Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Rule-Decoupled and Predictive Hierarchical Path Planning for Composite-Wing UAVs in Dynamic Urban Airspace

Hanyu Liu
Guangzhou College of Commerce
Meng Wang
wangmeng@gcc.edu.cn
Guangzhou College of Commerce
Ying Li
Guangzhou College of Commerce

Research Article

Keywords: composite-wing UAV, hierarchical path planning, rule decoupling, conflict prediction, finite acceleration, dynamic obstacle avoidance

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11498295/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Rule-Decoupled and Predictive Hierarchical Path Planning for Composite-Wing UAVs in Dynamic Urban Airspace

Hanyu Liu[1] · Meng Wang[2],* · Ying Li[1]

1 School of Modern Information Industry, Guangzhou College of Commerce, Guangzhou 511363, China

2 Department of Intelligent Science and Engineering, Guangzhou College of Commerce, Guangzhou 511363, China

* Corresponding author: Meng Wang （ORCID：0009-0000-6307-5921，wangmeng@gcc.edu.cn ）

Other authors:

Hanyu Liu: （ORCID：0009-0001-8543-3301， 2996703643@qq.com ）

Ying Li: （ORCID：0009-0009-2058-8431， 3270838573@qq.com ）

Abstract

Composite-wing unmanned aerial vehicles (UAVs) in dynamic urban airspace must coordinate planning with exclusion rules, moving traffic, and limited turning and braking capabilities. This paper presents Rule-Decoupled and Predictive Hierarchical Path Planning (RDPHPP). An independent interface propagates exclusion geometry, while direction-augmented multi-cost search, Bézier smoothing, and conditional circular fillet recovery construct acceptable references. Prediction and execution share an acceleration-limited integrator to compare continuation, speed yielding, and detours; speed-regime dynamic window control revalidates and executes candidates. On Shenzhen and Guangzhou maps, all 116 successful paths from 60 rule-replacement pairs satisfy exclusion constraints. A fixed-input fillet diagnostic increases reference acceptance from 40/60 to 57/60. Across 540 paired dynamic scenarios, RDPHPP completes 515 missions (95.37%). Its success-rate difference from the generalized velocity obstacle (GVO) adaptation is significant after multiplicity correction, whereas differences from Reactive and No-wait are not significant. In independent controlled execution, violations of the time-growing separation margin occur in 0/256 conditions accepted by finite-deceleration prediction and 27/264 accepted by an instantaneous approximation. These results support rule-configuration propagation, geometric reference construction, and finite-deceleration candidate evaluation, providing a verifiable workflow connecting hierarchical planning stages. The main dynamic study still records 25 collisions, and mission performance incurs flight-distance and modeled energy-consumption costs. The independent benefit of waiting and closed-loop safety on real aircraft require further validation.

Keywords: composite-wing UAV; hierarchical path planning; rule decoupling; conflict prediction; finite acceleration; dynamic obstacle avoidance

1 Introduction

Urban air mobility (UAM) connects low-altitude tasks, such as logistics delivery and urban inspection, with densely built environments, requiring routes to adapt to mission rules and moving traffic. Composite-wing platforms capable of vertical takeoff and landing and cruise flight offer one option for these applications. Their maneuvering conditions vary across flight phases, however, placing additional demands on the connection between planning and execution. Research on autonomous flight with actual tailsitter aircraft has demonstrated the importance of nonlinear aerodynamic constraints in trajectory generation [14].

Urban low-altitude flight requires routes between buildings and exclusion zones while avoiding traffic whose positions continually change. Buildings and exclusion rules define where passage is permitted; dynamic traffic determines when that space is available; and the aircraft's turning and braking capabilities determine whether an avoidance maneuver can be executed from its current state. For composite-wing unmanned aerial vehicles (UAVs) with lift assistance at low speeds, low-speed maneuvers provide additional response options, but deceleration, turning, and recovery to cruise speed still take time. A building-avoiding polyline must therefore

be assessed for its suitability as an executable reference and for whether the UAV can change its arrival time or route through the conflict region soon enough.

