Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

SPID: a deep reinforcement learning-based solution framework for siting low-altitude takeoff and landing facilities[#]

Xiaocheng LIU, Meilong LE[‡], Yupu LIU, Minghua HU

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

E-mail: lxc2307084@nuaa.edu.cn; lemeilong@126.com; liuyupu@nuaa.edu.cn; minghuahu@nuaa.edu.cn

Received July 28, 2025; Revision accepted Nov. 26, 2025; Crosschecked Dec. 8, 2025

Abstract: Siting low-altitude takeoff and landing platforms (vertiports) is a fundamental challenge for developing urban air mobility (UAM). This study formulates this issue as a variant of the capacitated facility location problem, incorporating flight range and service capacity constraints, and proposes SPID, a deep reinforcement learning (DRL)-based solution framework that models the problem as a Markov decision process. To handle dynamic coverage, the designed DRL framework-based SPID uses a multi-head attention mechanism to capture spatiotemporal patterns, followed by integrating dynamic and static information into a unified input state vector. Afterward, a gated recurrent unit (GRU) is used to generate the query vector, thereby enhancing sequential decision-making. The action network within the DRL network is regulated by a loss function that integrates service distance costs with unmet demand penalties, enabling end-to-end optimization. Subsequent experimental results demonstrate that SPID significantly enhances solution efficiency and robustness compared with traditional methods under flight and capacity constraints. Especially, across the social performance metrics emphasized in this study, SPID outperforms the suboptimal solutions produced by traditional clustering and graph neural network (GNN)-based methods by up to approximately 29%. This improvement comes with an increase in distance-based cost that is kept within 10%. Overall, we demonstrate an efficient, scalable approach for vertiport siting, supporting rapid decision-making in large-scale UAM scenarios.

Key words: Low-altitude planning; Vertiport siting; Deep reinforcement learning; Algorithm exploration
https://doi.org/10.1631/FITEE.2500534 **CLC number:** TP181

1 Introduction

Urban air mobility (UAM) is the low-altitude transportation of passengers and cargo within urban areas. It typically involves electric vertical takeoff and landing (eVTOL) aircraft, drones, and other low-altitude flying vehicles (Federal Aviation Administration, 2023). UAM represents a key direction for transforming urban transportation. To underscore its growth trajectory, the National Aeronautics and Space Administration (NASA) predicted that 500 million drone flights would have occurred in urban space by 2030 (National Aeronautics and Space Administration, 2020). Governments and industry stakeholders are actively promoting UAM to support this development, with low-altitude logistics being considered one of its most promising applications. For instance, Amazon launched drone delivery pilots in California and Texas at the end of 2022. Additionally, vertiports represent critical UAM infrastructure, supporting various related operations. Thus, the selection of vertiport sites is crucial to ensure efficient UAM development, making the scientific and rational planning of their layout a core issue (Deloitte, 2023).

[‡] Corresponding author

[#] Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2500534) contains supplementary materials, which are available to authorized users

ORCID: Xiaocheng LIU, https://orcid.org/0009-0001-4105-7789; Meilong LE, https://orcid.org/0000-0002-1748-0819; Yupu LIU, https://orcid.org/0009-0003-1355-6104

© Zhejiang University Press 2025

Site selection for UAM vertiports transcends a static planning task; it comprises a multi-stage decision-making process. The initial stage requires evaluating site selection costs and service levels under various project configurations (e.g., the number of facilities, capacity, and budget) to determine the optimal project configuration. Additionally, as UAM vertiports serve as the essential social infrastructure, their siting must encompass a broad management perspective, considering multiple social indicators (Berger, 2023). Moreover, as the UAM vertiport siting represents a strategic planning issue, the resulting plan must be forward-looking, requiring the current decisions to assess future service levels. Thus, this issue necessitates the integration of UAM operational constraints and site planning requirements, thereby balancing construction costs and service levels with other social considerations. To solve this complex issue, which requires multiple rounds of scheme comparison, any selected methodology must be efficient and highly generalizable.

The UAM vertiport site selection issue falls within the category of urban facility location problems, and such classical combinatorial optimization problems are traditionally solved using mathematical programming solvers or heuristic algorithms. Although mathematical solvers yield exact solutions, their significant time and memory requirements complicate computation as the problem size increases. Conversely, heuristic algorithms improve computational efficiency but often sacrifice solution quality and generalizability. Moreover, accelerating the solving process (computational efficiency) with heuristic algorithms requires the establishment of targeted rules based on the existing data and problem characteristics, which can be time-consuming and challenging. Consequently, existing solution approaches face a dilemma in practical engineering applications, struggling to balance computational efficiency and solution accuracy.

Deep reinforcement learning (DRL) methods automatically learn solving strategies through data-driven approaches, enabling them to rapidly generate feasible solutions to new problems. The DRL agent can be trained to capture the complex spatial interplay between demand and capacity, eliminating the need for hand-crafted rules and exhibiting strong generalization across scenarios. Thus, DRL can significantly accelerate the site selection process. During the early stage of decision-making, DRL reduces the time required to compare schemes under various parameter settings. During the final decision stage, DRL provides high-quality, initial, feasible solutions for the mixed-integer linear programming (MILP) model, thereby significantly reducing the computation time for exact solutions. Notably, the combination of DRL and MILP achieves an optimal balance between efficiency and accuracy, thus effectively satisfying practical engineering requirements.

In summary, we explore DRL methods to solve the UAM vertiport site selection problem, with the following main contributions:

1. Formulating the UAM site selection problem as a variant of the capacitated facility location problem (CFLP). This variant incorporates capacity constraints and allows unmet demands, thereby accurately reflecting the supplementary role of UAM within existing transportation frameworks. CFLP is more practical and exhibits significant engineering relevance.

2. Developing a DRL-based solution framework and deploying it as a high-quality, initial, feasible solution for MILP exact computation. This integrated approach ensures an optimal balance among computational efficiency, robustness, and scalability.

3. Introducing three key performance indicators (geographic coverage, effective coverage, and facility utilization) to quantify the social benefits and foresight of the final site selection plans.

2 Literature review

This study is aimed at optimizing the selection of urban low-altitude vehicle takeoff and landing sites (vertiports), considering multiple candidate bases and the demand distribution of customer nodes. As low-altitude logistics represents one of the most rapidly advancing applications in the UAM industry, most existing studies focus on logistics- and distribution-oriented vertiport site selection. The explored landing bases are conventionally referred to as unmanned aerial vehicle (UAV) centers, distribution hubs, hospitals, UAV stations, warehouses, drone hives, or logistics platforms. Additionally, their deployment for communication, particularly as temporary signal-enhancing base stations, has emerged as a research hotspot. Despite these

diverse application potentials, the aforementioned platforms share the common function of service provision, thus conceptually aligning with the takeoff and landing sites explored in this study. Therefore, our literature review is organized around three key UAM applications: logistics and distribution, UAV-based communication, and UAM-infrastructure planning.

Numerous studies in the field of low-altitude logistics and distribution mainly address landing site location. For instance, Shavarani et al. (2018) proposed a hierarchical model to minimize the total system costs by optimizing the locations of UAV launch and charging stations. Shavarani et al. (2019) further extended this model with a fuzzy multilevel model, incorporating uncertain demand and costs. Kim et al. (2019) explored stochastic siting under uncertain flight distances, and Meskar and Ahmadi-Javid (2024) investigated battery consumption rates based on payload effects. Abdel-Basset et al. (2024) introduced a multi-objective trajectory optimization algorithm based on *k*-means clustering and greedy sequencing. Meng et al. (2024) integrated clustering, center-of-mass location, and genetic algorithms to coordinate UAVs and ground vehicles for blood delivery tasks. Mahmoodi et al. (2025) proposed a multi-objective site-path optimization model that integrates time windows, dual-service demands, and recharging constraints. This model was solved using the non-dominated sorting genetic algorithm II (NSGA-II) combined with Bayesian risk modeling. Zhang et al. (2025) integrated safety considerations with economic objectives into the selection of urban air logistics hub locations. To address both concerns, a multi-objective MILP model that simultaneously minimizes costs and third-party safety risks is proposed and solved using an NSGA-II that features adaptive operator selection.

In the communication domain, UAVs are explored as mobile base stations. For instance, Gopi and Magarini (2021) applied reinforcement learning (RL) to optimize UAV base station siting, employing data-rate sum as the reward function. Ryu and Kim (2024) modeled partially observable multi-UAV deployment for supporting mobile terrestrial users. Their model was solved using DRL. To address scalability, Premkumar and Van Scoy (2025) clustered customers, followed by site drones, thereby effectively striking a balance between optimality and runtime.

The emergence of the UAM concept has rendered eVTOL platform siting and capacity planning a central issue. For instance, Lynskey et al. (2019) proposed a CFLP model for drone ports, which integrates the traveling salesman problem (TSP) routing and *k*-means clustering. Escribano Macias et al. (2023) framed the siting problem as a *p*-median variant to maximize travel time savings over ground transportation. Kumar et al. (2023) applied graph-based RL for decision support in siting eVTOL infrastructure. They used dual-graph representations, which are passed through graph convolutional networks (GCNs). Zhu et al. (2022) applied column generation and Benders decomposition to address robust siting under post-disaster demand uncertainty. Other notable contributions include hybrid clustering and *p*-median optimization (Rahman et al., 2024), genetic algorithms for infrastructure-accessible eVTOL sitings (Volakakis and Mahmassani, 2024), and data-driven Markov decision process (MDP) formulations for dynamic drone delivery planning (Paul et al., 2024).

Studies also explored travel time valuation for vertiport site selection using iterative weighted *k*-means (Zhao and Feng, 2024). Additionally, Zheng et al. (2024) explored multi-factor logistics hub siting using enhanced particle swarm optimization algorithms. Song and Lee (2025) evaluated the influence of vertiport siting on the transportation network topology, and Lee and Cho (2025) proposed a three-stage geospatial analysis for site selection, encompassing suitability, regulation, and location allocation. Sun et al. (2025) developed a framework for siting coastal UAV monitoring bases, effectively striking a balance between spatial coverage and cost.

The extant studies on the optimization of UAV landing sites conventionally adopt three objective-function types: (1) cost-based objectives for minimizing fixed facility costs, UAV acquisition and operating costs, and transportation costs (some cost-based objectives adopt a two-stage approach, namely, facility and operational cost optimization); (2) time-based objectives for minimizing the total delivery time or completion time (some models incorporate customer utility functions to reflect delay impacts); (3) coverage-based objectives, often employed in communication and medical tasks, maximize demand coverage or service quality. Regarding constraints, most models consider

UAV endurance (measured by distance, time, or energy), payload limits, facility counts, and budget constraints. More complex models also include practical factors, such as time windows, demand uncertainty, network accessibility, charging station queuing, and obstacle-aware routing.

Most studies explored classical facility location models, such as the *p*-median, *p*-center (Hakimi, 1965), and the maximal covering location problem (Church and ReVelle, 1974), to provide a solid theoretical foundation. Regarding solution methods, exact algorithms, primarily those implemented via solvers, including CPLEX and Gurobi, remain the most common, although a few studies proposed customized approaches. Heuristic methods (e.g., genetic algorithms, simulated annealing, or large neighborhood search) are often employed to efficiently handle large-scale or complex problems.

Beyond conventional operations research methods, recent advances in deep learning and DRL have opened new avenues for solving complex optimization problems. Particularly, DRL has demonstrated remarkable capability in learning decision-making policies for combinatorial optimization tasks without requiring extensive domain-specific heuristics. Most extant DRL studies on logistics optimization have primarily explored routing problems, aiming to determine the optimal visit sequences and travel paths based on the fixed facility locations. For instance, Kool et al. (2019) developed an attention-based model using encoder–decoder architectures to learn construction heuristics for the TSP and vehicle routing problem (VRP). Nazari et al. (2018) applied RL to VRP with stochastic demands, demonstrating policy generalization across problem instances. Hottung and Tierney (2020) proposed a neural large neighborhood search for capacitated VRP (CVRP), achieving a competitive performance with traditional methods in addition to offering computational efficiency on large-scale tasks. Overall, these route-planning-centered studies have established a crucial methodological foundation for applying machine learning to logistics optimization. However, this research line addresses a problem that differs fundamentally from facility location decision-making. Route planning operates within a predefined topological network, whereas facility siting involves the determination of the optimal sites within a dispersed demand distribution.

