference on machine learning* (pp. 9133-9143). PMLR.

[22] Li, J., Fridovich-Keil, D., Sojoudi, S., & Tomlin, C. J. (2021, December). Augmented lagrangian method for instantaneously constrained reinforcement learning problems. In *2021 60th IEEE Conference on Decision and Control (CDC)* (pp. 2982-2989). IEEE. [CrossRef]

[23] Wen, L., Duan, J., Li, S. E., Xu, S., & Peng, H. (2020, September). Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-7). IEEE. [CrossRef]

[24] Wang, W., Yu, N., Gao, Y., & Shi, J. (2019). Safe off-policy deep reinforcement learning algorithm for volt-var control in power distribution systems. *IEEE Transactions on Smart Grid, 11*(4), 3008–3018. [CrossRef]

[25] Tessler, C., Mankowitz, D. J., & Mannor, S. (2018). Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*.

[26] Xu, C., Ye, H., Yue, H., Tan, X., & Liao, X. (2020). Iterative construction of UAV low-altitude air route network in an urbanized region: theoretical system and technical roadmap. *Acta Geogr. Sin, 75*, 917–930.

[27] Gao, C. F., Hu, Z. H., & Wang, Y. Z. (2022). Optimizing the hub-and-spoke network with drone-based traveling salesman problem. *Drones, 7*(1), 6. [CrossRef]

[28] Leurent, F., & Boujnah, H. (2014). A user equilibrium, traffic assignment model of network route and parking lot choice, with search circuits and cruising flows. *Transportation Research Part C: Emerging Technologies, 47*, 28–46. [CrossRef]

[29] Morandi, V. (2024). Bridging the user equilibrium and the system optimum in static traffic assignment: a review. *4OR, 22*(1), 89–119. [CrossRef]

[30] Wardrop, J. G. (1952). Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers, 1*(3), 325–362. [CrossRef]

[31] Salazar, M., Tsao, M., Aguiar, I., Schiffer, M., & Pavone, M. (2019, June). A congestion-aware routing scheme for autonomous mobility-on-demand systems. In *2019 18th European Control Conference (ECC)* (pp. 3040-3046). IEEE. [CrossRef]

[32] Aftabuzzaman, M. (2007, September). Measuring traffic congestion-a critical review. In *30th Australasian transport research forum* (Vol. 1). ETM GROUP London UK.

[33] Daskin, M. S. (1985). Urban transportation networks: Equilibrium analysis with mathematical programming methods.

[34] Schneider, C. M., Moreira, A. A., Andrade Jr, J. S., Havlin, S., & Herrmann, H. J. (2011). Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences, 108*(10), 3838-3841. [CrossRef]

[35] Holme, P., Kim, B. J., Yoon, C. N., & Han, S. K. (2002). Attack vulnerability of complex networks. *Physical review E, 65*(5), 056109. [CrossRef]

[36] Garcıa, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research, 16*(1), 1437-1480.

[37] Ray, A., Achiam, J., & Amodei, D. (2019). Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708, 7*(1), 2.

[38] Altman, E. (2021). *Constrained Markov decision processes*. Routledge. [CrossRef]

[39] Ding, D., Zhang, K., Basar, T., & Jovanovic, M. R. (2020, December). Natural policy gradient primal-dual method for constrained Markov decision processes. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (pp. 8378-8390).

[40] Chow, Y., Nachum, O., Duenez-Guzman, E., & Ghavamzadeh, M. (2018, December). A lyapunov-based approach to safe reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 8103-8112).

[41] Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017, July). Constrained policy optimization. In *International conference on machine learning* (pp. 22-31). Pmlr.

[42] Xuan, C., Zhang, F., Yin, F., & Lam, H. K. (2023). Constrained proximal policy optimization. *arXiv preprint arXiv:2305.14216*.

[43] Dai, J., Ji, J., Yang, L., Zheng, Q., & Pan, G. (2023). Augmented proximal policy optimization for safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 7288–7295). [CrossRef]

[44] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In *International conference on machine learning* (pp. 1889-1897). PMLR.

