

A Risk-aware Spatio-temporal Scheduler for Large-scale UAV Operations

Wenxuan Wang, Arnab Majumdar, Washington Ochieng, and Jose Escribano

Department of Civil and Environmental Engineering

Imperial College London

London, United Kingdom

Abstract—Spatio-temporal scheduling of Unmanned Aerial Vehicles (UAV) is a major challenge for the UAV Traffic Management (UTM) system due to dense traffic and high mobility. To ensure safety and efficiency, UAV schedulers assign both geographic paths and operational time windows to prevent conflicts and minimize delays. However, scheduling also influences UAV operational risks by determining the potential exposure of UAV failures to third parties on the ground. Since operational risk has not been systematically considered in existing schedulers, this work integrates systematic risk assessment into a hybrid spatio-temporal scheduling framework that combines greedy search with Mixed-Integer Linear Programming. The proposed scheduler is evaluated in large-scale simulations of 5,000, 10,000, and 15,000 flights over central London during 24 hours. Results show that the risk-aware solver successfully schedules 13.24%-30.70% more flights while reducing operational risk by 85.71%-96.67% compared to the benchmark. These improvements become more pronounced at larger scales, demonstrating the scalability and potential of the proposed approach for real-world UTM deployment.

Keywords—spatio-temporal UAV scheduling; UAV operational risk assessment; MILP; greedy heuristic

I. INTRODUCTION

The deployment of Unmanned Aerial Vehicles (UAVs) is expected to revolutionize domains such as logistics, emergency services, and urban mobility [1]. According to SESAR, the European drone market is projected to exceed €10 billion annually by 2035. To support this emerging demand, the SESAR Joint Undertaking proposed U-Space, a blueprint for the UAV Traffic Management (UTM) system [2]. In line with these expectations, thousands of UAV flights may operate daily over urban regions, reshaping the use of low-altitude airspace. While the UTM system promises many benefits, large-scale urban UAV operations still raise concerns about safety and efficiency.

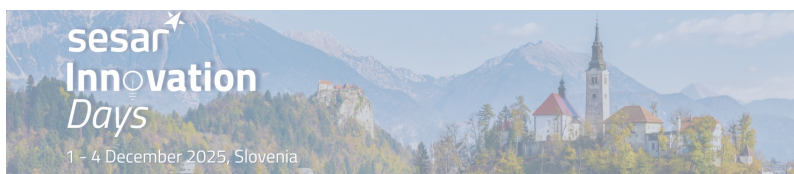
Given the higher densities, diverse mission profiles, and frequent interactions among UAVs, conflict resolution in the UTM system is a prerequisite for safety [3] and is typically classified into three stages: strategic, tactical, and escape [4]. Strategic resolution occurs during path planning before the flight begins. Tactical avoidance occurs shortly before departure and typically affects only segments of the planned path. Escape maneuvers are executed in real time, just seconds before a predicted conflict, using onboard detect-and-avoid systems. In the fourth edition of the U-Space Concept of operations, conflict resolution is defined as a strategic process,

requiring that all flight plans be free of conflicts [5]. This requirement implies that submitted flight requests, arriving at different times, must be checked and de-conflicted with existing plans before approval. Consequently, effective pre-flight scheduling is mandatory to ensure the safe and efficient use of low-altitude airspace [6]. In the UTM system, scheduling refers to the coordinating assignments of UAV flights to spatial paths and temporal slots (including the movement in each time step).

In recent years, several studies have researched the improvement of the UAV scheduling process. Most existing schedulers output the geographic assignment of flight paths, launch times, and operational windows [7], using methods ranging from reinforcement learning [8], pairwise negotiation [9], and heuristic search [10] to deterministic optimization techniques such as Mixed-Integer Linear Programming (MILP) [11] and dynamic programming [12], as well as hybrid combinations. Their common objectives are to maximize the coverage of events [13], minimize flight delays [14] (with conflict avoidance as a constraint), or minimize conflicts directly [15].

However, a major limitation of current schedulers is the lack of UAV operational risk assessment in scheduling. Most existing approaches account only for conflict-related risks, whereas UAV failures can also expose third parties on the ground to significant harm, with levels depending on multiple factors (introduced in Section II). Some recent studies extend the Specific Operational Risk Assessment (SORA) framework [16] to evaluate operational risks in a more structured way. A few studies, such as [17] and [18], have introduced risk considerations into path planning or scheduling. However, these typically rely on simplified models that cannot fully capture the risks to people on the ground. More comprehensive approaches, such as [19] and especially [20], propose systematic operational risk models; however, these have so far been applied only to single-agent pathfinding [21]. Integrating such risk-aware approaches into large-scale UAV schedulers, therefore, remains an open challenge.

Therefore, this paper proposes a risk-based spatio-temporal UAV scheduler built on a hybrid framework. Designed for large-scale UAV operations in urban environments, this scheduler jointly minimizes makespan (delays) and operational risk, with penalties applied to unscheduled flights. The main contributions are as follows: 1) A stochastic event generation model is developed based on prior demand prediction results for the



study area to generate realistic flight orientation-destination (OD) pairs for simulation and scheduler implementation. 2) A hybrid scheduler is constructed using heuristic preprocessing and MILP optimization, achieving a balance between computational efficiency and global optimality. 3) A risk model is integrated into the framework to ensure that scheduled paths remain within acceptable risk levels.

The remainder of the paper is organized as follows. Section II introduces the systematic risk model for urban UAV operations. Section III describes the scheduling problem. Section IV presents the proposed hybrid MILP-heuristic approach. Section V reports and discusses the simulation results. Finally, Section VI concludes the paper.

II. UAV OPERATIONAL RISK MODEL

This section provides a brief recap of the systematic UAV risk model adopted in the proposed scheduler. The full development of this risk assessment model is detailed in [20]. The core principle is to quantify the lethal risks posed by UAV failures to people on the ground, expressed as the product of failure likelihood L and severity S .