Notably, the integration of graph neural networks (GNNs) with RL has emerged as a potent paradigm for handling graph-structured optimization problems. For instance, Han XY et al. (2024) proposed a novel GNN–DRL integrated framework for resource allocation in computing power networks. They demonstrated that graph-based representations effectively capture the inherent structural dependencies of network optimization problems. Their approach also leveraged the aggregation capability of GNN for neighborhood information, thereby enabling the model to learn generalizable policies across diverse network topologies and scales. More directly relevant to spatial optimization, Liang et al. (2024) developed SPONet, a specifically designed DRL framework for urban spatial decision problems, including billboard siting optimization. They demonstrated that DRL-based approaches can achieve over 20× speedup compared with traditional solvers while maintaining solution quality, thus establishing the feasibility of learning-based methods for spatial facility siting problems. Li et al. (2025) developed an RL-based iterative algorithm to solve capacitated hub location problems in UAV delivery networks. They employed specialized GNN- and gated recurrent unit (GRU)-based policy networks to embed demand and spatial patterns, thereby demonstrating superior performance and robust generalizability across various-scale networks.

Within the UAM domain specifically, several recent studies have started exploring DRL applications. However, these studies have mainly focused on operational and tactical decision-making (e.g., routing and resource allocation) rather than strategic infrastructure planning (e.g., site selection). To address real-time routing and scheduling decisions for active UAM operations, Park et al. (2023) applied multi-agent RL to coordinate cooperative air transportation services. Garcia et al. (2023) developed a DRL framework for intelligent safety management in UAM, focusing on collision avoidance and secure flight operations. Regarding UAM communication infrastructure, Han RX et al. (2022) proposed DRL-assisted spectrum management for cellular-based UAM networks, optimizing wireless resource allocation to support UAM operations. To extend this research domain, Apaza et al. (2023) applied multi-agent DRL under resource constraints to integrate spectrum management with air traffic management. Although these studies demonstrate the

viability of DRL in UAM domains, they mainly address operational-level problems (e.g., service coordination, safety control, and communication-resource allocation) and barely consider strategic facility siting decisions.

In summary, previous DRL-based studies on logistics and transportation primarily focused on solving route planning problems, such as TSP, VRP, and CVRP, operating under the assumptions of fixed depot locations and predetermined visit sequences. To bridge this research gap, this study mainly addresses the strategic facility siting decision-making process. This involves determining the optimal vertiport locations and capacity allocations before the commencement of operations. Overall, this study represents a shift in research focus from tactical-level route optimization to strategic-level infrastructure planning, requiring distinct state representations, action spaces, and reward structures within the DRL framework. Following the approach of Liang et al. (2024), we explore the UAV facility siting problem from a macro-management perspective, formulating this problem as a CFLP variant that incorporates aerial operational constraints and solving it using a DRL-based framework. This approach enables an end-to-end solution to the facility siting problem under partial demand satisfaction, thereby providing a scalable alternative to traditional or heuristic methods.

3 Models

3.1 Capacitated facility location problem

In classical facility siting models, the p-median model minimizes the total travel time, making it suitable for time-sensitive services, such as UAM passenger transport. Contrarily, the maximal covering location model is designed to satisfy the maximum possible demand using a limited number of facilities and is traditionally applied in emergency logistics. However, since vertiports function as critical capacity nodes in a UAM network, their placement cannot be solely determined by the service distance. Thus, the site selection must also account for strict capacity constraints and the fact that a portion of the demand can be deferred. This is particularly relevant because UAM is still at its developmental stage, and many of its potential trips currently exhibit mature alternative modes. The low-altitude takeoff and landing sites can be considered low-altitude analogs of civil aviation terminal areas. Therefore, they are characterized by strict capacity limits on the number of daily operations.

The classical CFLP requires the full satisfaction of all demands. However, the problem addressed in this study allows for partially unmet demand, rendering it a variant of traditional CFLPs.

The objective of the standard CFLP is to select p facility locations within a given network or space, thereby minimizing the total service cost (typically represented by the sum of distances between facilities and demand points). Conversely, our model relaxes the full demand satisfaction constraint, and directly incorporates an unmet demand penalty into the objective function. This variant model offers a better capture of the phenomena observed in UAM operations, specifically where limited facility capacity prevents part of the demand from being immediately served. It also accurately reflects early-stage UAM systems, which are typically characterized by a few facilities and unevenly distributed demands. In this way, the model simultaneously enforces capacity constraints and preserves operational flexibility and cost efficiency.

First, to formalize the problem, we define the set of UAM demand points as $N=\{1, 2, \cdots, n\}$ and the set of candidate takeoff and landing sites for UAVs as $M=\{1, 2, \cdots, m\}$. The decision variables in the proposed model are defined as follows:

$$x_{ij}=\begin{cases}1, & \text{demand point } i \text{ is served by facility } j,\\ 0, & \text{otherwise,}\end{cases}$$

$$y_j=\begin{cases}1, & \text{facility is selected,}\\ 0, & \text{otherwise.}\end{cases}$$

Second, let d_{ij} denote the distance between demand point i and takeoff and landing site j, n_i the total demand at point i, and C_j the capacity limit of site j. Further, let N_i represent the demand portion at point i that can be served by site j, and let the neighborhood set for each demand point be defined as $N_j=\{j\in M|d_{ij}\le D_j\}$, where D_j is the service radius of site j. Finally, let p denote the maximum number of takeoff and landing sites that can be constructed.

Based on these definitions, the proposed CFLP variant allows for partially unmet demand, which can

be served in a delayed or deferred manner. The constraints of this model are formulated as follows:

$$\text{Minimize } z=\sum_{i=1}^{n}\sum_{j=1}^{m}\left[d_{ij}x_{ij}+U\left(\sum_{j=1}^{m}C_j-\sum_{i}^{n}n_i\left(1-x_{ij}\right)\right)\right] \quad (1)$$

$$\text{subject to } \sum_{j=1}^{m}x_{ij}\le 1,\ i=1,2,\cdots,n, \quad (2)$$

$$\sum_{j=1}^{m}y_j=p,\ i=1,2,\cdots,n, \quad (3)$$

$$x_{ij}\le y_j,\ i=1,2,\cdots,n,\ j=1,2,\cdots,m, \quad (4)$$

$$\sum_{i=1}^{n}n_ix_{ij}\le C_jy_j,\ j=1,2,\cdots,m, \quad (5)$$

$$x_{ij},y_j\in\{0,1\},\ i=1,2,\cdots,n,j=1,2,\cdots,m. \quad (6)$$

Constraint (2) ensures that each demand point is assigned to a maximum of one facility, and constraint (3) limits the total number of facilities selected for construction to a predefined number p. Constraint (4) ensures that the demand can only be satisfied by established facilities, and constraint (5) guarantees that the total demand allocated to each facility does not exceed its maximum service capacity. Additionally, U in the objective function represents the cost coefficient generated at the point where the demand is not satisfied.

3.2 Reformulation of the CFLP model as a Markov decision process model

To solve the problem via the DRL approach, the CFLP model must first be reformulated as an MDP model. This reformulation enables the utilization of DRL for the iterative update of the policy or value function based on temporal dynamics. In this study, the MDP formulation is based on the REINFORCE algorithm (Williams, 1992). Fig. 1 illustrates the schematic of the RL process. In this process, an intelligent agent (SPID) interacts with the environment, guided by the MDP-based model, to determine optimal siting decisions for the takeoff and landing facilities. The components of the MDP framework, defined as a quintuple (S, A, P, R, γ), are specified for the CFLP as outlined below. Table 1 presents a summary of all notations and variables employed in the MDP formulation.

State: the state at time step t (denoted as $s_t=M_t$) represents the solution set, comprising the facility locations selected in the previous $t-1$ steps. M_0 is the initial state, which corresponds to an empty solution set.

Action: an action at each decision step involves the selection of a node to serve as a facility. Here, the action a_t corresponds to the selection of the node z_j, such that $a_t=z_j$ if node z_j is selected as a facility at time step t.

Transition: given the current state s_t, the agent selects an action a_t that adopts the node z_j as a new facility. Immediately after adding z_j to the current solution set M_t, the environment transitions to the next state s_{t+1}, defined as $s_{t+1}=M_{t+1}=(M_t;\ z_j)$.

Discount factor: the discount factor is set as $\gamma=1$, indicating that future rewards are not discounted.

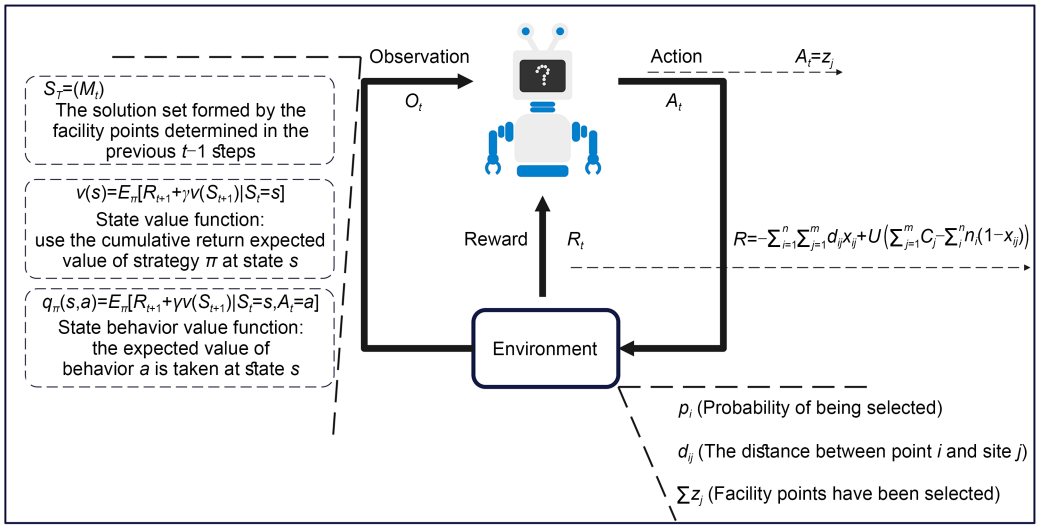

Fig. 1 RL expression of SPID

Table 1 Definitions of notations and variables

State and action variables	Description
s_t	State at time step t
M_t	Solution set of selected facilities up to step t
a_t	Action taken at time step t
z_j	Node selected as the facility at the decision step
Decision variables	**Description**
x_{ij}	Binary variable: 1 if demand i is allocated to facility j, and 0 otherwise
π	Complete solution sequence $\{\pi_1, \pi_2, \cdots, \pi_p\}$
π_t	Facility selected at time step t
$\pi(j)_t$	Set of demand points assigned to facility j at step t
Policy and learning	**Description**
θ	Policy parameters
$p_\theta(\pi\|s)$	Probability of selecting solution π given state s under policy θ
$p_\theta(\pi_t\|s, \pi_{1:t-1})$	Probability of selecting facility π_t given the current state and previous selections
$q_\pi(s,a)$	Action value function
α	Learning rate (step size) for the gradient ascent
Capacity and cost	**Description**
$C_j\pi(j)_t$	Remaining capacity of facility j at time step t
$C_i\pi_t$	Cost of assigning demand point i under policy π at step t
$L(\pi)$	Loss function (total cost)
$L(\theta\|s)$	Expected loss under policy parameters θ
$\nabla L(\pi)$	Loss function gradient

Since the final objective value is treated as the only reward, cumulative reward values $v(s)$ are excluded.