Existing methods provide foundations for these stages. A* searches for paths on a specified graph with a specified cost function [8]. Conventional two-dimensional distance-based search primarily addresses spatial connectivity and path length; the turning space needed for smoothing and the timing of dynamic traffic require further treatment. The dynamic window approach (DWA) rolls out short-horizon trajectories within a reachable velocity set and selects local actions [6]. It predicts the UAV's short-term motion but, when used alone, still requires support from global routes and configurable exclusion rules. Velocity obstacles (VO) and generalized velocity obstacles (GVO) exclude collision candidates through relative motion or constrained control [5, 23], without directly configuring urban rule geometry or constructing reference paths. Energy-oriented hierarchical planning for fixed-wing aircraft and online UAV mission planning have demonstrated cooperation between global search and local execution [2, 16]. This paper further examines how these modules propagate constraints, compare dynamic responses, and handle candidates that become invalid during execution within a single system.

These methods address spatial search, local action selection, or collision prediction individually. When combined for an urban mission, however, their modules must also exchange consistent rules, timing information, and motion conditions. This paper investigates three gaps that directly affect candidate execution.

The first is coupling between configuration and algorithms caused by hard-coded rules. If urban exclusion zones and their checks are embedded in search, smoothing, or local control code, changing cities or exclusion configurations requires modifying rule checks within the algorithms. Incomplete modifications can also cause different layers to use different feasible domains. City-specific rules should instead be separated into a configuration and compiled through a common interface into constraint geometry shared by search and acceptance checks. Rule changes can then propagate to the planning results while the search expansion logic remains unchanged.

The second is misalignment between prediction timing and execution with finite acceleration. Computing an estimated time of arrival (ETA) at a fixed flight speed, or approximating speed yielding as an instantaneous speed reduction, omits the forward motion during braking and subsequent acceleration. The predicted encounter position and passage time then deviate from actual motion. Even when traffic is predicted at constant velocity, the UAV cannot reach its target speed instantaneously. Prediction and execution should share an acceleration-limited integrator initialized from the same current state, rather than separately using fixed ETAs and constrained motion trajectories.

The third is the lack of a common motion envelope for accepting detour and waiting candidates. Checking only whether a spatial detour exists or when an obstacle clears the conflict region, without jointly checking the UAV's acceleration, turning, and braking constraints, can accept a maneuver that cannot turn or decelerate in time. A path may exist while its execution results in collision or infeasible braking. Candidates should pass trajectory acceptance checks under a common motion envelope before their costs are compared. Changes in state between computation and adoption also require adoption revalidation and explicit failure handling.

Risk-aware prediction and planning switches [27], safe interval path planning, and its kinematic extensions [3, 17] provide foundations for predictive switching and passage timing. The connections studied here require jointly considering the legal space defined by urban rules, arrival times resulting from the UAV's finite deceleration, and the motion states that candidates can actually reach.

Accordingly, the research question is: under configurable exclusion rules, constant-velocity traffic prediction, and bounded acceleration, how can acceptable paths be generated that support dynamic yielding or detours and connect the selected response continuously to local execution? To answer this question, this paper proposes Rule-Decoupled and Predictive Hierarchical Path Planning (RDPHPP). Its rule interface propagates configuration changes into a feasible domain shared by all layers. Reference construction prepares geometrically acceptable paths for dynamic responses. Time-aligned prediction incorporates the motion required for deceleration and turning into candidate comparisons. The execution layer continuously checks the selected plan and records invalid and infeasible states.

Figure 1 distinguishes two types of dynamic response. Speed yielding retains the reference route and delays arrival through finite deceleration; a spatial detour changes the reference route and reallocates the space used to pass through the conflict region. RDPHPP first constructs a reference satisfying curvature and clearance conditions, compares the two responses with a common motion model, and passes the accepted route or speed limit to the local controller. This process must retain both the conditions under which each candidate remains valid and the conditions that require its revocation.