As shown in Equation 1, the ground risk likelihood $L(g, f)$ for a grid g from flight f depends on the UAV failure probability λ_f , the probability p_g that f will crash on grid g , and the population density ρ_g within that grid.

$$L(g, f) = p_g \rho_g \lambda_f \quad (1)$$

The severity modeling begins by determining the UAV's impact energy E_{imp} , which depends on the crash location relative to the failure point. The severity of a flight f over grid g is expressed as Equation 2, where f_r and f_b are functions that map impact energy (and, for f_b , the dimension D) to expected lethality. The severity also varies by the grid type g - road (x_r , vehicles), pavement (x_p , pedestrians), or building (x_b) - as these environments provide different levels of protection. The variables x_p , x_r , and x_b are binary indicators, with exactly one equal to 1 for any given grid cell.

$$S(g, f) = x_p + x_r f_r(E_{imp}) + x_b f_b(D, E_{imp}) \quad (2)$$

For pavements, any direct UAV impact on pedestrians is assumed to result in lethal injuries. Within the road and building categories, the lethality mapping functions f_r and f_b depend on more specific factors.

For vehicles within the impacted road segment, f_r is additionally influenced by parameters such as the vehicle's weight, characteristic dimensions, maximum energy absorption capacity, and number of occupants.

For buildings, f_b varies according to the loading mode (point, distributed, or helipad loading), impact location (roof or wall), and structural characteristics (e.g., standard coating versus glass panels).

This model enables real-time assessment of operational risk in each grid cell along a UAV's path. Minimizing the risk is, therefore, one of the scheduler's objectives, alongside minimizing makespan and the number of unscheduled flights.

III. PROBLEM DESCRIPTION

In this paper, the spatio-temporal UAV scheduling problem is formulated as the assignment of a feasible path and launch time to each flight. This involves optimizing three interdependent decisions: path planning, path selection, and scheduling. These decisions are modeled within a mathematical representation of the airspace and UAV operations.

This study focuses on the low-altitude airspace over central London. Given the fragmented structure of this airspace due to obstacles, the study area is discretized into a uniform grid of square cells, which serve as the basic units for flight paths and risk evaluation. Each cell is assigned a risk level computed using external data from OpenStreetMap, including ground factors such as population and building density, in accordance with the UAV operational risk model. Cells with the highest risk are designated as no-fly zones, while the rest are classified into ordered risk levels, from high to low, and may be used for UAV operations.

Time is discretized into uniform one-second steps across the scheduling horizon. Operational priorities are used to distinguish the relative importance of different service categories. UAVs may only move between adjacent grid cells, including diagonals, with travel distances determined by the chosen direction.

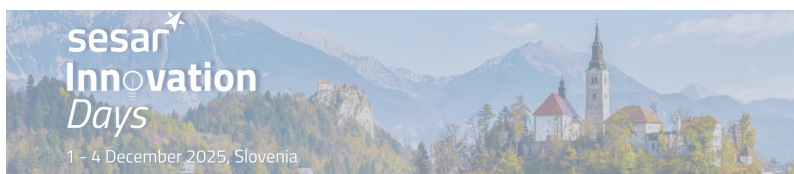
The basic operational assumptions applied in the environmental model are as follows:

1. Only uncrewed operations are considered, consistent with the U-space concept; path changes due to human intervention are excluded.
2. UAVs operate at a fixed en-route altitude of 400 ft above ground level, enabling a two-dimensional airspace model consistent with the corridor-based concept and reducing computational complexity.
3. Conflict detection is based on cell-time occupancy: simultaneous occupancy of one cell or head-on edge swaps is prohibited.
4. Hovering prior to launch is permitted, whereas en-route hovering is not allowed.

Despite these simplifying assumptions, the spatio-temporal UAV scheduling problem remains computationally challenging. It requires the joint determination of multiple variables under strict constraints. The challenge grows on a larger scale, motivating the need for scalable approaches.

IV. METHODOLOGY

The architecture of the proposed spatio-temporal UAV scheduler is shown in Figure 1. This hybrid heuristic-MILP framework consists of five main components: (1) a risk map modeling the low-altitude operational environment through computed risk levels of grid cells; (2) a greedy preprocessor generating risk-compliant shortest paths with time windows for as many flights as possible; (3) an MILP-based core scheduler optimizing predefined candidate paths and scheduling the flights left unscheduled by the preprocessor; (4) an iterative constraint generation incrementally introducing conflict-avoidance constraints to the MILP scheduler; and (5) a re-



path detouring flights that cannot simultaneously satisfy risk and delay constraints within the hybrid scheduler.

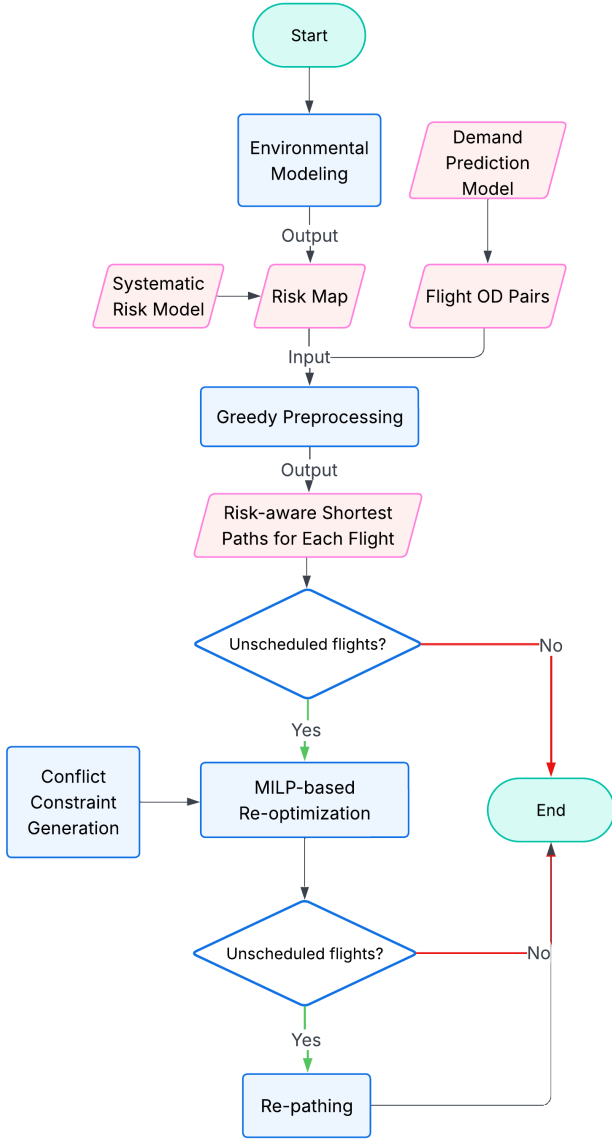


Figure 1. Architecture of the proposed scheduler.