Reward: the optimization direction of the model is driven by the reward function. In this study, the reward is defined as the negative of the objective function in Eq. (7)

$$R=-\sum_{i=1}^{n}\sum_{j=1}^{m}d_{ij}x_{ij}+U\left(\sum_{j=1}^{m}C_j-\sum_{i}^{n}n_i\left(1-x_{ij}\right)\right). \quad (7)$$

This formulation enables the agent to minimize the total service cost and penalize capacity violations.

In the MDP framework, p_θ denotes the policy that governs the probability of the agent selecting an action a_t given the state s_t. At each time step, the agent selects one node as a facility location; this process continues until a total of p facilities are selected, forming the final solution $s_T=M_t$. The above selection process is represented by the following:

$$\pi=\pi_1, \pi_2, \cdots, \pi_p, p\le n, \quad (8)$$

$$p_\theta(\pi|s_t)=\prod_{t=1}^{p}p_\theta\left(\pi_t|s_t,\pi_{1:t-1}\right), p\le n. \quad (9)$$

Each facility j has a fixed capacity C_j, and its remaining capacity is dynamically updated at each time step based on the demand assigned to it. The update rule is defined as follows:

$$C_j\pi(j)_t=C_j-\sum_{i|x_{ij}=1}n_i. \quad (10)$$

For demand allocation, each demand point is assigned to a feasible facility (i.e., a facility that is not overloaded) via a proximity-based allocation strategy denoted as $\pi(j)_t$. Here, $C_j\pi(j)_t$ represents the remaining capacity of facility j at the time of the decision step t. A large penalty coefficient U is introduced (U is set as 10^6 in this study) to penalize infeasible

assignments. The assignment of demand points to facilities is governed by the following rules:

$$j \in \pi, \text{ if } C_j \pi(j)_t > n_i, \tag{11}$$

$$C_i \pi_t = \min\{d_{ij} n_i\}, \forall j \in \pi, d_{ij} \le r, \tag{12}$$

$$C_i \pi_t = U n_i, \text{ if } \pi(j)_t = \varnothing, \tag{13}$$

where r is the coverage radius.

The model is designed to minimize the total cost of satisfying all demand points. By setting the discount factor to $\gamma=1$, the action value function $q_\pi(s, a)$ is defined as the expected sum of rewards:

$$q_\pi(s_k, a) = p_\theta(\pi|s) R_{S_1} + \cdots + p_\theta(\pi|s) R_{S_k}. \tag{14}$$

To solve the model, a Monte–Carlo-based policy gradient method is employed. This algorithm updates the parameters θ using stochastic gradient ascent. The update rule is expressed as follows:

$$\theta_{t+1} = \theta_t + \alpha \nabla L(\pi). \tag{15}$$

Here, an increase in $L(\pi)$ indicates that the current policy yields higher rewards, and parameter updates are implemented in the direction that maximizes the anticipated reward. The loss function for training is defined as $L(\theta|s) = E_{p_\theta(\pi|s)}[L(\pi)]$, where

$$L(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij} x_{ij} + U\left(\sum_{j=1}^{m} C_j - \sum_{i}^{n} n_i (1 - x_{ij})\right). \tag{16}$$

4 Methodology

The UAM vertiport siting problem shares common characteristics with traditional ground facility siting problems, as both can be considered CFLPs. However, as UAM vertiports are integrated facilities featuring ground and air operations, site selection must account for various constraints that are closely related to low-altitude operations.

4.1 Analysis of the characteristics of urban air mobility vertiport siting problem

The UAM vertiport siting problem can be analyzed from multiple perspectives, including air traffic control regulators, operators, and demand stakeholders. This problem can also be analyzed based on the full UAM operational process. To ensure research generality, the problem is constructed from the perspective of the entire UAM operational process, focusing on the strategic location selection phase for vertiports.

UAM operations can be systematically divided into three sequential decision-making stages. The first stage involves site feasibility and candidate set generation, which is part of the upstream preparatory analysis. At this stage, spatial analysis is used to identify candidate vertiport sites. The key factors considered at this stage include airspace constraints, structural feasibility, regulatory compliance, land ownership, and acquisition costs. The output of this stage is a discrete set of candidate locations, each of which is associated with its development cost and operational constraints, thus providing the necessary input for subsequent models.

The second stage involves strategic location selection and capacity planning, forming the core of this study. Based on the candidate sites generated at the first stage, decisions are made regarding the specific location selection, demand distribution from the origin node to the selected sites, and system-level evaluation of performance metrics, such as coverage, accessibility, and service level. The proposed SPID model addresses the problem at this stage, where the input parameters (candidate locations, demand nodes, coverage radius r, and capacity) implicitly include upstream constraints, and the model output (selected facility sites) provides the key input for downstream operational planning.

The third stage involves operational planning and four-dimensional trajectory design, which serves as the downstream extension of the location decision. Once the facility locations are determined, this stage focuses on fine-tuning operational issues, including four-dimensional trajectory planning, flight scheduling, energy management, and detailed demand–capacity matching. Although these operational decisions are crucial for system implementation, they are essentially downstream issues that are derived directly from the site selection results. The location model does not explicitly optimize trajectory details; rather, it ensures the operational feasibility and efficiency of the selected locations by maintaining proper facility spacing and adhering to operational analysis-derived capacity limits.

In summary, the strategic site selection for UAM vertiports serves as a crucial link between upstream feasibility analysis and downstream operational planning. To clarify the core constraints at each stage and their representation in the proposed model, Table 2 summarizes the stage divisions, key constraints, and corresponding input parameters across the UAM operational process. Furthermore, it systematically explains how physical, airspace, and operational constraints from each stage are integrated into the strategic site selection phase. Specifically, the physical and regulatory constraints from the first stage are internalized as input parameters for the second stage, laying the foundation for the feasibility of the site selection solutions. Concurrently, operational efficiency constraints from the third stage are proactively explored in the site selection phase via pre-set candidate facility locations and capacity limitations, thus ensuring the feasibility and effectiveness of subsequent strategic-level operations.

To further evaluate the comprehensive performance of the site selection solution from a system perspective and respond to the strategic decision-making needs of public facility planning, the following set of performance metrics is introduced in this study. These metrics quantify and reflect multiple objectives, such as facility coverage, resource utilization, and operational efficiency.

1. Geographic coverage rate S_g:

$$S_g=\left(1-\frac{N_g}{N_u}\right)\times 100\%, \tag{17}$$

where N_g is the number of geographically uncovered users and N_u is the total number of users.

This indicator reflects the proportion of users who are geographically covered by a minimum of one facility.

2. Effective coverage rate S_e:

$$S_e=\frac{N_e}{N_u}\times 100\%, \tag{18}$$

where N_e is the number of users assigned to a facility without exceeding its capacity.

This indicator measures the proportion of users who are covered and assigned to a facility without violating the capacity constraints.

3. Facility utilization rate S_f:

$$S_f=\frac{\text{Total allocated demand}}{\text{Total facility capacity}}\times 100\%. \tag{19}$$

This indicator represents the overall utilization of facility resources relative to their maximum service capacity.

4. Weighted distance cost C_w:

$$C_w = d_{ij}x_{ij}, \left\{i|x_{ij} = 1\right\}. \tag{20}$$

This metric reflects the total demand-weighted Euclidean distance between each demand point and its assigned facility. Lower values indicate shorter average travel distance for users, reflecting increased operational efficiency.

5. Unweighted distance cost C_u:

$$C_u=x_{ij}, \left\{i|x_{ij} = 1\right\}. \tag{21}$$

This metric reflects the average Euclidean distance between demand points and their assigned facilities, without demand weighting. This metric provides a distance-based evaluation that is independent of the demand magnitude.

4.2 Problem restoration

CFLP is inherently a combinatorial optimization problem, where the vertiport site selection does not

Table 2 Correspondence between the input parameters of SPID and the key constraints at each operational stage of urban air mobility

Stage	Key factor	SPID model's input parameters
1	Physical and regulatory constraints (e.g., airspace, structure, and regulations)	Pre-screened set of candidate sites Inherent capacity parameters of the sites
2	Strategic objectives (e.g., coverage, capacity, demand, and budget)	
3	Operational efficiency constraints (e.g., airspace management, scheduling, and energy management)	Candidate site locations: implicit airspace management feasibility; Service radius and facility capacity limitations: support operational scheduling and energy management

follow a specific sequence. However, when CFLP is reformulated as an MDP for solution via DRL, site selection is discretized and becomes sequential.

This artificially imposed decision order breaks the inherent symmetry of the original combinatorial problem by mapping essentially equivalent solutions onto different decision trajectories. This can significantly expand the state space and introduce "order bias," where the model is more likely to obtain high-quality solutions under certain selection orders than others.

To address these issues, Liang et al. (2024) proposed a dynamic coverage vector $\boldsymbol{D}_i$ method that effectively mitigates the adverse effects of decision sequencing on combinatorial optimization problems.

$\boldsymbol{D}_i$ is introduced for each demand point i to represent its coverage state. This vector is dynamically updated after each siting action a_t. Notably, each $\boldsymbol{D}_i$ represents a column vector with elements in the range [0, 1]. Thus, the closer an element is to 0, the less likely the corresponding demand point is to be selected in subsequent steps, indicating that it is either already sufficiently covered or is very close to a selected facility. All coverage vectors $\boldsymbol{D}_i$ are initially set to $\mathbf{1}$:

$$\boldsymbol{D}_i=\mathbf{1},\ i=1,2,\cdots,n. \tag{22}$$

Let π_t denote the node selected at time step t, and let $C_{r(\pi_t)}$ represent the set of demand points within a coverage radius r of node π_t. The points in $C_{r(\pi_t)}$ are sorted in ascending order of distance from π_t. Let $w_{i(\pi_t)}$ denote the rank (position index) of demand point i in this ordered list. Let $K_{r(\pi_t)}$ be defined as the subset of $C_{r(\pi_t)}$ comprising demand points that are actually served by facility π_t. Let $N_{C_{r(\pi_t)}}$ be the cardinality of $K_{r(\pi_t)}$ and c_i the number of times point i is covered. The iterative update of $\boldsymbol{D}_i$ during decoding is governed by the following equations:

$$c_i=\sum_{\pi_0}^{\pi_{t-1}}1,\ i\in C_{r(\pi_t)}, \tag{23}$$

$$\boldsymbol{D}_i=\boldsymbol{D}_i\times\frac{w_{i(\pi_t)}}{N_{C_{r(\pi_t)}}}\times\frac{1}{c_i},\ i=1,2,\cdots,n, \tag{24}$$

$$\boldsymbol{D}_i=\begin{cases}\mathbf{0}, & \text{if } i\in K_{r(\pi_t)},\\ \mathbf{1}, & \text{otherwise.}\end{cases} \tag{25}$$

The above update rules indicate that the more frequently node i is covered, or the higher its priority in the distance-based ordering (smaller $w_{i(\pi_t)}$), the smaller $\boldsymbol{D}_i$ becomes, thereby reducing its likelihood of being selected in future steps. Notably, the coverage degree and proximity to the selected node are both incorporated into the decision-making process.

To accurately calculate the probability of selecting a node as a facility at each t, the model learns a refined representation that integrates $\boldsymbol{D}_i$ and a learnable weight matrix $\boldsymbol{W}^1$, yielding a dynamic embedding vector $\boldsymbol{F}_i$ that is defined as follows:

$$\boldsymbol{F}_i=\boldsymbol{D}_i\boldsymbol{W}^1. \tag{26}$$

4.3 Implementation of the SPID method

Based on the introduction of $\boldsymbol{D}_i$, the information required for model inference is categorized into two types: static and dynamic information. Static information comprises the two-dimensional coordinates and demand values of each point, representing fixed attributes that remain constant throughout model training. Conversely, dynamic information reflects system states, such as the coverage status of each demand point and the capacity utilization of each facility.