[45] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[46] Yu, M., Yang, Z., Kolar, M., & Wang, Z. (2019, December). Convergent policy optimization for safe reinforcement learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (pp. 3127-3139).

[47] Zhang, L., Shen, L., Yang, L., Chen, S., Yuan, B., Wang, X., & Tao, D. (2022). Penalized proximal policy optimization for safe reinforcement learning. *arXiv preprint arXiv:2205.11814*.

[48] Cai, T., Luo, S., Xu, K., He, D., Liu, T. Y., & Wang, L. (2021, July). Graphnorm: A principled approach to accelerating graph neural network training. In *International Conference on Machine Learning* (pp. 1204-1215). PMLR.

[49] Brody, S., Alon, U., & Yahav, E. (2021). How attentive are graph attention networks?. *arXiv preprint arXiv:2105.14491*.

[50] Nair, V., & Hinton, G. E. (2010, June). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning* (pp. 807-814).

[51] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (Vol. 30, No. 1, p. 3).

[52] Janner, M., Li, Q., & Levine, S. (2021). Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems, 34*, 1273-1286.

[53] Bomze, I. M., Mertikopoulos, P., Schachinger, W., & Staudigl, M. (2019). Hessian barrier algorithms for linearly constrained optimization problems. *SIAM Journal on Optimization, 29*(3), 2100-2127. [CrossRef]

[54] Alumur, S., & Kara, B. Y. (2008). Network hub location problems: The state of the art. *European journal of operational research, 190*(1), 1–21. [CrossRef]

[55] Anda, C., Erath, A., & Fourie, P. J. (2017). Transport modelling in the age of big data. *International Journal of Urban Sciences, 21*(sup1), 19-42. [CrossRef]

[56] Immers, L., & Stada, J. (1998). *Traffic demand modelling*. Katholieke Universiteit Leuven.

[57] Fávero, L. P., Belfiore, P., & de Freitas Souza, R. (2023). *Data science, analytics and machine learning with R*. Academic Press. [CrossRef]

[58] Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. *Statistical science, 8*(1), 10–15. [CrossRef]

[59] Buesser, P., Daolio, F., & Tomassini, M. (2011, April). Optimizing the robustness of scale-free networks with simulated annealing. In *International conference on adaptive and natural computing algorithms* (pp. 167-176). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[60] Wandelt, S., Lin, W., Sun, X., & Zanin, M. (2022). From random failures to targeted attacks in network dismantling. *Reliability Engineering & System Safety, 218*, 108146. [CrossRef]

[61] Shekhtman, L. M., Danziger, M. M., Vaknin, D., & Havlin, S. (2018). Robustness of spatial networks and networks of networks. *Comptes Rendus Physique, 19*(4), 233–243. [CrossRef]

[62] Wang, Z., Delahaye, D., Farges, J. L., & Alam, S. (2022). Complexity optimal air traffic assignment in multi-layer transport network for Urban Air Mobility operations. *Transportation Research Part C: Emerging Technologies, 142*, 103776. [CrossRef]

[63] Shi, Z., & Ng, W. K. (2018, June). A collision-free path planning algorithm for unmanned aerial vehicle delivery. In *2018 international conference on unmanned aircraft systems (icuas)* (pp. 358-362). IEEE. [CrossRef]

[64] Sun, L., Deng, H., Wei, P., & Xie, W. (2025). On a fair and risk-averse urban air mobility resource allocation problem under demand and capacity uncertainties. *Naval Research Logistics (NRL), 72*(1), 111–132. [CrossRef]

[65] Dong, S., Wang, H., Mostafavi, A., & Gao, J. (2019). Robust component: a robustness measure that incorporates access to critical facilities under disruptions. *Journal of The Royal Society Interface, 16*(157). [CrossRef]

[66] Fu, H., Akamatsu, T., Satsukawa, K., & Wada, K. (2022). Dynamic traffic assignment in a corridor network: Optimum versus equilibrium. *Transportation Research Part B: Methodological, 161*, 218–246. [CrossRef]