In this framework, MILP provides an exact formulation of the UAV scheduling problem, allowing operational constraints to be explicitly modeled. However, its NP-hard nature makes solving large-scale instances computationally prohibitive. A heuristic component is, therefore, employed to accelerate the initial search process, generating fast approximations and pruning the solution space. Also, iterative MILP constraint generation ensures optimality within the reduced domain. This hybrid approach balances scalability and solution quality, enabling conflict-free scheduling of tens of thousands of UAVs within an acceptable computation time.

A. Greedy Preprocessing

Greedy algorithms are widely used in air traffic scheduling, where scalability is critical, as they provide feasible—though not necessarily optimal—solutions with minimal computational overhead. In this framework, a greedy scheduling heuristic is applied in the pre-scheduling stage to maximize the number of flights accommodated within the time horizon, thus reducing the core scheduler’s workload to optimizing the pre-scheduled paths and resolving a smaller set of difficult cases.

The objective function is given in Equation 3, where $x_f \in \{0, 1\}$ is a binary decision variable indicating whether flight f is scheduled ($x_f = 1$) or not ($x_f = 0$). This function maximizes the total number of scheduled flights f , weighted by their priorities w_f .

$$\max \sum_{f \in \mathcal{F}} w_f x_f \quad (3)$$

The constraints of the preprocessing model are summarized in Equations 4-7. Equation 4 enforces a risk constraint. For each flight f , the total risk R_f is computed as the sum of per-cell contributions along its path P_f , as defined by the UAV operational risk model. R_f must not exceed the maximum allowable threshold $R_{\max}$ (set to 0.1 in this work).

$$\sum_{g \in P_f} L(g, f) S(g, f) \leq R_{\max}, \quad \forall f \in \mathcal{F} \quad (4)$$

Equation 5 ensures conflict avoidance. Let h_f be the hop time per cell for flight f , and t_f its launch time. If P_f traverses the k -th cell g , the occupancy interval $I_{f,k}$ is $[t_f + kh_f, t_f + (k+1)h_f]$. Two flights f_1 and f_2 are in conflict if they attempt to occupy the same cell g simultaneously. Thus, their intervals must remain disjoint.

$$I_{f_1, k_1} \cap I_{f_2, k_2} = \emptyset, \quad \forall f_1 \neq f_2, \forall g \in P_{f_1} \cap P_{f_2} \quad (5)$$

Equation 6 imposes an endurance constraint. The total operation time of each flight f , defined as the sum of hop durations $h_{f,g}$ along its path P_f , must not exceed the maximum allowable endurance $T_f^{\max}$ (set to 600 seconds in this study) due to battery limitations.

$$\sum_{g \in P_f} h_{f,g} \leq T_f^{\max}, \quad \forall f \in \mathcal{F} \quad (6)$$

Finally, Equation 7 restricts launch flexibility. The actual launch time t_f is the planned release time r_f plus a non-negative delay d_f . The delay must remain within the maximum allowable holding window $\Delta_{\max}$.

$$0 \leq d_f \leq \Delta_{\max}, \quad t_f = r_f + d_f, \quad \forall f \in \mathcal{F} \quad (7)$$

Algorithm 1 presents the pseudocode for the greedy preprocessing mechanism. The preprocessor sorts flights primarily by descending priority and secondarily by release time, ensuring that high-priority flights are always considered first. This

ordering prevents lower-priority flights from blocking higher-priority ones during sequential scheduling.

Conflict detection operates at both the node and edge levels. At the node level, a cell is marked as occupied throughout its occupancy interval. At the edge level, constraints prevent the simultaneous occupancy of adjacent cells as well as head-on swaps between two flights traversing the same edge. If a conflict-free slot is found within the allowed slack—defined as the maximum permitted launch holding beyond the planned release—then the flight is scheduled, and its occupied cells and edges are permanently recorded in the global reservation table. Otherwise, the flight is marked as unscheduled and passed to the MILP-based core scheduler for subsequent optimization.

Algorithm 1 Greedy Risk-Constrained Preprocessing

```

PREPROCESS( $\mathcal{F}$ )
  Input:  $\mathcal{F}$  with  $r_f, w_f, R_{\max}, T_{\max}^f, \Delta_{\max}$ 
  Output: Scheduled flights  $\mathcal{S}$  with candidate paths  $\mathcal{P}$ ,
  unscheduled flights  $\mathcal{U}$ 
   $\mathcal{S} \leftarrow \emptyset, \mathcal{P} \leftarrow \emptyset, \mathcal{U} \leftarrow \emptyset$ 
   $\mathcal{F} \leftarrow \text{SORT}(\mathcal{F}, \text{keys} = (r_f, -w_f))$ 
  for each flight  $f \in \mathcal{F}$  do
    success  $\leftarrow$  False
    for  $d_f = 0$  to  $\Delta_{\max}$  do
       $t_f \leftarrow r_f + d_f$ 
       $P_f \leftarrow \text{SCHEDULE}(f, t_f)$ 
       $R_f \leftarrow \sum_{(x,y) \in P_f} R(x, y)$ 
      if  $R_f \leq R_{\max}$  then
        conflict  $\leftarrow \text{CHECKCONFLICT}(P_f, \mathcal{P})$ 
        if conflict = False then
           $\mathcal{S} \leftarrow \mathcal{S} \cup \{f\}$ 
           $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_f\}$ 
          success  $\leftarrow$  True
          break
    end for
    if success = False then
       $\mathcal{U} \leftarrow \mathcal{U} \cup \{f\}$ 
  end for
  return  $\mathcal{S}, \mathcal{P}, \mathcal{U}$ 

```

B. MILP Core Scheduler with Iterative Constraint Generation

The core scheduling stage is formulated as a MILP because it primarily involves decisions on when each UAV flies, which path it follows, and within which time window—choices that are inherently discrete. MILP is well suited to such problems, as it can directly handle binary and integer decision variables. Unlike the greedy preprocessor, which emphasizes fast, feasible solutions, the MILP stage focuses on finding a global optimum. Accordingly, the preprocessed flights and their candidate paths from the previous stage serve as inputs to the core scheduler. The MILP model then initializes the following decision variables based on these preprocessing outputs: $t_f \in \mathbb{R}^+$, the actual start time of flight f ; $e_f \in \mathbb{R}^+$, the end time of f ; $u_f \in \{0, 1\}$, the scheduling indicator ($u_f = 1$ if f is scheduled in the MILP model, 0 otherwise);

and $z_{f,g,k} \in \{0, 1\}$, the conflict-ordering variable indicating the precedence of f in grid cell g at step k .