Fig. 2 shows that the static and dynamic information within the spatial perceiver module of SPID is first transformed from the original representations into high-dimensional embeddings that can be processed by the model. The static information is subsequently divided into two separate streams to enhance the ability of the model to learn and represent different feature types.

One stream of static embeddings is employed as input to the inferential decision module of SPID, where a GRU component captures the temporal dependencies and contextual information of the decision sequence, including historical decisions, current states, and constraints to be satisfied. The other stream is integrated with dynamic information and passed into the scaled dot product attention component, enabling the computation of attention weights through a multi-head attention mechanism.

Both branches complement each other; the GRU facilitates sequence modeling, and the attention mechanism focuses on feature selection. Synergistically, they guide the model to determine the static features

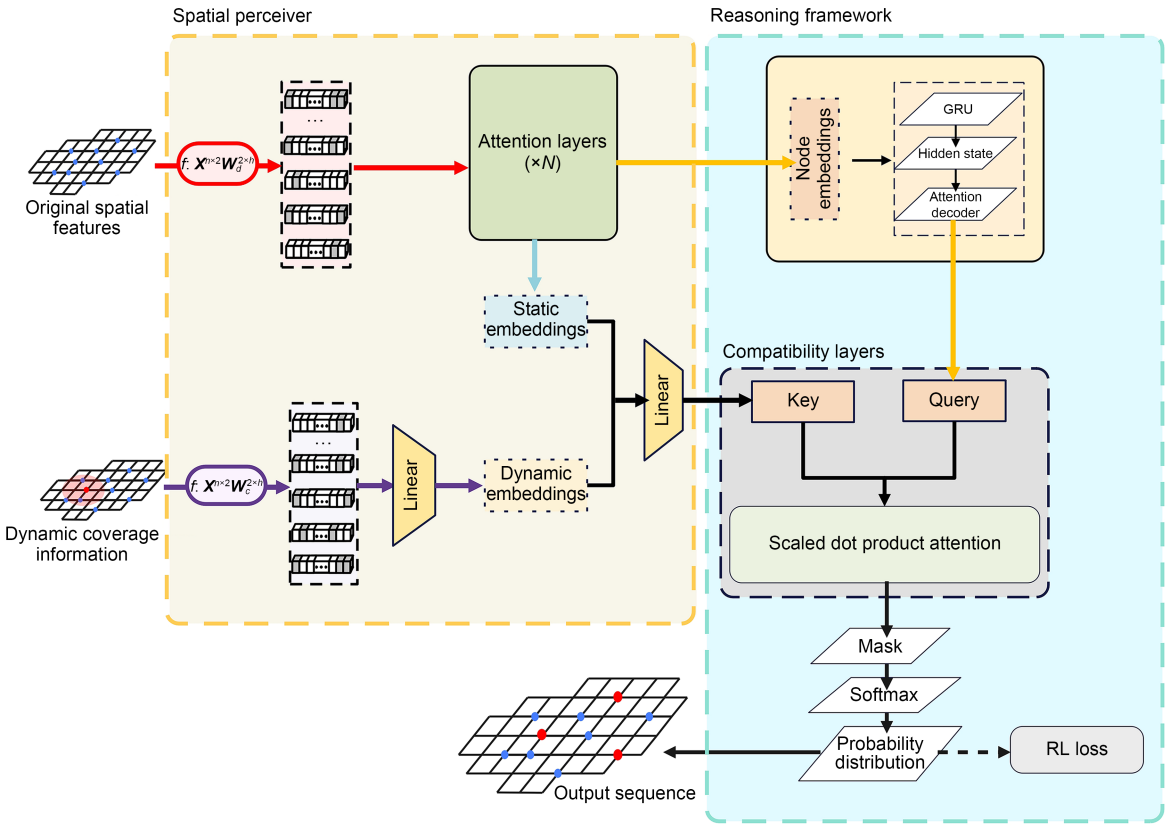

Fig. 2 SPID framework

that must be emphasized at each decision point, thereby achieving adaptive feature selection and weight assignment.

To ensure solution feasibility, SPID incorporates a masking mechanism. This approach enables the direct integration of problem constraints and decision-making rules into the inference process of the neural network. Specifically, SPID employs three types of mask matrices: the access matrix (for recording historical selections), the service matrix (for tracking cumulative demand fulfillment), and the dynamic coverage matrix (for representing the current decision features). The current decision features are encoded in the form of a matrix comprising $\boldsymbol{D}_i$.

After each siting decision, SPID updates all three mask matrices to ensure the correct handling of the constraints of the facility siting problem. These constraints include capacity limits, service radius restrictions, and duplicate selection avoidance.

Finally, the model generates a probability distribution over all candidate points using a softmax function, after which it selects the point with the highest expected reward as the current decision. Fig. 3 illustrates the inference and decision-making workflow of SPID.

5 Experiment and results

This section presents a thorough experimental validation of the SPID framework for addressing the UAM vertiport siting problem. The evaluation is structured into three progressive phases: (1) ablation experiments, which assess the contribution of each component to the overall solution quality; (2) comparison experiments, in which SPID is benchmarked against mainstream clustering- and GNN-based methods; (3) experiments to examine the adaptability of the framework across the three UAM development stages, highlighting its ability to adjust to the evolving UAM operational scenarios. The SPID training framework is detailed in the supplementary materials.

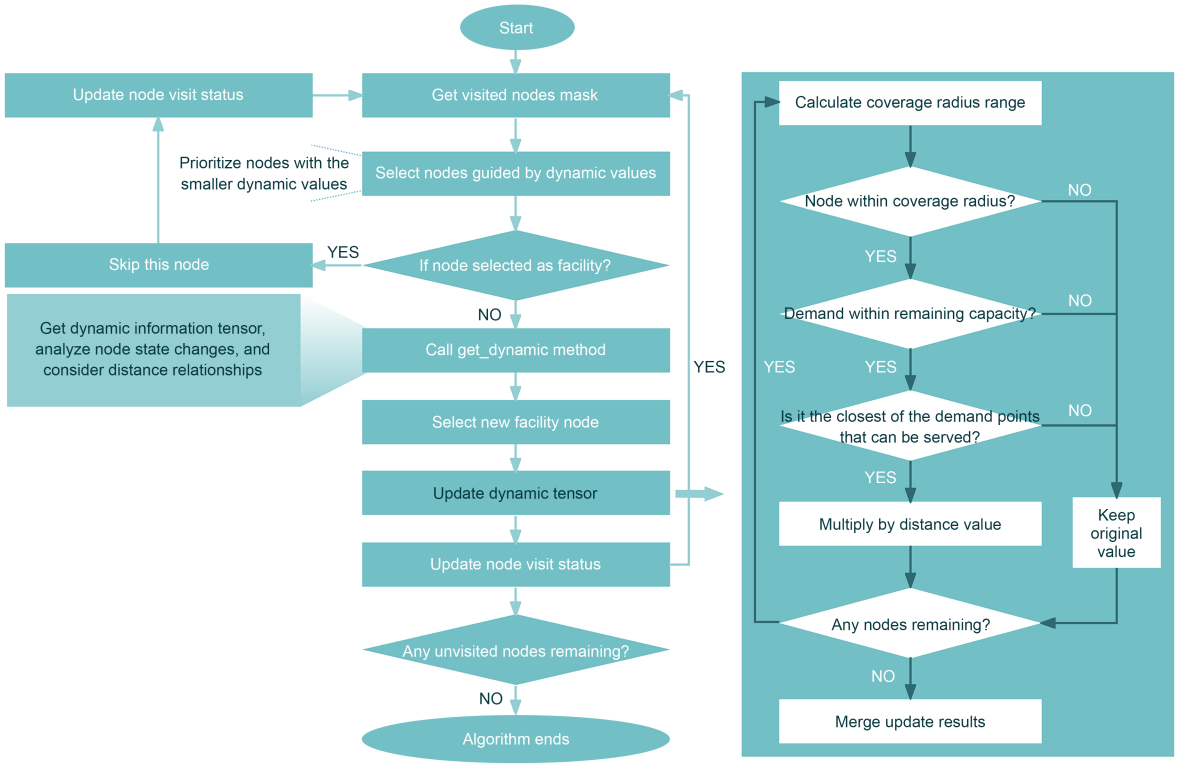

Fig. 3 SPID RL state workflow chart

5.1 Ablation experiments and results

The SPID framework integrates a multi-layer graph attention network (GAT) encoder with a decoder that combines GRUs and an attention mechanism. To isolate and quantify the contributions of each architectural component to the quality and computational efficiency of the UAM facility location solution, ablation experiments are conducted using two model variants: one without the GRU (GRU-free) and the other without the attention mechanism (attention-free). The GRU-free variant excludes recurrent-context modeling and replaces the GRU update with a simple embedding averaging: $h_t=\text{mean}(\{v_i|i\in\text{selected}\})$. The attention-free variant excludes the adaptive node-scoring mechanism and assigns a uniform probability to unselected locations: $u_j=0$, $\forall j$.

To facilitate a fair comparison, all model variants share the same training protocol. The network uses a 128-dimensional embedding space, incorporating three layers of GAT encoders (each equipped with eight attention heads), and a 128-dimensional decoder. Training is conducted using the Adam optimizer at a learning rate of 5×10^{-5} by a 1×10^{-4} weight decay, a batch size of 4, and a gradient clipping norm of 10. Each model is trained for up to 640 epochs on 1000 problem instances, with early stopping (patience=15). The architecture and hyperparameter configurations of the SPID framework are detailed in the supplementary materials. Table 3 reports the test performance obtained across the 1000 instances, and Fig. 4 shows the value function trajectories for all three models during training in the ablation study.

The results of the ablation experiments demonstrate that the full SPID model delivers the best performance, thus validating the synergistic contribution of the GRU and attention mechanisms. Specifically, the GRU component improves the performance of the model by 0.094% by maintaining sequential context, whereas the attention mechanism enhances the performance by 0.134% through adaptive node selection. This phenomenon arises because the baseline solution quality is already close to its upper bound, leaving only limited room for the two components to further improve the solution quality. In terms of runtime, simplifying the model architecture consistently leads to faster inference. However, the moderate computational overhead is justifiable for UAM infrastructure

Table 3 Performance table of the test set

Model	Average loss	Standard deviation	Minimum value	Maximum value	Time (ms)
Complete model	**17.4123**	0.5234	15.8923	19.2341	23.4
GRU-free model	17.4287	0.5412	**15.8734**	19.3456	19.8
Attention-free model	17.4356	**0.5123**	15.9123	**19.1234**	**18.7**

The bold indicates the best results

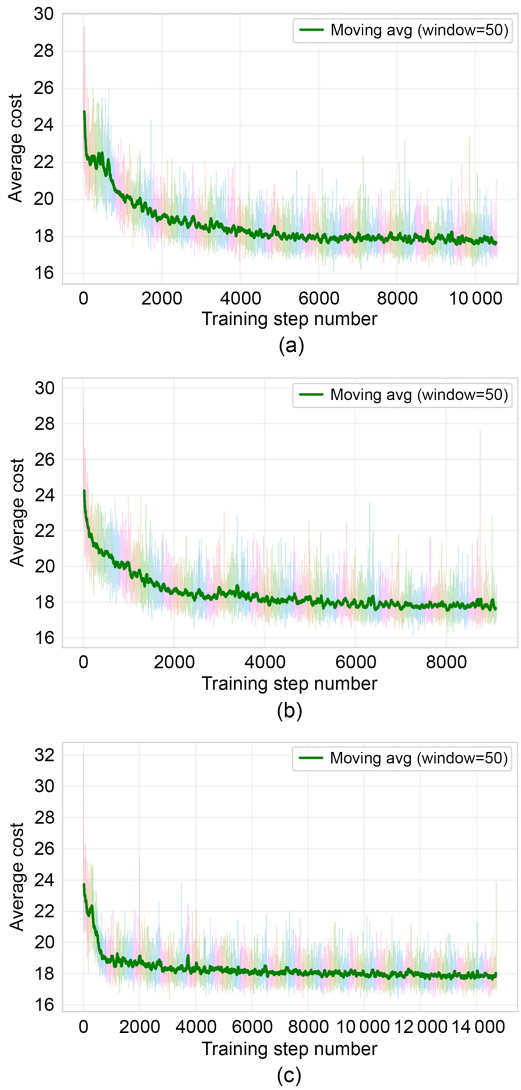

Fig. 4 Model training value function curve: (a) complete model; (b) GRU-free model; (c) attention-free model. Avg: average

planning, where facility siting decisions have long-term strategic implications.