[67] Batista, S. F. A., Leclercq, L., & Menéndez, M. (2021). Dynamic Traffic Assignment for regional networks

with traffic-dependent trip lengths and regional paths. *Transportation Research Part C: Emerging Technologies, 127*, 103076. [CrossRef]

[68] Gao, Z., Yu, Y., Wei, Q., Topcu, U., & Clarke, J. P. (2024). Noise-aware and equitable urban air traffic management: An optimization approach. *Transportation Research Part C: Emerging Technologies, 165*, 104740. [CrossRef]

[69] Liang, M., Xu, M., & Wang, S. (2025). A novel multi-objective evolutionary algorithm for transit network design and frequency-setting problem considering passengers' choice behaviors under station congestion. *Transportation Research Part B: Methodological, 197*, 103238. [CrossRef]

[70] Wachi, A., Shen, X., & Sui, Y. (2024). A survey of constraint formulations in safe reinforcement learning. *arXiv preprint arXiv:2402.02025*.

[71] Stuive, L., & Gzara, F. (2024). Airspace network design for urban UAV traffic management with congestion. *Transportation Research Part C: Emerging Technologies, 169*, 104882. [CrossRef]

[72] Gao, C. F., Hu, Z. H., & Wang, Y. Z. (2022). Optimizing the hub-and-spoke network with drone-based traveling salesman problem. *Drones, 7*(1), 6. [CrossRef]

[73] Wu, W., Wang, Z., Lin, L., Chang, X., & Tian, L. (2025). An efficient coverage path planning method for UAV in complex concave regions. *Scientific Reports, 15*(1), 37227. [CrossRef]

[74] Salazar, M., Tsao, M., Aguiar, I., Schiffer, M., & Pavone, M. (2019, June). A congestion-aware routing scheme for autonomous mobility-on-demand systems. In *2019 18th European Control Conference (ECC)* (pp. 3040-3046). IEEE. [CrossRef]

[75] Gore, N., Arkatkar, S., Joshi, G., & Antoniou, C. (2023). Modified bureau of public roads link function. *Transportation Research Record, 2677*(7), 966–990. [CrossRef]

[76] Gentile, G. (2016). Solving a dynamic user equilibrium model based on splitting rates with gradient projection algorithms. *Transportation Research Part B: Methodological, 92*, 120-147. [CrossRef]

[77] Yang, H., & Bell, M. G. H. (1998). Models and algorithms for road network design: a review and some new developments. *Transport Reviews, 18*(3), 257–278. [CrossRef]

[78] Morandi, V. (2024). Bridging the user equilibrium and the system optimum in static traffic assignment: a review. *4OR, 22*(1), 89–119. [CrossRef]

[79] Marechal, M., & de Grange, L. (2024). Generalization of Beckmann's transformation for traffic assignment models with asymmetric cost functions. *Journal of Advanced Transportation, 2024*(1), 2921485. [CrossRef]

[80] AequilibraE. (2026). Traffic assignment procedure. Retrieved January 22, 2026, from https://www.aequilibrae.com/develop/python/traffic_assignment/assignment_procedures.html

[81] Patro, S. G. O. P. A. L., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

[82] Liu, M., Zhu, M., & Zhang, W. (2022). Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*.

[83] Zhang, L., Shen, L., Yang, L., Chen, S., Yuan, B., Wang, X., & Tao, D. (2022). Penalized proximal policy optimization for safe reinforcement learning. *arXiv preprint arXiv:2205.11814*.

[84] Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Appendix

A Supplementary Note 1: Patterns of OD demand on the Air Route Network

1. **Random-Node-Pair (RNP) Pattern**

   In the RNP pattern, origins o and destinations d are randomly sampled from the node set V to generate a set of OD pairs. This pattern is suitable for modeling point-to-point on-demand services, such as consumer-to-consumer (C2C) intra-city express delivery, as well as emergency supply transportation scenarios without fixed hubs. Under this demand pattern, the primary requirements of the network are global connectivity and average path efficiency [71].