As expressed in the objective function (Equation 8), the scheduler minimizes the makespan defined as the completion time e_f of the last flight (Equation 9) while also penalizing unscheduled flights. The weights α and β balance efficiency and coverage: a larger α emphasizes reducing makespan, while a larger β prioritizes accommodating more flights. In this model, α and β are both set to 0.5 with the same prioritization.

$$\min [\alpha \cdot \text{makespan} + \beta \sum_f (1 - u_f)] \quad (8)$$

$$\text{makespan} = \max_f (e_f), \quad \forall f \in \mathcal{F} \quad (9)$$

Although both stages address similar operational constraints, the greedy preprocessor enforces feasibility by sequentially checking each flight against existing schedules and assigning slots until no further flights can be accommodated. In contrast, the MILP stage encodes all constraints as global algebraic relationships applied simultaneously across all flights. This ensures consistent handling of timing, conflicts, and scheduling flexibility within a unified optimization framework.

The end time e_f of each flight f is defined as its start time plus duration (Equation 10), while the start time t_f must not occur earlier than the expected release time r_f (Equation 11).

$$t_f = e_f - d_f, \quad \forall f \in \mathcal{F} \quad (10)$$

$$t_f \geq r_f, \quad \forall f \in \mathcal{F} \quad (11)$$

For spatio-temporal consistency, if P_f is the ordered sequence of cells traversed by f , then the entry and exit times into the k -th cell are determined by the start time t_f and hop duration h_f (Equation 12).

$$t_{f,k}^{\text{in}} = t_f + kh_f, \quad t_{f,k}^{\text{out}} = t_f + (k+1)h_f \quad (12)$$

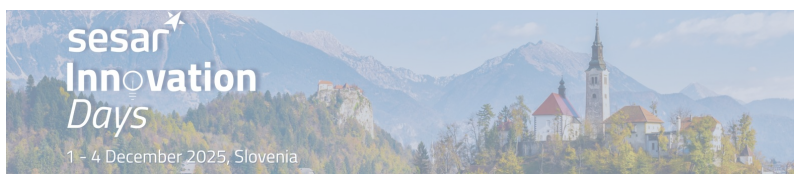
Conflict avoidance is guaranteed by the following: if two flights f_1 and f_2 attempt to occupy the same cell, one must exit before the other enters. This is enforced through big- $\mathcal{M}$ inequalities using binary ordering variables $b_{f_1, f_2, k_1, k_2} \in \{0, 1\}$ (Equations 13–14). Here, $\mathcal{M}$ is a sufficiently large constant (set to 10^5 in this study) used to relax precedence constraints when the corresponding binary variable is inactive.

$$t_{f_1, k_1}^{\text{out}} \leq t_{f_2, k_2}^{\text{in}} + \mathcal{M}(1 - b_{f_1, f_2, k_1, k_2}) \quad (13)$$

$$t_{f_2, k_2}^{\text{out}} \leq t_{f_1, k_1}^{\text{in}} + \mathcal{M}b_{f_1, f_2, k_1, k_2} \quad (14)$$

Scheduling flexibility is enabled by allowing actual start times t_f to shift around the planned release time r_f . This is controlled by the binary variable u_f , which activates slack in the release-time constraint when a flight is unscheduled (Equation 15 and 16).

$$t_f \geq r_f - \mathcal{M}(1 - u_f), \quad \forall f \in \mathcal{F} \quad (15)$$



$$e_f \leq r_f + \mathcal{M}u_f, \quad \forall f \in \mathcal{F} \quad (16)$$

Algorithm 2 outlines the core scheduling mechanism, where we employ an iterative constraint generation (i.e., lazy constraint) approach to efficiently handle the large number of separation constraints in the scheduling stage. Instead of enumerating all pairwise conflict constraints upfront—leading to excessively large and often intractable MILP formulations—the scheduler begins with only the core temporal constraints. After solving this reduced model, conflicts in the provisional schedule are detected, and only the corresponding separation constraints (Equations 13–14) are added iteratively. Each conflict batch is limited to 5,000. The process repeats until no conflicts remain or the solution converges. This strategy improves both scalability and solution quality in large-scale scheduling with limited resources.

Algorithm 2 MILP-based Core UAV Scheduler with Iterative Constraint Generation

MILPSCHEDULE($\mathcal{S}, \mathcal{P}, \mathcal{U}$)

Input: $\mathcal{S}, \mathcal{P}, \mathcal{U}$

Output: Scheduled flights $\mathcal{S}'$ with paths $\mathcal{P}'$, unscheduled flights $\mathcal{U}'$

$\mathcal{S} \leftarrow \emptyset, \mathcal{U} \leftarrow \emptyset$

Initialize $\mathcal{M}$ with variables t_f, e_f, u_f, b_{ijck}

$\mathcal{M}.$ ADDCONSTRAINTS(Equations 9-12,15)

$\mathcal{M}.$ SETOBJECTIVE(Equation 8)

repeat

Solve $\mathcal{M} \rightarrow \{t_f, e_f, p_f \in \mathcal{P}\}$

$\mathcal{C} \leftarrow \text{DETECTCONFLICTS}(\mathcal{F}, \mathcal{P}, \mathcal{S})$

for each $(i, j, c, k_i, k_j) \in \mathcal{C}$ **do**

$\mathcal{M}.$ ADDITERATIVECONSTRAINTS(Equations

13-14)

end for

until $\mathcal{C} = \emptyset$ or objective stabilizes

$\mathcal{P}' \leftarrow \mathcal{M}$

$\mathcal{S}' \leftarrow \{f \in \mathcal{F} : u_f = 1\}$

$\mathcal{U}' \leftarrow \{f \in \mathcal{F} : u_f = 0\}$

return $\mathcal{S}', \mathcal{P}', \mathcal{U}'$

V. SIMULATION AND RESULTS

The simulations are designed to emulate large-scale UAV operations and evaluate the performance of the proposed scheduler. All experiments were conducted on a macOS system with an Apple M1 processor and 8 GB of memory.