The training dynamics further clarify the functional roles of each component. The SPID model demonstrates the smoothest convergence trajectory, achieving the optimal validation loss at the 50th epoch while maintaining high stability throughout subsequent training. Conversely, the GRU-free variant achieves the fastest convergence, although it ultimately yields slightly lower solution quality. However, the attention-free variant achieves a substantial increase in convergence time. Overall, the ablation experiments indicate that the GRU and attention mechanisms synergistically enhance the model performance, with the SPID model achieving the best balance among the accuracy, stability, and convergence efficiency. Furthermore, the SPID model demonstrates high consistency between the training and test phases, with a minimal training–test performance gap of only 0.0002, confirming its robustness and generalizability in solving the facility siting problem.

5.2 Results of the comparative experiments

To comprehensively evaluate the performance of the proposed SPID method, it is systematically compared with two baseline methods: traditional clustering- and GNN-based methods. Both approaches are mainstream solutions for location optimization problems and provide a scientific benchmark for assessing the proposed DRL-based framework.

For the GNN-based methods, each model is trained for 50 epochs using the same training instances as SPID. All GNN models use 128-dimensional hidden representations to match the embedding dimensions of SPID. Additionally, they are optimized using the Adam optimizer with the same loss function to ensure a fair comparison.

The problem scenarios for the comparison experiments are set as $p=5$, $C_j=60$, $r=0.3$, and $\sum d_i=550$, where candidate locations and demands follow a uniform random distribution within the unit square $[0, 1]^2$. Further, 1000 problem instances are generated using these scenario parameters to ensure comprehensive test. Table 4 presents the performances of all nine methods (SPID, five clustering-based methods, and three GNN-based methods) evaluated on the 1000 test instances. Furthermore, Fig. 5 shows a radar chart of

Table 4 Comprehensive comparison of the performances of SPID and the baseline methods

Method	S_g (%)	S_e (%)	S_f (%)	C_w	C_u	Inference time (ms/instance)	Training time (h)
SPID (ours)	**95.02**	**57.10**	**98.25**	35.45	7.17	28.5	1.38
GraphSAGE	70.16	51.02	91.45	**32.96**	6.56	15.2	1.29
GAT	70.64	51.11	90.60	33.95	6.59	12.8	1.41
GCN	70.06	50.86	91.52	34.07	**6.53**	**7.1**	0.95
Spectral	73.89	54.58	92.45	36.21	7.32	42.3	–
Hierarchical	70.82	54.32	92.01	36.15	7.29	38.6	–
k-means	69.34	53.86	91.23	36.08	7.28	8.9	–
k-medians	68.97	53.67	90.98	36.12	7.31	9.2	–
DBSCAN	52.18	38.42	72.15	45.67	9.12	156.7	–

The bold indicates the best results. "–" indicates that these methods do not involve the training process

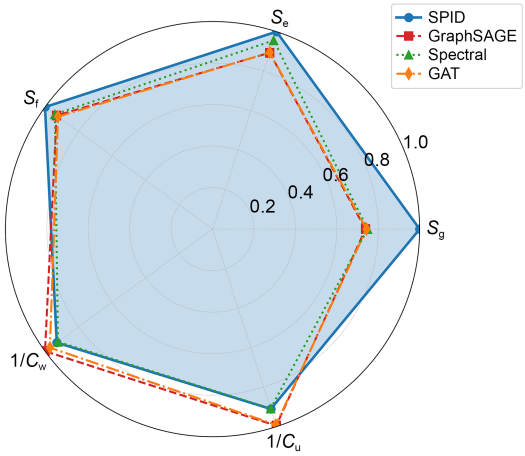

Fig. 5 Multi-dimensional performance radar chart

multi-dimensional performance metrics for comparing SPID with GNN-based algorithms.

Notably, the SPID method achieves the best performance on social indicators, S_g, S_e, and S_f. Across the social performance metrics emphasized in this study, SPID outperforms the suboptimal solutions produced by traditional clustering- and GNN-based methods by up to approximately 29%. In terms of distance cost control (the C_u indicator), the GCN-based method attains the lowest cost, while SPID exhibits a 9.8% increase compared to GCN, which is still within this acceptable margin. The GRU and attention mechanisms in SPID increase the computational overhead, making the inference time approximately 4 and 3.2 times those of GCN and *k*-means, respectively. For applications requiring real-time dynamic adjustments, such as the rapid deployment of temporary facilities in disaster response (or emergency logistics), this computational delay could represent a practical limitation.

The multi-dimensional radar chart further highlights the comprehensive advantages of SPID in solving facility location problems. The chart reveals that SPID forms a significantly expanded polygon in the S_g, S_e, and S_f dimensions, with facility utilization approaching an ideal state. This result indicates that SPID achieves broad coverage and ensures efficient resource allocation, thereby avoiding the dilemma of high coverage but wasted capacity or high utilization but insufficient coverage. Conversely, although GraphSAGE demonstrates competitiveness in distance indicators ($1/C_w$ and $1/C_u$), the significant contraction of its coverage dimensions exposes the flaw of over-optimizing operational costs at the expense of service equity. This difference is particularly crucial in public facility planning: for essential services, such as hospitals and schools, service accessibility must be prioritized over mere cost minimization. From the perspective of long-term return on investment, the adaptability and stability of the location solutions of SPID offer enhanced robustness against uncertainties, including population distribution changes and demand fluctuations, thereby simultaneously reducing the risk of future replanning and promoting social equity.

As a public infrastructure, UAM vertiport sites are evaluated based on two core indicators: coverage and effective coverage, reflecting the scientific quality of their selected locations. Fig. 6 shows that SPID achieves the best siting results, significantly outperforming the other models in terms of coverage rates. Under the same facility number and capacity constraints, SPID serves almost all candidate locations, with its S_g improving by almost 20 percentage points compared with the second-best model (Spectral). The

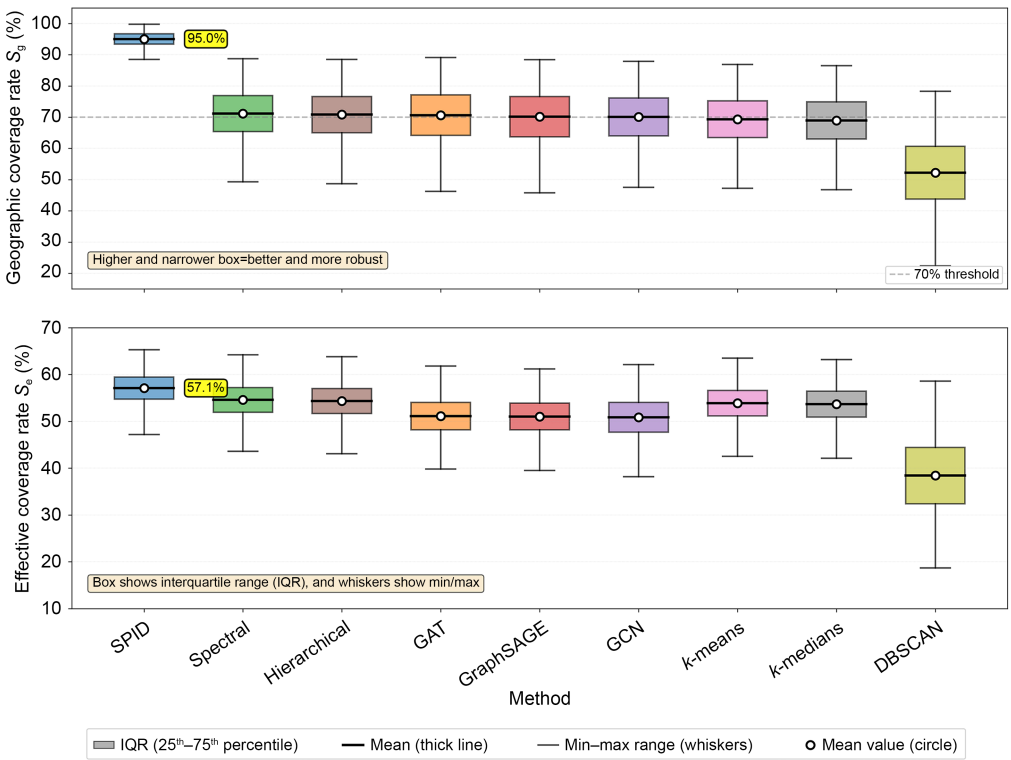

Fig. 6 Comparison chart of the geographic coverage rate S_g and the effective coverage rate S_e of the compared methods across 1000 test instances

lower service blind spot ratio of the SPID solution aligns it more with the required service attributes for public facilities. Regarding S_e, SPID maintains its outstanding leading position, although the margin of advantage over other methods has narrowed.

The simultaneous achievement of high S_g and S_e confirms that the sequential decision-making mechanism of SPID can achieve an optimal balance between spatial layout and capacity allocation. This capability is especially valuable in the face of future dynamic demand fluctuations. For instance, when population growth in a certain area causes a surge in demand, the initial location selection would have already covered that area (with a high S_g), and only local capacity adjustments will be required. This coverage avoids the need for complete location re-selection. This renders the location results of SPID more strategically forward-looking and better suited to urban infrastructure development requirements.

Additionally, the narrow box plot combined with the high-median distribution pattern of SPID indicates its excellent performance and reflects the high stability of its solution quality for problems with diverse characteristics. Conversely, the second-best model, Spectral, exhibits high sensitivity to the characteristics of the problem instances, with its interquartile range being four times that of SPID, reaching approximately 40 percentage points. To further quantify the robustness of each model, the coefficient of variation (CV=σ/μ) is employed as a standardized volatility indicator. Notably, S_g CV of SPID is approximately 0.021, which is significantly lower than those of Spectral (0.054), GraphSAGE (0.071), and DBSCAN (0.137). These results further confirm the strong robustness of SPID against input data perturbations and problem instance heterogeneity. This robustness indicates that the SPID model is better equipped to handle demand forecast errors, spatial data noise, and dynamic environmental changes.

Overall, SPID achieves an optimal trade-off between unconstrained spatial coverage and constrained capacity distribution. Moreover, the reliability and long-term value of its solutions are evident in the theoretical research findings (e.g., robust convergence and optimal balance of trade-off) and their potential for practical deployment across various operational scenarios.

5.3 Experiments across development stages

5.3.1 Experiment configuration

Low-altitude aviation is currently at its developmental stage. At this stage, the operational capacity of takeoff and landing platforms is generally sufficient to meet existing demands. However, with the evolution of the sector, the supply–demand balance is expected to shift (from an initial oversupply state to relative scarcity, and eventually to undersupply). To capture this progression, three simulation scenarios are designed to simulate the early, intermediate, and rapid growth stages of this aviation sector. Each scenario explores the facility location problem for low-altitude flight services under varying demand conditions.

For theoretical validation, these scenarios are used to assess the performance of the model under different constraint configurations: (1) flight radius constraints only, (2) capacity constraints only, and (3) both constraints applied simultaneously. All simulations in this section are performed using SPID and Gurobi. Gurobi is widely recognized in operations research and industry as a leading commercial mathematical programming solver, particularly for mixed-integer optimization problems. Therefore, it provides a reliable reference solution for evaluating the accuracy and quality of the outputs of SPID. Table 5 presents a summary of the experimental design and corresponding analyses of the various developmental stages.