2. **Warehouse-Customer (WC) Pattern**

   The WC pattern assumes the presence of a fixed warehouse hub node $v_h \in V$ in the urban area. For each OD pair, the origin is fixed at the warehouse node, and the destination is randomly sampled from the node set to represent potential customer request locations. This pattern reflects mainstream business-to-consumer (B2C) e-commerce of last-mile delivery and typical logistics structures of hub-and-spoke. Under the WC pattern, demands form high-density radial flows from the hub, which are prone to creating bottlenecks and congestion accumulation. This pattern serves as a critical stress-testing scenario for the evaluation of capacity allocation and load balance [72].

3. **Space-Uniform (SU) Pattern**

   In the SU pattern, two sets of spatial coordinates

are first randomly generated within the bounding box of the study area, and are then mapped to their nearest network nodes as origins and destinations. This procedure yields a spatially uniform sampling over the geographic region, allowing the representation of relatively homogeneous demand scenarios while avoiding excessive concentration of demand in areas with dense network nodes. The SU pattern is representative of coverage-oriented tasks such as urban inspection and security surveillance. It emphasizes balanced spatial service and the avoidance of coverage blind spots, consistent with the task-level objectives of Coverage Path Planning (CPP) [73].

B Supplementary Note 2: UE Traffic Assignment under the BPR Function

1. Bureau of Public Roads (BPR) Function

The BPR function has been widely adopted in traffic assignment and network design studies. Its parametric formulation maps edge traffic flow to travel time and exhibits desirable properties such as monotonicity, differentiability, and convexity, which facilitate its integration into optimization models and algorithm training frameworks. Accordingly, this study employs the BPR function to characterize the nonlinear relationship between edge travel time and traffic flow [74]. For any edge $e \in E$ in a given network $G = (V, E)$, t_e^0 denotes its free-flow travel time, ρ_e denotes its capacity, and f_e is the traffic flow on the edge. The congestion travel time based on the BPR function is defined as [75]:

$$t_e(f_e) = t_e^0 \left(1 + \alpha_{\text{BPR}} \left(\frac{f_e}{\rho_e}\right)^{\beta_{\text{BPR}}}\right), \tag{A.1}$$

where $\alpha_{\text{BPR}} > 0$ and $\beta_{\text{BPR}} > 1$ are congestion parameters.

2. UE Traffic Assignment

Given an OD demand matrix $Q = \{q_{od}\}_{(o,d)\in W}$, the demand of each OD pair (o, d) is denoted by q_{od}. Let $P_{od} = \{p\}$ represent the set of feasible paths for (o, d), and x_p denote the flow on path p. The path flows should satisfy the flow conservation constraints [76, 77]:

$$\sum_{p\in P_{od}} x_p = q_{od}, \quad x_p \geq 0. \tag{A.2}$$

The edge flow $f_e(x)$ can be obtained by aggregating the corresponding path flows [77]:

$$f_e(x) = \sum_{(o,d)\in W} \sum_{p\in P_{od}} \delta_{ep} x_p, \tag{A.3}$$

where $x = \{x_p \mid (o, d) \in W, p \in P_{od}\}$ is the path flow vector, and $\delta_{ep} = 1$ indicates that edge e is on path p, and $\delta_{ep} = 0$ otherwise.

Under the UE condition, for any OD pair (o, d), all utilized paths exhibit equal travel times, which are no greater than the travel times of any unused paths [78]. When the BPR function is adopted, the UE condition can be equivalently formulated as the following optimization problem [77, 79]:

$$\min_x \sum_{e\in E} \int_0^{f_e(x)} t_e(w)\, dw, \quad \text{s.t.} \sum_{p\in P_{od}} x_p = q_{od}, \quad x_p \geq 0, \quad \forall (o, d) \in W. \tag{A.4}$$

This study adopts the Traffic Assignment module of the open-source transportation modeling tool, AequilibraE to solve the traffic assignment of UE on a given network [80], and calculates the total system travel time as the performance evaluation of the network.