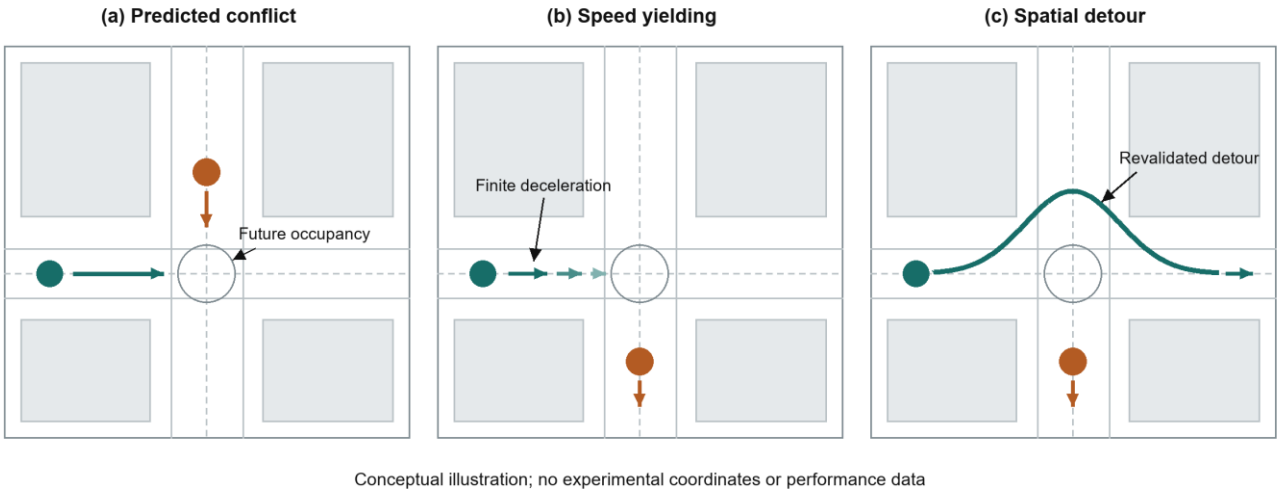


Figure 1. Conceptual relationship between a predicted conflict and two dynamic responses. (a) The original reference produces a predicted conflict with crossing traffic; (b) finite deceleration changes the arrival time; (c) a detour changes the spatial route. This programmatically drawn illustration explains the response types only; it does not represent an experimental map, an actual trajectory, or a safety guarantee.

The five main contributions are as follows.

(1) Rule interface and acceptance checks at the system level. Exclusion-rule configurations, search costs, and observation snapshots are passed separately. Geometric compilation converts rules into hard constraints shared by search, smoothing, and local execution. Compared with encoding urban rules separately within each module, configuration replacement propagates through a common interface to planning and acceptance checks while preserving the search expansion logic. Rule-replacement experiments directly examine this constraint propagation.

(2) Direction-augmented multi-cost search and conditional circular fillet recovery at the algorithm level. The A* state retains the incoming direction and combines distance, turning, and obstacle-proximity costs to reserve geometric space for subsequent connections. Bézier smoothing and conditional circular fillet recovery are attempted sequentially within a shared planning budget. Compared with selecting a path solely by distance and passing its polyline directly to smoothing, the proposed procedure uses curvature and clearance checks to determine whether a reference can be adopted. A fixed-input on/off diagnostic isolates the effect of circular fillet recovery.

(3) Time-aligned predictive arbitration at the algorithm level. A shared acceleration-limited integrator rolls out continuation, speed-yielding, and detour candidates from the same current state. Their costs are compared after common motion and timing checks, with explicit management of the waiting budget, commitment revocation, and temporary swept-region guidance. Compared with fixed ETAs or instantaneous deceleration approximations, candidate decisions include the forward distance traveled during deceleration and are revalidated when the state changes before adoption.