The experimental environment uses the following configuration: operating system (Windows 11.0 (22631.2) running on a 64-bit Windows platform), CPU model (12th Gen Intel® Core™ i9-12900H), and solver (Gurobi Optimizer version 11.0.3 build v11.0.3rc0).

Table 5 Summary of the experimental settings

Development stage	Supply–demand ratio	Key constraint
Early stage	Sufficiency	Flight radius
Intermediate stage	Tight	Both
Rapid growth stage	Shortage	Capacity

The Python programming language is used for all coding and implementation. The training loss plots for the three stages are illustrated in the supplementary materials.

5.3.2 Experiment of the early stage

Here, the facility location problem during the early stage of low-altitude aviation is examined (the facility capacity is sufficient at this stage). Additionally, key constraints at this stage include the spatial distribution of demand points and UAV flight limitations. As this scenario aligns directly with the current practical needs, it is used to assess the effectiveness of SPID in facility location optimization. For the assessment, comparative experiments are conducted across three distinct problem sizes (20, 50, and 100 nodes), with SPID benchmarked against the exact mathematical programming solver Gurobi.

All algorithms produced feasible solutions within the acceptable time. Fig. 7 shows the node configurations, and Table 6 presents the experimental results. For the 100-node case, 50 additional experiments with varying capacities and demand distributions are run. Fig. 8 shows the selected results and summary charts.

The experimental results highlight three key advantages of SPID in facility location optimization: (1) stable near-optimal performance across diverse

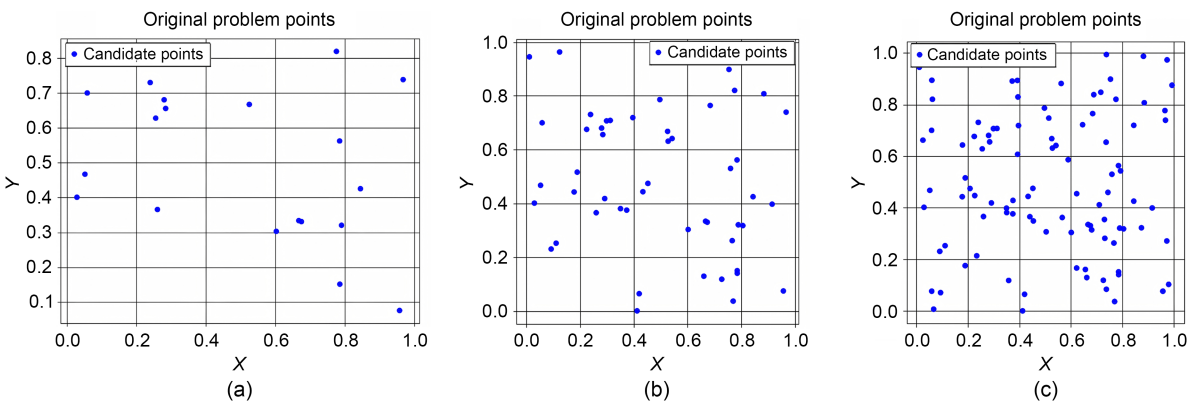

Fig. 7 Schematic of the experimental node settings: (a) 20 nodes; (b) 50 nodes; (c) 100 nodes

Table 6 Experimental results

Scenario setting	Algorithm	Objective function value	Solving time (s)	Gap
20 nodes Facility points: 3 Total demand: 113.92 Individual facility capacity: 60	SPID	3.2257		−0.002
	Gurobi	3.2237	0.019	
50 nodes Facility points: 4 Total demand: 279.98 Individual facility capacity: 100	SPID	7.8333		0
	Gurobi	7.8333	0.101	
100 nodes Facility points: 5 Total demand: 554.97 Individual facility capacity: 120	SPID	16.0871		−0.0861
	Gurobi	16.0010	2.64	

problem sizes; (2) strong multi-objective balancing, particularly in coverage and resource utilization; (3) superior computational efficiency compared with exact algorithms, making it more suitable for large-scale applications.

Although slightly less effective than Gurobi in minimizing transportation cost, SPID offers better overall performance in terms of service quality and runtime, thereby enhancing its practical value. Ranging between 20 and 100 nodes, SPID consistently maintains solution gaps below 0.1. Gurobi achieves lower transportation cost (31.03 vs. 49.83, −37.73%) and shorter total distance (6.64 vs. 11.49, −42.21%). However, SPID excels in terms of demand coverage (50.62% vs. 48.36%) and facility utilization (93.63% vs. 89.46%), reflecting its capability to balance efficiency and service. SPID also demonstrates strong computational performance, producing solutions within 6 s. Further, Gurobi fails to produce solutions within 1 h without initialization. However, when the output from SPID is deployed as an initialization point, its solving time decreases to 0.71 s, demonstrating the quality of SPID-obtained solutions and the value of DRL-based acceleration.

To assess the stability of SPID-obtained solutions, multiple experiments were conducted alongside a correlation analysis (Fig. 8), and the results reveal that the key metrics exhibited strong correlations. Additionally, S_g and S_e exhibited high correlation (r_c=0.734), as did facility utilization and service efficiency (r_c=0.740).

Conversely, transportation cost correlated negatively with most indicators, particularly S_g (r_c=−0.341)

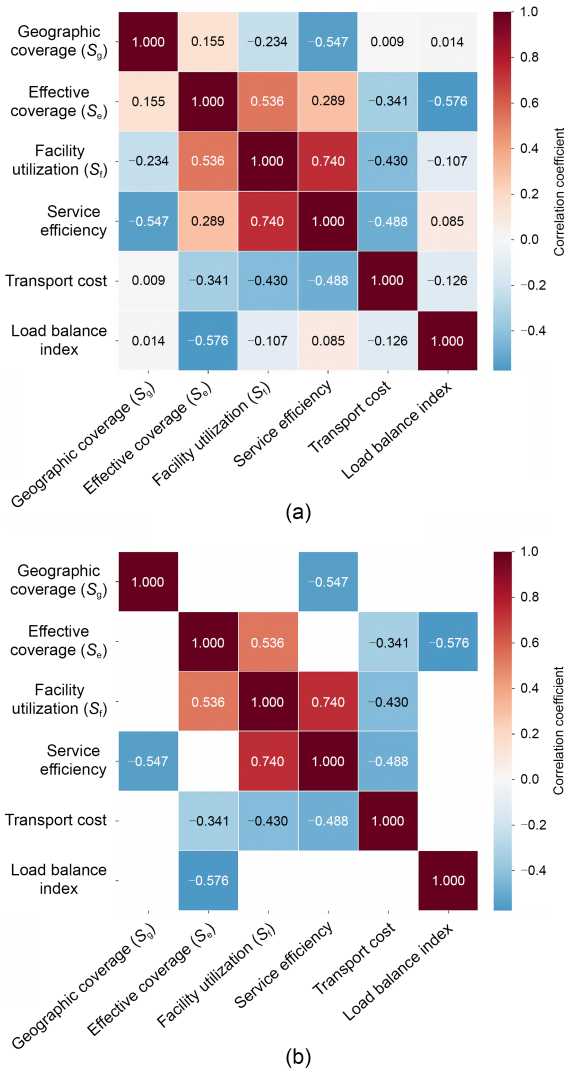

Fig. 8 Schematic of the experimental node settings: (a) complete correlation matrix; (b) strong correlations only with $|r_c|\geq 0.3$

and S_e ($r_c=-0.311$), indicating that higher costs reduce service reach. The load-balance index exhibited overall weaker correlations, with its most significant correlation being a negative relationship with service efficiency ($r_c=-0.276$). This indicated that uneven load distribution may hinder performance.

Further analysis of strong correlations ($|r_c|\geq 0.3$) confirmed two key patterns (Fig. 8b): (1) S_g and S_e are tightly linked, and (2) facility utilization is closely associated with service efficiency. These patterns provide theoretical support for optimizing service configuration and enhancing system performance.

5.3.3 Experiment of the intermediate stage

This subsection explores the facility location problem during the mid-stage development of low-altitude aviation. This stage is characterized by limited facility capacity, where capacity-related constraints begin to play a critical role in location decisions. To assess the effectiveness and solution performance of SPID in these scenarios, it is compared against Gurobi using a 100-node problem instance. The following experiment parameters were set: $p=5$, $C_j=60$, $r=0.3$, $\sum d_i=555.0$, and the results are shown in Fig. 9.

Experimental results reveal that SPID achieves a transportation cost of 49.83, a total distance of 11.49, a demand coverage rate of 50.62%, and a facility utilization rate (S_f) of 93.63%. In comparison, Gurobi achieves a lower transportation cost (31.03) and shorter distance (6.64), with a slightly lower demand coverage (48.36%) and S_f (89.46%).

SPID outperformed Gurobi in service-related metrics, with higher coverage (+2.26%) and utilization (+4.17%) scores. Conversely, Gurobi performed better in the cost (−37.73%) and distance (−42.21%) metrics. Overall, these findings reflect the capability of SPID to balance service quality and resource utilization, as well as the cost-efficiency advantage of Gurobi.

Regarding computational efficiency, SPID completed facility location optimization in approximately 6 min. Gurobi, without an initial solution, failed to converge within 1 h. However, when initialized with the output of SPID, it reached a global optimum in just 0.71 s (mixed-integer programming gap=0.0003), confirming the high quality of SPID-obtained solutions and their value as accelerators of exact solvers.

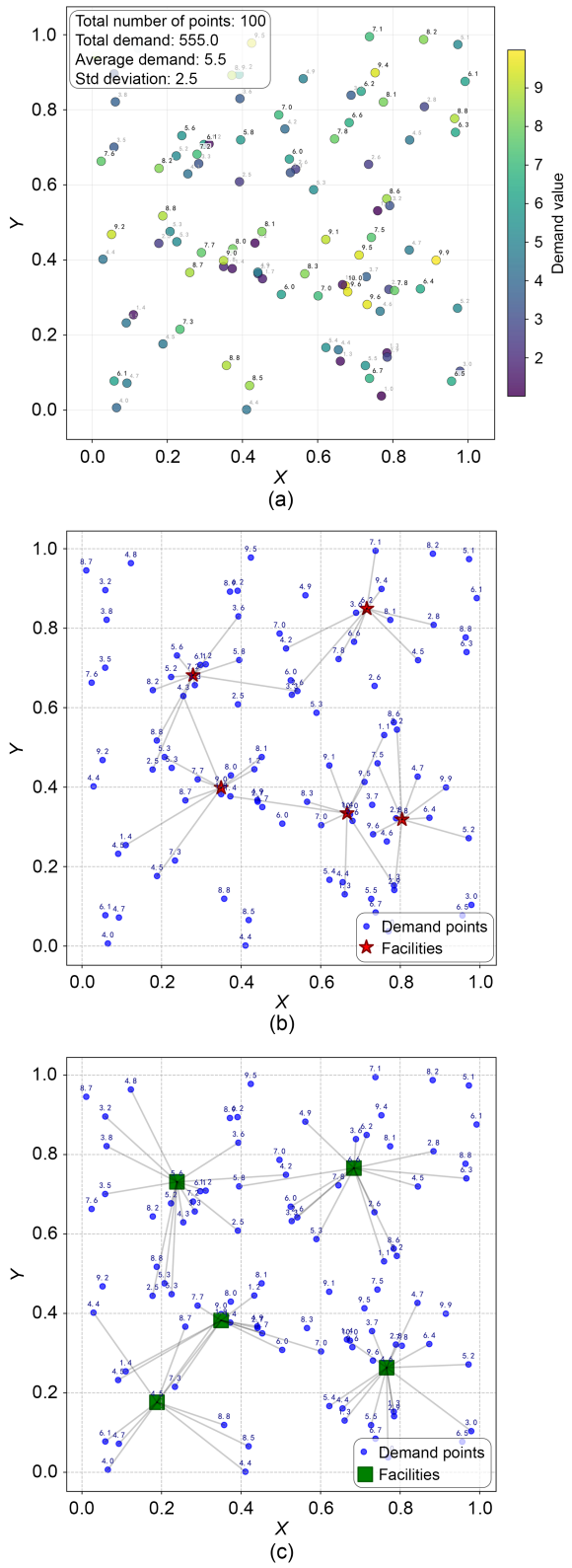

Fig. 9 Intermediate stage simulation results: (a) the initial distribution of demand points; (b) Gurobi solution; (c) SPID solution

5.3.4 Experiment of the rapid growth stage

This subsection explores the facility location problem during the rapid growth phase of low-altitude aviation. At this stage, facility capacity becomes increasingly constrained, whereas the endurance of low-altitude aircraft is expected to improve significantly. Consequently, capacity-related constraints become the primary factors influencing location decisions. To assess the effectiveness of SPID in these scenarios, comparative experiments are conducted against the exact solver Gurobi on a 100-node problem instance. Table 7 presents a summary of the results. Notably, two experimental scenarios are designed for the comparison:

Scenario 1: p=5, C_j=60, r=1.2, and $\sum d_i$=550.0; the results are shown in Figs. 10c and 10d.