A. Flight Event Generation

The study area is defined as central London, bounded by latitudes 51.47°-51.54° N and longitudes 0.20°-0.05° W. This rectangular region spans approximately 80 km², covering dense urban districts between Westminster and the City of London. The area is discretized into a uniform grid of 200 m × 200 m cells, resulting in a grid of approximately 54 × 41 cells. Building footprints, population density, and road

networks are extracted from OpenStreetMap and overlaid onto the grid to compute risk scores for each cell. The risk levels are illustrated in Figure 2. This risk-aware grid serves as the spatial domain for generating flight OD pairs and implementing risk-sensitive scheduling.

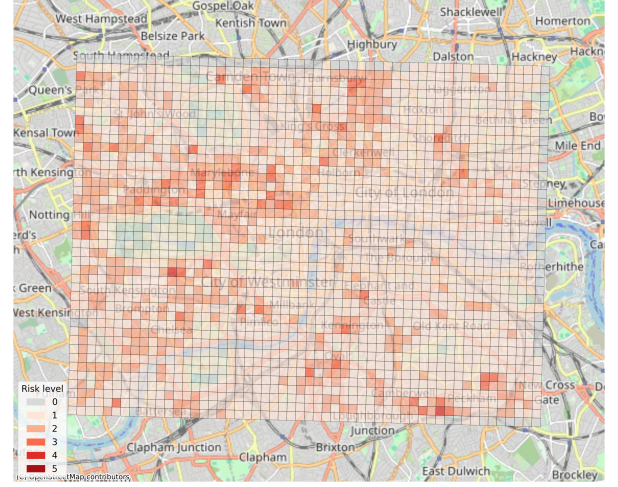


Figure 2. Risk map of central London.

The scheduling horizon is set to 24 hours, discretized into one-second intervals. Flight events are synthetically generated using a stochastic procedure that samples origins, destinations, expected launch times, and speeds across four UAV service categories: emergency, police, delivery, and transport. Following the UAV demand forecasts in our previous study, category-specific temporal patterns are incorporated to emulate realistic urban operations. A fixed random seed is used to ensure reproducibility. The flight proportions are set to 78.67% delivery, 7.96% transport, 12.69% police, and 0.68% emergency, reflecting predicted urban UAV demand. For each category, expected release times, speeds, and priorities of OD pairs are generated according to the probabilistic rules described below, with the finish time determined by sampled speed and path lengths, capped by the daily horizon.

1) *Emergency response:* For emergency flights, origins o are sampled around a corner c_{corner} of the grid, while destinations d are sampled near the opposite diagonal corner c_{diag} . As shown in Equation 17, σ denotes the standard deviation of the Gaussian distribution, and I is the identity matrix, ensuring an even spread of activities in all directions.

$$o \sim \mathcal{N}(c_{\text{corner}}, \sigma^2 I), \quad d \sim \mathcal{N}(c_{\text{diag}}, \sigma^2 I). \quad (17)$$

This design ensures diagonal coverage of the study area, emulating long-range emergency dispatch. Launch times occur throughout the day, with a 70% probability of being daytime-biased (13:00 ± 4 h); otherwise, they are drawn uniformly. Speeds v are sampled from a continuous uniform distribution over the range [20, 28]. The assigned priority is 3.

2) *Police service:* For police flights (Equation 18), origins o are drawn from k Gaussian clusters centered at c_i . Destinations d are sampled from the opposite half of the grid relative to

o with probability p_{opp} ; otherwise, they are drawn uniformly from the entire grid.

$$o \sim \mathcal{N}(c_i, \sigma^2 I), \quad i \in 1, \dots, k \quad (18)$$

$$d \sim U(c_{1-i}, p_{opp}), \quad i \in 1, \dots, k \quad (19)$$

An OD pair is accepted only if the distance between o and d satisfies the endurance limits given in Equation 20 to ensure feasible police missions.

$$s_{\min} \leq \|o - d\| \leq s_{\max}. \quad (20)$$

Launch times follow a bimodal Gaussian distribution, with peaks at $02:00 \pm 1$ h and $22:00 \pm 2$ h. Speeds are sampled uniformly over $[12, 20]$, and the assigned priority is 2.

3) *Delivery*: For delivery flights, a hub-and-spoke structure is adopted. Hubs h are sampled uniformly across the grid, while spokes s_j are generated around spoke centers c_j according to Equation 21, where σ_{sp} controls the spread of destinations around each center.

$$h \sim \mathcal{N}(c, \sigma^2 I), \quad s_j \sim \mathcal{N}(c_j, \sigma_{sp}^2 I). \quad (21)$$

Each OD pair is formed as either $h \rightarrow s_j$ or $s_j \rightarrow h$ with equal probability. OD distances d must also satisfy the constraint in Equation 20. Launch times follow peaks in working hours ($10:00 \pm 2$ h and $15:00 \pm 2$ h). Speeds v are sampled uniformly over $[8, 18]$, and the assigned priority is 1.

4) *Passenger transport*: For passenger transport flights, predefined m urban hubs are selected uniformly. OD pairs are formed as $h_i \rightarrow h_j$ with $i \neq j$, and both endpoints are perturbed by Gaussian noise to capture local dispersion (Equation 22 and 23).

$$o \sim \mathcal{N}(h_i, \sigma_{\text{hub}}^2 I), \quad i \in 1, \dots, m \quad (22)$$

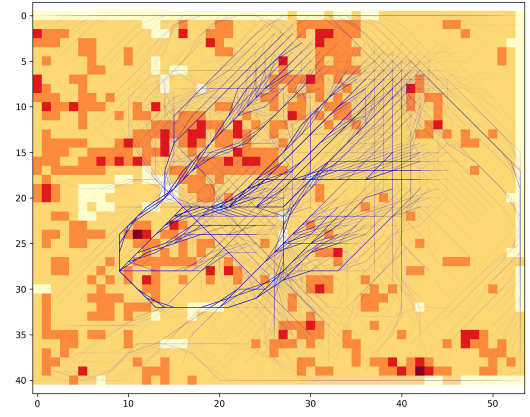
$$d \sim \mathcal{N}(h_j, \sigma_{\text{hub}}^2 I), \quad i \in 1, \dots, m \quad (23)$$

Launch times follow commuter peaks at $08:00 \pm 1.5$ h and $18:00 \pm 1.5$ h. Speeds v are sampled uniformly over $[16, 24]$, and the assigned priority is 1.