C Supplementary Note 3: Input State of Graph and Features

The graph-structured input is centered on the currently designed network, and explicitly distinguishes between the selected edge set used for message passing and the candidate edge set used for decision making. Node features are constructed by concatenating static topological features with dynamic features induced by OD demands. Static features include the min–max normalized [81] node degree, in-degree, and out-degree. Dynamic OD-related features include the proportion of demand for which the node serves as an origin and as a destination, denoted by O_{norm}, D_{norm}, and the proportion of demand whose shortest paths pass through the node, denoted by P_{norm}, with their logarithmic transformations $\log(1 + \cdot)$. P_{norm} is obtained by counting, for each OD pair, the amount of demand passing through intermediate nodes along the shortest paths (weighted by free-flow travel time). The edge features of each directed edge (u, v) include the min–max normalized free-flow travel time t_{uv}^0, capacity ρ_{uv}, current flow f_{uv}, and saturation $sat_{uv} = f/\rho$. For edges that have been added to the

air route network, the flow corresponds to the flow value after UE assignment on the current network, whereas for candidate edges that have not been added, the flow value is set to zero.

It is worth noting that the forward propagation of the graph neural network is performed based on the current topology consisting of the already selected edges, while the features of the candidate edges that have not been added are incorporated as part of the subsequent action representation. Moreover, to obtain a stable graph-level representation, a virtual node v^* is added to the original graph, whose initial feature is set to zero, and all nodes are connected unidirectionally to this virtual node, such that the embedding of the virtual node can be regarded as a global aggregation vector. In addition, since the robustness threshold used in the experiments is related to the robustness of the initial backbone network, different networks may have different thresholds. Therefore, to make the policy sensitive to the robustness constraint of the current network, the environment provides a constructed robustness increment constraint value $d = -(R_{\text{thr}} - R(G_0))$ at the beginning of each episode as a global conditional input to the neural network, serving as a conditioning signal to guide model training [82].

D Supplementary Note 4: Details of Model Training

1. Trajectory Sampling

Trajectory sampling constitutes a fundamental step in RL, which is used to generate information such as states, actions, and rewards experienced by the agent when interacting with the environment. In the P3O-based augmentation design, trajectory sampling is performed through interactions with the environment to collect a complete trajectory, including the state at each time step, the selected action, and the resulting reward and cost. These data are used in subsequent policy updates to calculate advantage estimates of actions and ultimately update the network design policy. Each trajectory τ consists of multiple time steps, where each time step t includes the state s_t, action a_t, reward r_t and cost c_t of the CMDP process. In P3O, trajectory sampling is carried out through iterative interactions between the agent and the environment. For the k-th outer iteration, the procedure is summarized as follows:

1. **State initialization**: Starting from the initial backbone network G^{init}, the environment generates the initial state s_0 based on the current network configuration;
2. **Action execution**: Given the current state s_t, the agent selects an augmentation action a_t according to the current policy, i.e., chooses a new edge e_{new} from the candidate edge set $E^0 \setminus E_t$;
3. **Environment feedback**: Based on the selected edge, the environment updates the designed network, calculates the new traffic flow distribution, TSTT, and robustness value, and returns the next state s_{t+1}, reward r_t, and cost c_t;
4. **State update**: The agent proceeds to select the next augmentation edge based on the updated state s_{t+1}, continuing the interaction with the environment;
5. **Termination condition**: When the predefined augmentation budget B is reached, i.e., when B augmentation edges have been selected, the current trajectory sampling terminates. A trajectory τ sampled by the current policy π_{θ_k} is obtained and added to the trajectory set $D_k = \{\tau\}$. Once the number of trajectories collected in the trajectory set reaches the specified buffer capacity, sampling is suspended and the training phase begins, during which policy advantage estimation and parameter updates are performed sequentially.

2. Advantage Estimation

Advantage estimation is used to evaluate the superiority of a specific action relative to other actions. The advantage is calculated by comparing the return of an action under the current policy with the average return at that state. Advantage estimation helps guide the agent during policy updates to prioritize actions that can maximize performance.