Scenario 2: p=5, C_j=60, r=1.2, and $\sum d_i$=596.6; the results are shown in Figs. 10e and 10f.

The experimental results reveal significant performance differences between SPID and Gurobi in the rapid growth phase scenarios. SPID yields higher objective functions and longer total distances, but achieved better S_e and S_f. It also provides more balanced demand allocation. For instance, Fig. 10c shows that SPID covered third-quadrant demand points that Gurobi failed to. This supports previous findings that higher demand satisfaction may increase the overall cost.

With the increasing r of the facility, overlapping service areas will make the allocation strategy more critical. Notably, under stringent capacity constraints, SPID exhibits limited capability to address multi-coverage conflicts. As shown in Fig. 11, when Gurobi re-optimizes the allocations based on SPID-facilitated facility placements, the objective functions decrease significantly (very close to the original results obtained by Gurobi). This finding indicates that SPID performs well in site selection but requires improvement in allocation refinement.

Consistent with this observation, the final Gurobi re-allocation experiment re-optimizes allocation with SPID-based siting plan, yielding a demand coverage rate of 49.22% (Fig. 11d). This exceeds the 47.57% (Fig. 11c) coverage obtained when Gurobi directly optimizes siting, corresponding to an improvement of approximately 3%. Taken together, these results indicate that SPID's performance gap arises primarily from limitations in its allocation strategy rather than inaccuracies in site selection, and that its more balanced and strategically sound spatial deployment (Fig. 11d) provides a stronger foundation for downstream allocation. Overall, the evidence supports the robustness of SPID within the siting allocation framework and highlights its forward-looking value for long-horizon planning under demand uncertainty.

6 Conclusions

A novel solution to the low-altitude UAM facility location problem is demonstrated by reformulating the classical CFLP (an inherently combinatorial optimization problem) as an MDP, which is subsequently solved efficiently via DRL-based solver, SPID. The proposed SPID model delivers consistently strong performance in siting decisions. Notable performance differences relative to the exact solver, Gurobi, only arise when allocation quality becomes critical (typically under scenarios with extremely wide facility coverage (i.e., high degrees of overlapping service areas)). In such cases, applying Gurobi to the re-optimization of demand allocation based on SPID-solved siting decisions reduces the distance-based cost and improves demand coverage, thereby further validating the robustness of the approach. Further, SPID offers notable advantages in computational efficiency. Although model training requires a modest upfront time investment, inference with a pretrained model solves new problems

Table 7 Experimental results

Scenario	Algorithm	Objective function	Total distance	S_e (%)	S_f (%)
1	SPID	145.87	5.53	46.78	86.54
	Gurobi	28.28	36.68	49.29	96.07
2	SPID	128.86	30.54	45.07	94.76
	Gurobi	152.94	25.71	47.57	99.20

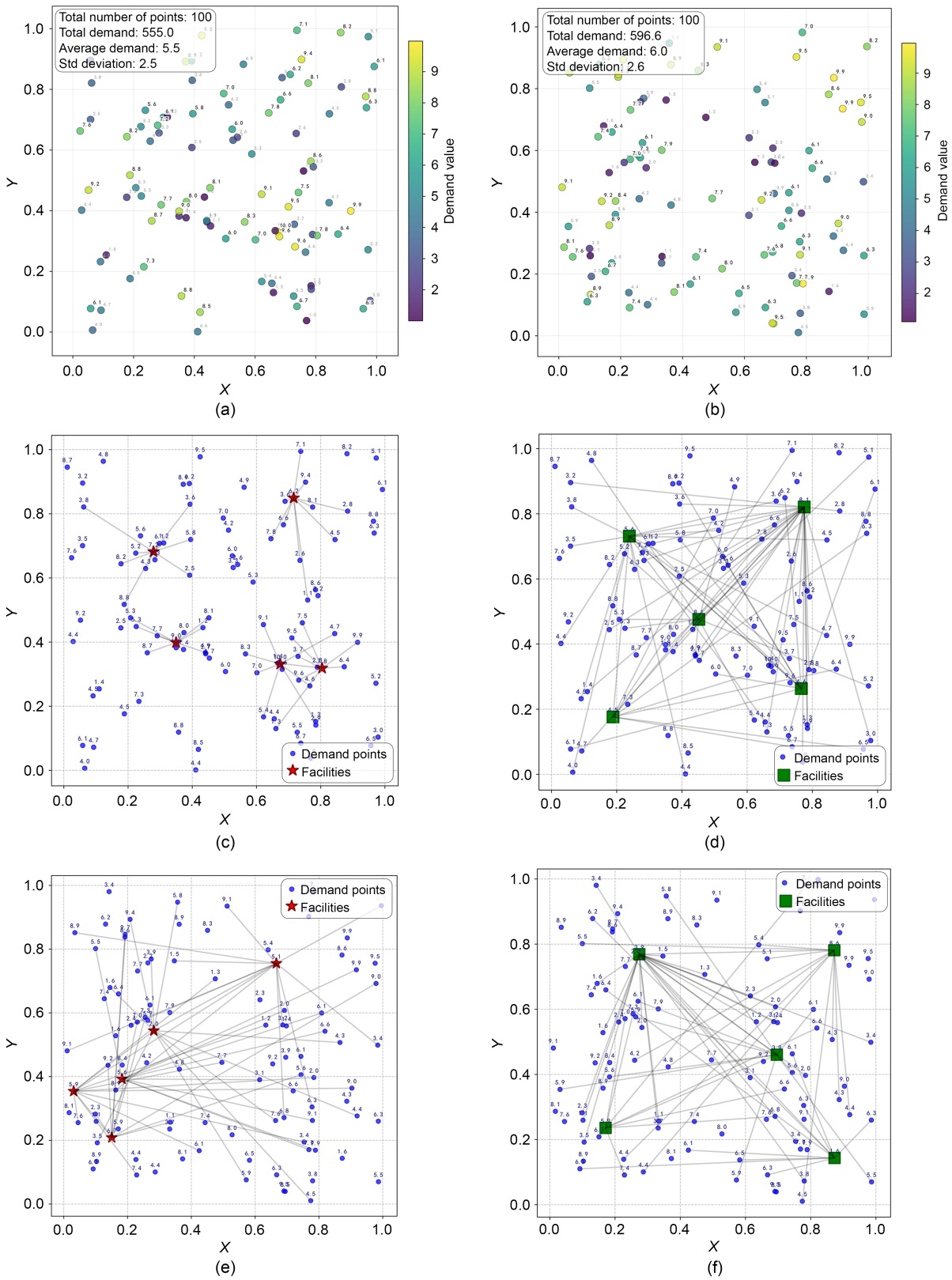

Fig. 10 Experimental results for the rapid growth phase: (a) the initial distribution of demand points for scenario 1; (b) the initial distribution of demand points for scenario 2; (c) Gurobi solution for scenario 1; (d) SPID solution for scenario 1; (e) Gurobi solution for scenario 2; (f) SPID solution for scenario 2

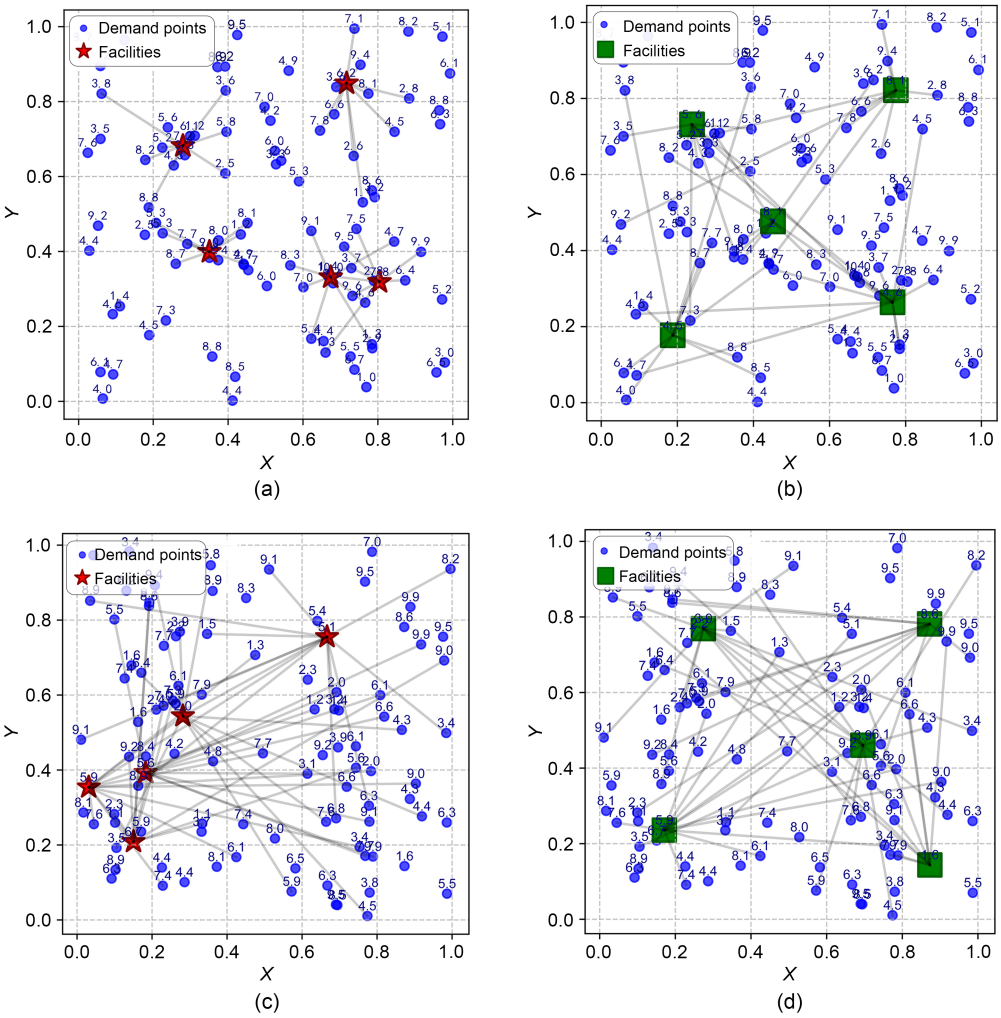

Fig. 11 Gurobi-resolved experimental results via post-redistribution solutions with different total transportation costs: (a) 28.48 in scenario 1; (b) 74.46 in scenario 1; (c) 152.94 in scenario 2; (d) 131.41 in scenario 2

within 6 min. This performance substantially outperforms Gurobi, which may fail to return a solution within 1 h, especially without a high-quality feasible starting point to narrow the initial search space. Overall, SPID delivers a reliable site selection performance under standard and challenging conditions, even without a dedicated allocation sub-model. Its emphasis on maximizing service coverage and promoting equitable distribution makes it particularly suitable for public service applications and regulatory contexts that prioritize accessibility over cost minimization. Future modifications may include integrating an allocation module to improve the model performance in complex allocation scenarios.