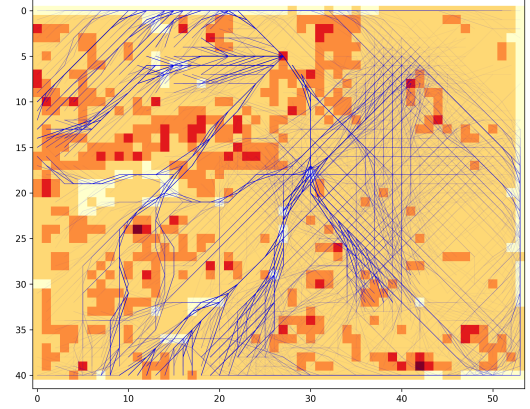
B. Result Evaluation

To assess the effectiveness of the scheduler in mitigating operational risk, we visualize planned flight paths for scenarios with 5,000, 10,000, and 15,000 flights, as shown in Figure 3. In these plots, cell colors indicate the underlying risk levels. Across all scales, the scheduled flows consistently avoid the highest-risk regions and concentrate primarily in lower-risk areas, demonstrating the scheduler's ability to embed risk awareness into large-scale UAV traffic planning.

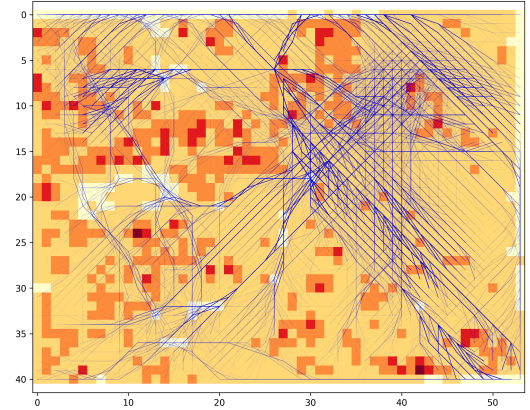
To further assess the scheduler's risk avoidance capability, we analyze scheduled path distributions across risk levels using three complementary indicators: cell coverage (Figure 4a), cell crossings (Figure 4b), and traffic volume (Figure 4c).



(a) Planned paths for 5,000 flights.



(b) Planned paths for 10,000 flights.



(c) Planned paths for 15,000 flights.

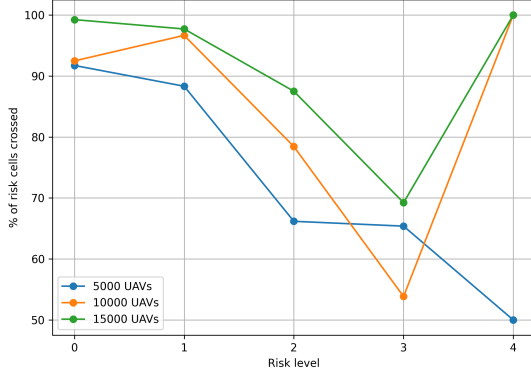
Figure 3. Planned UAV paths under different traffic scales.

The cell coverage (Figure 4a) measures the proportion of cells visited by UAVs at each risk level. Results show that nearly all level 0-1 cells are traversed (over 90%), while coverage decreases sharply at higher levels, falling below 70% for 10,000 flights and below 55% for 15,000 flights at level-3 cells. A minor exception arises in Figure 4a, where every available level-4 cell is traversed in the 10,000- and 15,000-flight scenarios. However, full coverage does not imply frequent visits; rather, it results from only a handful of flights that cannot avoid the highest-risk cells, as confirmed in Table I.

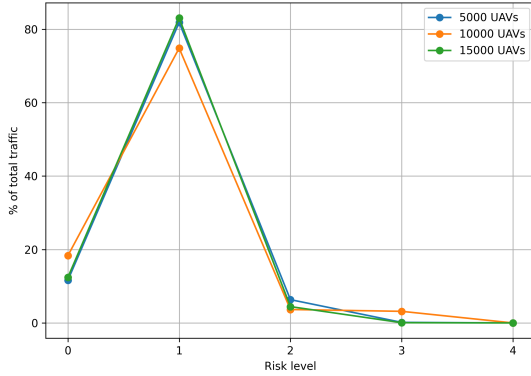
The distribution of cell crossings across risk levels captures where most UAV movements occur. As shown in Figure 4b, more than 75-83% of crossings are concentrated in level 1, 12-18% in level 0, fewer than 5% in levels 2-3, and level 4 is almost entirely avoided.

The share of traffic volume across risk levels reflects how many flights entered each risk level, as shown in Figure 4c. Over 75-80% of flights operated in level 1, around 15-20% in level 0, and fewer than 5% in levels 2-4 combined.

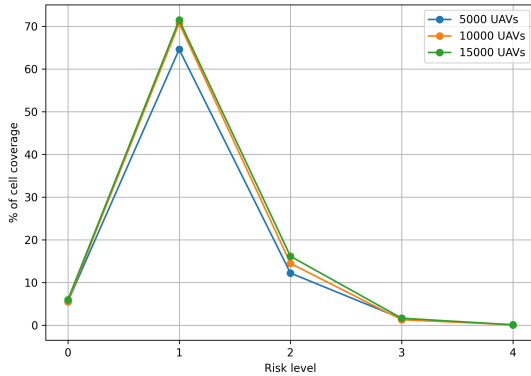
Overall, these results show that UAV operations are overwhelmingly concentrated in the safest regions, with only negligible exposure to higher-risk zones.



(a) Share of visited cells at each risk level.



(b) Share of total UAV crossings at each risk level.



(c) Share of traffic volume at each risk level.

Figure 4. Distribution of UAV Operations Across Risk Levels: Coverage, Crossings, and Traffic Share.