The objective of advantage estimation is to compute the advantage function A_t for each state-action pair (s_t, a_t), which measures the performance improvement of action a_t compared with the average behavior at that state. It is defined as:

$$A_t = Q(s_t, a_t) - V(s_t), \tag{A.5}$$

where $Q(s_t, a_t)$ is the expected return after executing action a_t at state s_t, and $V(s_t)$ is the value function of state s_t, representing the expected long-term return when following the current policy from that state. In P3O, advantage estimation is used to compute the expected improvements in reward and cost under

the current policy, so as to enable effective gradient updates of the policy. P3O inherits the generalized advantage estimation (GAE) method from PPO, which introduces a parameter λ to control the smoothness of the advantage estimation, considering both immediate returns and balancing the uncertainty of future returns. GAE combines the advantages of temporal-difference (TD) methods and Monte Carlo (MC) methods, reducing variance while maintaining relatively low bias. The definition of GAE is given as:

$$\hat{A}_t^{\mathrm{GAE}} = \sum_{l=0}^{T} (\gamma\lambda)^l \delta_{t+l}, \tag{A.6}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the TD error, γ is the discount factor, and λ is a hyperparameter that controls the bias-variance trade-off.

During P3O training, generalized advantage estimation is applied to both task rewards and robustness-related rewards. This enables the agent to identify augmentation edges that yield stronger improvements in system efficiency and robustness, balance the inherent trade-off between these objectives, and select actions that jointly enhance performance and robustness.

3. Parameter Update

After computing the advantage functions, the parameters of the policy network and the value networks can be updated to optimize model performance. The update procedures are detailed as follows.

(1) Update Policy Network $\pi(\theta)$

The policy update in P3O follows the same multi-step inner-loop optimization scheme as PPO, where multiple gradient updates are performed using the same batch of sampled data. Moreover, P3O transforms the constrained policy iteration problem into a loss function that can be optimized directly using first-order methods by combining the clipped surrogate objective with an exact penalty formulation [83]. Specifically, constraint violations are incorporated into the objective as penalty terms via ReLU($\max\{0,\cdot\}$), and proves that when the penalty coefficient κ is sufficiently large, the resulting penalized problem shares the same set of optimal solutions as the original constrained problem. The exactness of this finite penalty factor constitutes one of the theoretical foundations of P3O. During data sampling, the same clipped surrogate objective as in PPO is adopted to restrict the variation of the importance sampling ratio $r(\theta) = \pi_\theta(a|s)/\pi_{\theta_k}(a|s)$, thereby maintaining the proximal nature of policy updates. Here, π_{θ_k} denotes the old policy used to sample the current trajectory set at the k-th policy update, and π_θ denotes the updated policy during optimization. The final policy optimization objective of P3O is given as:

$$L_{\mathrm{P3O}}(\theta) = L_r^{\mathrm{CLIP}}(\theta) + \kappa \cdot \max\{0, L_c^{\mathrm{CLIP}}(\theta)\}. \tag{A.7}$$

The clipped surrogate of the reward term is consistent with that of standard PPO:

$$L_r^{\mathrm{CLIP}}(\theta) = \mathbb{E}\left[-\min\left(r(\theta)\hat{A}_r, \mathrm{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_r\right)\right]. \tag{A.8}$$

The constraint term adopts a clipped form that is more sensitive to constraint violations, and includes a threshold-related offset term $(1-\gamma)(J_c(\pi_k) - d)$:

$$\begin{aligned} L_c^{\mathrm{CLIP}}(\theta) = \mathbb{E}\Big[&\max\left(c(\theta)\hat{A}_c, \mathrm{clip}(c(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_c\right) \\ &+(1-\gamma)J_c(\pi_k) - d\Big], \end{aligned} \tag{A.9}$$