(4) Speed-regime DWA and explicit braking failure handling at the execution level. Reachable controls are screened against a speed-dependent envelope, and the state at which a candidate is adopted is revalidated. When normal actions are infeasible, finite braking and conditional local recovery are checked in sequence. If neither is feasible, the system records an infeasible state and outputs a fallback target. Instead of treating a zero-

speed command as an already completed safe stop, the procedure checks the motion required to reach that target and distinguishes a braking target, feasible braking, and braking failure.

(5) Paired two-city simulations and negative results at the validation level. Rule replacement, static references, and dynamic execution are compared in paired experiments on frozen Shenzhen and Guangzhou maps. Independent controlled diagnostics examine the acceptance conditions for finite deceleration. Beyond final success rates, the evaluation retains reference rejection, candidate invalidation, collisions, small surface-to-surface separation, efficiency costs, and long-tail computation times to examine each layer's contribution and scope.

These contributions build on established search, curve-connection, and local-control components to specify a generation, acceptance, and execution procedure under shared rules and constrained motion. The composite-wing application is abstracted through low-speed lift assistance and speed-regime maneuver envelopes; the study is limited to constant-altitude two-dimensional kinematic simulation. Section 2 reviews related work. Sections 3 and 4 present the problem and method. Section 5 analyzes five experiment groups. Section 6 discusses the method's strengths, its relation to prior work, and its limitations. Section 7 concludes the paper.

2 Related Work

2.1 Urban Constraints and Hierarchical UAV Planning

Planning constraints in urban environments include both impassable geometry and regions restricted by mission or operational rules. Risk-map research uses urban ground risk to guide path planning [18], but ground-risk quantification and airspace exclusion rules have different meanings: the former typically affects cost, whereas the latter directly excludes illegal paths in this paper. Exclusion rules are therefore compiled into constraint geometry, while a separate cost interface describes preferences for distance, turning, and obstacle proximity. Rule changes still affect the distance field and path cost. Decoupling here means separating configuration from implementation interfaces, rather than mathematical independence.

Aiello et al. proposed energy-oriented global search and local obstacle avoidance for fixed-wing aircraft, demonstrating the feasibility of hierarchical treatment of static and dynamic threats [2]. Mugnai et al. connected local obstacle avoidance, trajectory planning, and a mission state machine, showing how velocity and acceleration constraints affect mission trajectories [16]. Their cooperation between global and local modules provides a foundation for this work. RDPHPP further specifies the interfaces that propagate rule changes, the acceptance conditions for reference paths, and the motion constraints shared by prediction and actual actions, allowing the feasibility conditions of a module's output to remain checkable at the next layer.

Recent studies have further improved cooperation between modules. He et al. improved global reference tracking through two-layer A* optimization and DWA evaluation [9]. Xu et al. combined expanded-neighborhood search, circular fillet smoothing, and artificial-potential-field dynamic obstacle avoidance [25]. These results establish a clear basis for combining search, smoothing, and local avoidance. The contribution to clarify here is how urban rules propagate, how deceleration timing aligns with execution, and how different responses undergo common acceptance checks, rather than the combination itself.

2.2 Dynamic Prediction, Velocity Obstacles, and Temporal Decisions

DWA rolls out the UAV's short-horizon trajectories in reachable velocity space and checks their feasibility [6]. This paper models these short-horizon ego-motion rollouts separately from responses based on longer-horizon traffic prediction; Reactive retains the former. VO describes collision-inducing velocity sets through relative motion, and GVO extends this formulation to systems with motion constraints [5, 23]. FIMVO further integrates multiple models and fuzzy decision-making [4]. These studies provide substantial foundations for motion constraints and traffic prediction. The comparison here uses a nonreciprocal GVO adaptation over a shared control set; the design focuses on organizing route changes, finite deceleration, and local execution within one acceptance procedure.