Beyond conflict avoidance, the scheduler's overall performance across all scenarios is summarized in Table I. The results confirm that all flights are successfully accommodated, even at the highest density of 15,000 UAVs. As expected, operational risk increases slightly with scale, with the number of highest-risk cell crossings rising from just 1 for 5,000 flights to 5 for 15,000 - still negligible relative to the total traffic. Average travel distance also grows modestly, from 4.64 km for 5,000 flights to 4.88 km for 10,000. Interestingly, average launch delays decrease sharply as density increases - from nearly 14 minutes for 5,000 flights to well under one minute for 10,000 and 15,000 flights - because more flights are detoured rather than held back. This trend highlights the scheduler's flexibility in managing large-scale operations by exploiting path diversity and balancing constraints.

TABLE I. PERFORMANCE OF THE HYBRID SCHEDULER

Number of UAVs	5,000	10,000	15,000
Proportion of Flights scheduled	100%	100%	100%
Number of high-risk cells crossed	1	4	5
Average travel distance (km)	4.64	4.88	4.82
Average launch delay (mins)	13:51	00:07	00:03
Runtime	00:05:24	00:24:32	00:45:08

An example of the temporal distribution of UAV operations is shown in Figure 5, which illustrates the spatial positions of UAVs in the 15,000-flights scenario at six-hour intervals. At the start ($t = 0$ h), UAVs are sparse and clustered near departure nodes. By $t = 6$ h, traffic disperses across the area, with flows concentrating along low-risk corridors. At peak demand ($t = 12$ h), UAV density is highest, with widespread coverage and intensive use of low-risk cells. By $t = 18$ h, traffic density decreases, but operations still follow preferred low-risk paths, reflecting the temporal dynamics of large-scale UAV scheduling.

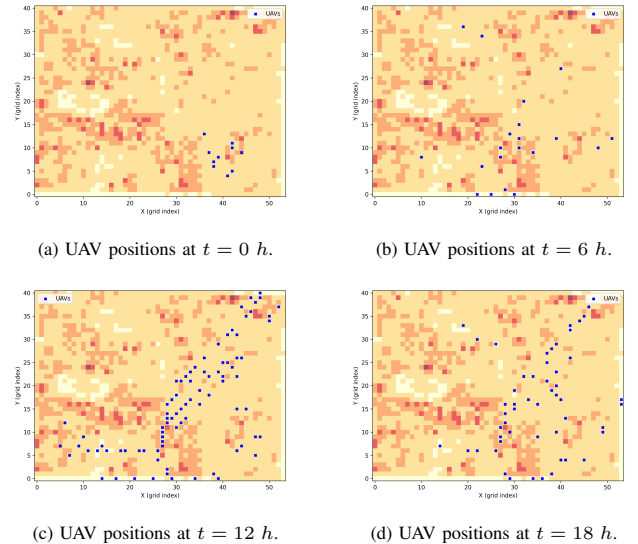


Figure 5. UAV positions at six-hour intervals in the 15,000-event scenario

TABLE II. SENSITIVITY ANALYSIS OF THE RISK THRESHOLD $R_{\max}$

Value of $R_{\max}$	$+\infty$	0.2	0.1 (default)	0.05
Proportion of flights scheduled	100%	100%	100%	100%
Number of high-risk cells crossed	110	4	4	4
Average travel distance (km)	4.81	4.82	4.88	5.11
Average launch delay (m:s)	00:07	00:07	00:07	00:07
Runtime (h:m:s)	00:24:27	00:23:27	00:24:32	00:23:19

The sensitivity analysis in Table II examines how different values of the risk threshold $R_{\max}$ affect the scheduler's performance. When no restriction is applied ($R_{\max} = +\infty$), 110 high-risk cells are crossed. Introducing even a mild limit ($R_{\max} = 0.2$) immediately reduces this number to four, and further tightening ($R_{\max} = 0.1$ or 0.05) yields no additional improvement—indicating rapid convergence toward a near-optimal risk setting. Meanwhile, the proportion of scheduled flights remains at 100% across all scenarios, showing that risk control does not compromise feasibility. The average travel distance increases slightly as $R_{\max}$ becomes more restrictive (from 4.81 to 5.11 km), reflecting minor path detours to avoid high-risk zones. Both launch delay and runtime remain basically unchanged, confirming that tighter risk control imposes no measurable computational or temporal penalty.

C. Benchmark Comparison

The proposed risk-aware hybrid scheduler is evaluated against a risk-unaware benchmark that relies solely on a greedy heuristic. Table III reports the results of the baseline across the same indicators used for the hybrid scheduler.

TABLE III. PERFORMANCE OF THE GREEDY-BASED BENCHMARK

Number of UAVs	5,000	10,000	15,000
Proportion of flights scheduled	96.16%	86.76%	69.30%
Number of high-risk cell crossings	30	100	35
Average travel distance (km)	4.54	4.84	4.80
Average launch delay (m:s)	20:17	22:46	26:06
Runtime (h:m:s)	00:04:07	00:07:06	00:16:16

The results confirm that the proposed scheduler consistently outperforms the greedy-based benchmark in terms of capacity, safety, and efficiency. In all three scenarios, the hybrid scheduler achieves a 100% scheduling rate, whereas the benchmark rapidly loses capacity as demand grows, dropping from 96.16% for 5,000 flights to only 69.30% for 15,000 flights.

Most importantly, the hybrid scheduler maintains consistently low operational risk, with only 1–5 high-risk cell crossings across all scales. In contrast, the greedy-based benchmark records 30 crossings for 5,000 flights, increasing to 100 for 10,000 flights before dropping to 35 for 15,000. This fluctuation indicates that the greedy algorithm lacks robustness in risk management, whereas the hybrid framework systematically minimizes such occurrences.

In terms of efficiency, both approaches produce comparable average travel distances, though the greedy-based benchmark yields slightly shorter paths (e.g., 4.80 km versus 4.82 km for 15,000 flights). This difference is expected, as the hybrid

scheduler accepts minor detours to avoid high-risk cells. In contrast, the hybrid scheduler maintains negligible launch delays—under one minute even for 15,000 flights—while the benchmark incurs delays exceeding 20 minutes, reaching 26:06 minutes for 15,000 flights.

However, comparing the results of the greedy-only benchmark and the hybrid scheduler shows that the hybrid approach requires a longer computation time. The difference is negligible for 5,000 flights (approximately 7 minutes versus 5 minutes) but grows with increasing traffic density, becoming most pronounced at 15,000 flights (around 16 minutes versus 45 minutes), though still within an acceptable range.