where ϵ denotes a hyperparameter, and d represents the equivalent formulation of the robustness threshold, defined as $d = -(R_{\mathrm{thr}} - R(G_0))$ (see Section 3.3.1). Positive values of L_c^{CLIP} indicate increased constraint pressure. Through the activation of the penalty term through $\max\{0,\cdot\}$, policy updates can be pulled back toward the boundary of the feasible region while improving returns. In practical training, we adopt the combination of advantage normalization and a fixed penalty coefficient κ as recommended by P3O to improve stability. The loss function $L_{\mathrm{P3O}}(\theta)$ is then used to update θ via gradient descent:

$$\theta \leftarrow \theta - \eta \nabla_\theta L_{\mathrm{P3O}}(\theta), \tag{A.10}$$

where η denotes the learning rate, and ∇_θ represents the gradient of the parameter θ. Meanwhile, to prevent the policy from leaving the proximal region during penalty amplification, a KL early stopping mechanism is introduced during policy updates to enforce trust-region control, thereby limiting approximation errors and stabilizing learning [71]:

$$\delta^{\mathrm{KL}} = \mathbb{E}_{s\sim d_{\pi_\theta}}\left[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s) \| \pi_\theta(\cdot \mid s))\right]. \tag{A.11}$$

If $\delta^{\mathrm{KL}} \notin [\delta_k^-, \delta_k^+]$, the parameter update for the current batch is terminated early.

(2) *Update Value Network $V^r(\phi^r)$ and $V^c(\phi^c)$*

After each policy update, P3O updates the reward value function and the cost value function separately using supervised regression, by minimizing the mean squared error between the outputs of the value networks and the corresponding discounted returns. Under the generalized advantage estimation framework, the corresponding value regression target is given by:

$$\hat{V}_t^r = V^r(s_t;\phi_k^r)+\hat{A}_t^r, \quad \hat{V}_t^c = V^c(s_t;\phi_k^c)+\hat{A}_t^c. \quad \text{(A.12)}$$

This target is equivalent to the λ-return of TD, which can incorporate multi-step information while controlling variance, thereby improving the stability of value learning [84]. Based on the above regression targets, the reward value network and the cost value network are updated using the mean squared error (MSE) loss:

$$L_V^r(\phi^r) = \mathbb{E}_{(s_t,\hat{r}_t)\sim D_k}\left[(V^r(s_t;\phi^r) - \hat{V}_t^r)^2\right], \quad \text{(A.13)}$$

$$L_V^c(\phi^c) = \mathbb{E}_{(s_t,\hat{r}_t)\sim D_k}\left[(V^c(s_t;\phi^c) - \hat{V}_t^c)^2\right]. \quad \text{(A.14)}$$

The dual Critic updates described above provide two essential learning signals for policy optimization in P3O. The reward advantage guides policy updates toward improved system performance, while the cost advantage captures constraint pressure and is incorporated into the penalty term. Together, these two signals support the feasibility of simultaneously improving system efficiency and maintaining robustness boundary constraints during proximal policy updates.

Bingyu Zhu received the B.E. degree from Beihang University, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Reliability and Systems Engineering, Beihang University, China. His research interests include complex system reliability, intelligent network design, complex network robustness and reinforcement learning. (Email: zhuby@buaa.edu.cn)

Shanghan Li received the B.Eng. degree in Electrical Engineering and Automation from Nantong University, Nantong, China, in 2017, and the M.Eng. degree in Control Engineering from Kunming University of Science and Technology, Kunming, China, in 2020. He is currently pursuing the Ph.D. degree in Systems Engineering with the School of Reliability and Systems Engineering, Beihang University, Beijing, China. His current research interests include low-altitude aviation system-of-systems architecture design based on safety decision-making and artificial intelligence. (Email: by2214110@buaa.edu.cn)

Yimeng Liu received the Ph.D. degree in the School of Reliability and Systems Engineering from Beihang University, China, in 2024. She is currently a Postdoctoral Researcher with Hangzhou International Innovation Institute, Beihang University, China. Her research interests are in complex networks and system reliability, and safety test in low-altitude system of systems. (Email: liuyimeng94@163.com)