This study operates under several simplifying assumptions that are applied to the problem formulation and the solution setup. On the modeling side, the framework assumes a steady-state demand distribution, which is appropriate for long-term infrastructure planning but neglects short-term temporal fluctuations. Put differently, it does not explicitly capture exogenous demand fluctuations driven by time-of-day patterns, seasonal variation, or special events. The proposed formulation also omits several practical deployment constraints, including regulatory requirements (e.g., airspace restrictions and zoning limits), environmental impacts (e.g., noise and energy consumption), and three-dimensional airspace coordination (e.g., detailed flight path planning). These factors are indeed critical for real-world deployment, but can be effectively treated as additional constraint layers. Future studies may incorporate them by extending the state space of the MDP

or by formulating the problem as a multi-objective optimization task.

On the solution side, the main objective of this study is to assess the potential of DRL as a viable alternative to traditional operations research techniques for solving MILPs. Crucially, the goal is not to claim superiority over all possible DRL approaches. Thus, SPID is specifically compared against baselines that reflect established practice in facility location problems: exact solvers, heuristic algorithms, and GNN-based methods. In this study, we do not benchmark against more general-purpose DRL algorithms, such as proximal policy optimization or soft actor–critic, which are designed for broader classes of control and decision-making tasks. A systematic comparison with such algorithms remains a crucial direction for future studies in evaluating the broad applicability of DRL-based MILP solvers.

In summary, the SPID framework can be effectively enhanced along several dimensions. Potential extensions for future studies include modeling temporal demand dynamics to capture short-term fluctuations, integrating an explicit demand allocation module to improve performance under high service overlap, and integrating more realistic deployment constraints (e.g., regulatory and environmental factors). These additions would increase the applicability of the framework. Future studies must also include a systematic comparison with state-of-the-art DRL methods developed specifically for combinatorial optimization, which will better characterize performance limits and identify opportunities for architectural improvement. Thus, this study represents an initial step toward a comprehensive planning system for the UAM infrastructure rather than a complete deployment-ready solution.

Contributors

Xiaocheng LIU and Yupu LIU designed the research. Xiaocheng LIU processed the data and drafted the paper. Meilong LE and Yupu LIU helped organize the paper. Meilong LE revised the paper. Minghua HU revised and finalized the paper.

Conflict of interest

Minghua HU is a guest editor of the Special Feature on Engineering and Technology for Low-Altitude Economy Infrastructure of *Frontiers of Information Technology & Electronic Engineering*; he was not involved with the peer review process of this paper. All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Abdel-Basset M, Mohamed R, Hezam IM, et al., 2024. Multi-objective trajectory optimization algorithms for solving multi-UAV-assisted mobile edge computing problem. *J Cloud Computing*, 13:35. https://doi.org/10.1186/s13677-024-00594-z

Apaza RD, Li HX, Han RX, et al., 2023. Multi-agent deep reinforcement learning for spectrum and air traffic management in UAM with resource constraints. IEEE/AIAA 42nd Digital Avionics Systems Conf, p.1-7. https://doi.org/10.1109/DASC58513.2023.10311311

Berger R, 2023. Urban Air Mobility Market Update. https://www.rolandberger.com [Accessed on June 30, 2025].

Church R, ReVelle C, 1974. The maximal covering location problem. *Papers Region Sci*, 32:101-118. https://doi.org/10.1111/j.1435-5597.1974.tb00902.x

Deloitte, 2023. Advanced Air Mobility Industry Insight. https://www2.deloitte.com [Accessed on June 28, 2025].

Escribano Macias J, Khalife C, Slim J, et al., 2023. An integrated vertiport placement model considering vehicle sizing and queuing: a case study in London. *J Air Transp Manage*, 113:102486. https://doi.org/10.1016/j.jairtraman.2023.102486

Federal Aviation Administration, 2023. Aerospace Forecast Fiscal Years 2023–2043. https://www. faa. gov/data_research/aviation/aerospace_forecasts/faa-aerospace-forecast-fy-2023-2043 [Accessed on June 28, 2025].

Garcia CP, Li WG, Hirata NST, et al., 2023. ISUAM: intelligent and safe UAM with deep reinforcement learning. IEEE 29th Int Conf on Parallel and Distributed Systems, p.378-383. https://doi.org/10.1109/ICPADS60453.2023.00064

Gopi SP, Magarini M, 2021. Reinforcement learning aided UAV base station location optimization for rate maximization. *Electronics*, 10(23):2953. https://doi.org/10.3390/electronics10232953

Hakimi SL, 1965. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Oper Res*, 13(3):462-475. https://doi.org/10.1287/opre.13.3.462

Han RX, Li HX, Apaza R, et al., 2022. Deep reinforcement learning assisted spectrum management in cellular based urban air mobility. *IEEE Wirel Commun*, 29(6):14-21. https://doi.org/10.1109/MWC.001.2200150

Han XY, Xie MX, Yu K, et al., 2024. Combining graph neural network with deep reinforcement learning for resource allocation in computing force networks. *Front Inform Technol Electron Eng*, 25(5):701-712. https://doi.org/10.1631/FITEE.2300009

Hottung A, Tierney K, 2020. Neural large neighborhood search for the capacitated vehicle routing problem. 24th European Conf on Artificial Intelligence, p.443-450.

https://doi.org/10.3233/FAIA200124

Kim D, Lee K, Moon I, 2019. Stochastic facility location model for drones considering uncertain flight distance. *Ann Oper Res*, 283(1-2):1283-1302. https://doi.org/10.1007/s10479-018-3114-6

Kool W, Van Hoof H, Welling M, 2019. Attention, learn to solve routing problems! https://doi.org/10.48550/arXiv.1803.08475

Kumar PK, Witter J, Paul S, et al., 2023. Graph learning based decision support for multi-aircraft take-off and landing at urban air mobility vertiports. AIAA SciTech 2023 Forum. https://doi.org/10.2514/6.2023-1848

Lee S, Cho N, 2025. Optimal location of urban air mobility (UAM) vertiport using a three-stage geospatial analysis framework. *Fut Transp*, 5(2):58. https://doi.org/10.3390/futuretransp5020058

Li M, Zhao P, Bao ZA, et al., 2025. A reinforcement learning-based iterative method for capacitated hub location problems in UAV networks. Integrated Communications, Navigation and Surveillance Conf, p.1-11. https://doi.org/10.1109/ICNS65417.2025.10976790

Liang HJ, Wang SH, Li HL, et al., 2024. SpoNet: solve spatial optimization problem using deep reinforcement learning for urban spatial decision analysis. *Int J Digit Earth*, 17(1): 2299211. https://doi.org/10.1080/17538947.2023.2299211

Lynskey J, Thar K, Oo TZ, et al., 2019. Facility location problem approach for distributed drones. *Symmetry*, 11(1):118. https://doi.org/10.3390/sym11010118

Mahmoodi A, Sajadi SM, Sadeq AM, et al., 2025. Enhancing unmanned aerial vehicles logistics for dynamic delivery: a hybrid non-dominated sorting genetic algorithm II with Bayesian belief networks. *Ann Oper Res*, early access. https://doi.org/10.1007/s10479-025-06504-z

Meng ZY, Yu K, Qiu R, 2024. Location-routing optimization of UAV collaborative blood delivery vehicle distribution on complex roads. *Complex Intell Syst*, 10(6):8127-8141. https://doi.org/10.1007/s40747-024-01591-0

Meskar M, Ahmadi-Javid A, 2024. Optimizing drone delivery paths from shared bases: a location-routing problem with realistic energy constraints. *J Intell Robot Syst*, 110(4):142. https://doi.org/10.1007/s10846-024-02129-9

National Aeronautics and Space Administration, 2020. STEM LEARNING: Package Delivery Drone Simulation Coding Activity Guide. https://www.nasa.gov/wp-content/uploads/2020/05/aam-package-delivery-drone-simulation-activity-guide_0.pdf [Accessed on June 26, 2025].

Nazari M, Oroojlooy A, Takáč M, et al., 2018. Reinforcement learning for solving the vehicle routing problem. Proc 32nd Int Conf on Neural Information Processing Systems, p.9861-9871.

Park C, Kim GS, Park S, et al., 2023. Multi-agent reinforcement learning for cooperative air transportation services in city-wide autonomous urban air mobility. *IEEE Trans Intell Veh*, 8(8):4016-4030. https://doi.org/10.1109/TIV.2023.3283235

Paul A, Levin MW, Waller ST, et al., 2024. Data-driven optimization for drone delivery service planning with online demand. https://doi.org/10.48550/arXiv.2404.02442

Premkumar VGR, Van Scoy B, 2025. Optimal positioning of unmanned aerial vehicle (UAV) base stations using mixed-integer linear programming. *Drones*, 9(1):44. https://doi.org/10.3390/drones9010044

Rahman MA, Basheer MA, Khalid Z, et al., 2024. Logistics hub location optimization: a k-means and p-median model hybrid approach using road network distances. https://doi.org/10.48550/arXiv.2308.11038

Ryu K, Kim W, 2024. Energy efficient deployment of aerial base stations for mobile users in multi-hop UAV networks. *Ad Hoc Netw*, 157:103463. https://doi.org/10.1016/j.adhoc.2024.103463

Shavarani SM, Nejad MG, Rismanchian F, et al., 2018. Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon Prime Air in the city of San Francisco. *Int J Adv Manuf Technol*, 95(9):3141-3153. https://doi.org/10.1007/s00170-017-1363-1

Shavarani SM, Mosallaeipour S, Golabi M, et al., 2019. A congested capacitated multi-level fuzzy facility location problem: an efficient drone delivery system. *Comput Oper Res*, 108:57-68. https://doi.org/10.1016/j.cor.2019.04.001

Song KH, Lee H, 2025. Network topology-driven vertiport placement strategy: integrating urban air mobility with the Seoul Metropolitan railway system. *Appl Sci*, 15(7):3965. https://doi.org/10.3390/app15073965

Sun J, Shu SJ, Hu HD, et al., 2025. Location optimization of unmanned aerial vehicle (UAV) drone port for coastal zone management: the case of Guangdong coastal zone in China. *Ocean Coast Manag*, 262:107576. https://doi.org/10.1016/j.ocecoaman.2025.107576

Volakakis V, Mahmassani HS, 2024. Vertiport infrastructure location optimization for equitable access to urban air mobility. *Infrastructures*, 9(12):239. https://doi.org/10.3390/infrastructures9120239

Williams RJ, 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn*, 8(3-4):229-256. https://doi.org/10.1007/BF00992696

Zhang CX, Du WB, Guo T, et al., 2025. Multi-objective hub location for urban air mobility via self-adaptive evolutionary algorithm. *Adv Eng Inform*, 64:102974. https://doi.org/10.1016/j.aei.2024.102974

Zhao Y, Feng T, 2024. Strategic integration of vertiport planning in multimodal transportation for urban air mobility: a case study in Beijing, China. *J Clean Prod*, 467:142988. https://doi.org/10.1016/j.jclepro.2024.142988

Zheng L, Xu G, Chen WB, 2024. Using improved particle swarm optimization algorithm for location problem of drone logistics hub. *Comput Mater Con*, 78(1):935-957. https://doi.org/10.32604/cmc.2023.046006

Zhu TK, Boyles SD, Unnikrishnan A, 2022. Two-stage robust

facility location problem with drones. *Transp Res Part C Emerg Technol*, 137:103563.
https://doi.org/10.1016/j.trc.2022.103563

List of supplementary materials

1 Overview
2 Training procedure for the SPID algorithm
3 Model detailed architecture and hyperparameter configurations
4 Model training parameter figure
Algorithm S1 Training procedure for SPID
Table S1 SPID network architecture configuration
Table S2 SPID training hyperparameter configuration
Fig. S1 Training diagram of the model used in the multi-development stage experiment with different parameter values