Overall, the comparison highlights the hybrid scheduler's advantages in simultaneously minimizing delays and operational risks for large-scale UAV operations, alongside the drawback of increased computation time at larger operations. These findings demonstrate the potential of the proposed framework in real-world UTM applications and underscore the need for future computational enhancements.

VI. CONCLUSION

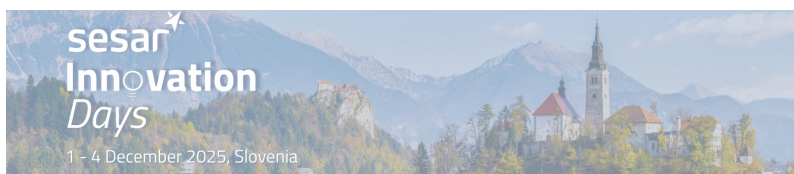
This paper develops a risk-aware spatio-temporal scheduling framework for large-scale urban UAV operations that integrates operational risk assessment, greedy search, and MILP with iterative constraint generation. The proposed scheduler enhances both efficiency and safety in low-altitude urban airspace by minimizing makespan, launch delays, and exposure to high-risk cells within a reasonable computation time.

Simulation results with up to 15,000 UAVs demonstrate that the proposed framework effectively balances conflict avoidance, risk mitigation, and operational efficiency, particularly at larger scales. It successfully schedules 13.24%-30.70% more flights while reducing operational risk by 85.71%-96.67% compared to the benchmark, underscoring the scalability and practicality of the scheduler.

Overall, this work contributes to a systematic approach for risk-informed UAV traffic management. Future research will address uncertainties in urban airspace, improve computational efficiency, and extend the solver toward tactical conflict resolution. The codes will be made publicly available once they are extended as part of future research.

REFERENCES

- [1] T. Chauvin, D. Gianazza, and N. Durand, "Orca-a*: A hybrid reciprocal collision avoidance and route planning algorithm for uas in dense urban areas," in *Sesar innovation days 2024*, 2024.
- [2] S. J. Undertaking, "U-space blueprint," SESAR Joint Undertaking, Tech. Rep., 2021, edition 2021, Brussels.



- [3] R. Isufaj, M. Omeri, and M. A. Piera, "Multi-uav conflict resolution with graph convolutional reinforcement learning," *Applied Sciences*, vol. 12, no. 2, p. 610, 2022.
- [4] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Review of conflict resolution methods for manned and unmanned aviation," *Aerospace*, vol. 7, no. 6, p. 79, 2020.
- [5] S. J. Undertaking, "U-space concept of operations, 4th edition," SESAR Joint Undertaking (CORUS-XUAM project), Technical Report Version 4.0, 2023.
- [6] H. Ghazzai, A. Kadri, M. Ben Ghorbel, H. Menouar, and Y. Massoud, "A generic spatiotemporal uav scheduling framework for multi-event applications," *IEEE Access*, vol. 7, pp. 215–229, 2019.
- [7] P. Wan, G. Xu, J. Chen, and Y. Zhou, "Deep reinforcement learning enabled multi-uav scheduling for disaster data collection with time-varying value," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 6691–6702, 2024.
- [8] H. Ghazzai, H. Menouar, A. Kadri, and Y. Massoud, "Future uav-based its: A comprehensive scheduling framework," *IEEE Access*, vol. 7, pp. 75 678–75 695, 2019.
- [9] F. Ho, R. Geraldes, A. Gonçalves, B. Rigault, B. Sportich, D. Kubo, M. Cavazza, and H. Prendinger, "Decentralized multi-agent path finding for uav traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 997–1008, 2022.
- [10] B. Pang, K. H. Low, and C. Lv, "Adaptive conflict resolution for multi-uav 4d routes optimization using stochastic fractal search algorithm," *Transportation Research Part C: Emerging Technologies*, vol. 139, p. 103666, 2022.
- [11] Q. Tan, Z. Wang, Y.-S. Ong, and K. H. Low, "Evolutionary optimization-based mission planning for uas traffic management (utm)," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 952–958.
- [12] N. Peng, Y. Xi, J. Rao, X. Ma, and F. Ren, "Urban multiple route planning model using dynamic programming in reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8037–8047, 2022.
- [13] M. Hu, W. Liu, K. Peng, X. Ma, W. Cheng, J. Liu, and B. Li, "Joint routing and scheduling for vehicle-assisted multidrone surveillance," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1781–1790, 2018.
- [14] Y. Wu, K. H. Low, and X. Hu, "Trajectory-based flight scheduling for airmetro in urban environments by conflict resolution," *Transportation Research Part C: Emerging Technologies*, vol. 131, p. 103355, 2021.
- [15] L. Yang, X. Zhang, Y. Zhang, and G. Xiangmin, "Collision free 4d path planning for multiple uavs based on spatial refined voting mechanism and pso approach," *Chinese Journal of Aeronautics*, vol. 32, no. 6, pp. 1504–1519, 2019.
- [16] M. Lissone and S. G. JARUS, "Joint authorities for rulemaking of unmanned systems," in *Proceedings of the ICAO RPAS Symposium*. Montreal: ICAO RPAS Symposium, 2017.
- [17] N. P. Borges, C. G. Ghedini, and C. H. C. Ribeiro, "Mitigating risk in uav aid delivery missions: A safe path prioritization approach," *IEEE Access*, vol. 12, pp. 168 765–168 778, 2024.
- [18] Y. Liu, H. Huang, Q. Xu, S. Xu, and J. Wang, "Safer conflict-based search: Risk-constrained optimal pathfinding for multiple connected and automated vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 17 382–17 396, 2025.
- [19] X. Hu, B. Pang, F. Dai, and K. H. Low, "Risk assessment model for uav cost-effective path planning in urban environments," *IEEE Access*, vol. 8, pp. 150 162–150 173, 2020.
- [20] W. Wang, E. Beech, A. Majumdar, W. Ochieng, and J. Escribano, "Agent-based modeling approach for operational risk assessment of large-scale low-altitude uav traffic," in *2025 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, 2025, pp. 1–11.
- [21] W. Wang, M. Zhang, E. Beech, A. Majumdar, W. Ochieng, and J. Escribano, "risk-based truck-drone delivery optimization," in *The 12th Triennial Symposium on Transportation Analysis conference*, 2025, pp. 1–4.