Risk-aware prediction and planning switches have clear precedents [27]. This paper uses simple constant-velocity obstacle extrapolation and focuses on alignment with the UAV's finite acceleration and deceleration,

rather than proposing a new target-motion estimator. Safe interval path planning (SIPP) represents dynamic occupancy through available time intervals [17], and safe interval path planning with interval projection (SIPP-IP) extends this idea to kinodynamic constraints [3]. Unlike time-expanded search, the proposed method generates detour paths in two-dimensional space and then checks temporal feasibility through finite-horizon rollouts. This implementation facilitates reuse of the global planner but may miss feasible spatiotemporal solutions.

PE-Planner incorporates dynamics, disturbance estimation, and a global reference into local predictive optimization, while DYNUS combines spatiotemporal corridors, uncertainty, and emergency trajectories [13, 19]. Algorithmic precedents therefore exist for consistency between finite motion and prediction. Using frozen two-dimensional maps and candidate rollouts, this paper specifically examines how the deceleration process and the state at adoption affect explicit acceptance conditions.

2.3 Motion Feasibility and Safety Conditions

Curvature-constrained smoothing provides a foundation for converting discrete paths into flight trajectories [26], but acceptable curvature alone does not ensure dynamic collision avoidance, and local maneuverability remains speed-dependent. MADER achieves decentralized planning in dynamic environments through shared trajectories [22], and Robust MADER further handles communication delays [12]. RLSS studies trajectory planning through local separation [21]. Their information assumptions and feasibility requirements are integral to interpreting their safety results. Traffic in this paper neither actively yields nor exchanges committed trajectories with the UAV, so safety guarantees from cooperative multi-UAV methods cannot be transferred directly.

Continued safety also depends on feasible subsequent actions. At every planning iteration, SUPER generates both a safe trajectory in known free space and an exploratory trajectory for higher-speed flight [20]. gatekeeper revalidates safety through trajectory commitment and backup control [1]. The finite-braking checks and failure labels in this paper make infeasible states traceable, but do not establish comparable conditions for continued safety.

Existing research has advanced search and local tracking, temporal planning under motion constraints, trajectory commitment, and safety revalidation, but its information assumptions and validation endpoints differ. Table 1 compares specific designs against the three interface questions studied here: how rules propagate as independent inputs, whether prediction includes the UAV's finite-motion process, and how candidates enter acceptance checks at the next layer. For work that does not study urban rule configuration, the table identifies interfaces that would need to be added, without treating unaddressed capabilities as algorithmic deficiencies.

Table 1. Relationship between related methods and the three research gaps. Comparisons reflect the designs reported in the cited papers. An interface described as needing specification or addition is required to connect the method to the task studied here; this does not imply that the original method cannot be extended.

Study	Rule-decoupling interface	Prediction and finite motion	Candidate and execution acceptance
A*–DWA integration [9]	Optimizes search and reference tracking; urban-rule propagation requires separate specification	Local trajectory rollouts; this paper additionally examines longer-horizon finite deceleration	Global references guide local control; this paper explicitly separates reference and dynamic acceptance checks
Circular fillet smoothing and potential fields [25]	Obstacle geometry enters search; an exclusion-configuration interface needs to be added	Potential-field responses to moving targets; this paper compares passage times	Improves geometry; this paper applies common curvature, clearance, and motion checks
Fixed-wing hierarchical planning [2]	Energy-oriented search and local avoidance; a rule interface needs to be added	Global–local cooperation; this paper examines finite-deceleration timing	This paper additionally specifies references, proposals, and adoption revalidation
GVO / FIMVO [4, 23]	Collision avoidance through constrained control; reference paths and rule geometry require	Relative motion, multiple models, or constrained dynamics	This paper implements the GVO adaptation using a shared reachable set

Study	Rule-decoupling interface	Prediction and finite motion	Candidate and execution acceptance
	integration		
SIPP-IP [3]	Dynamic occupancy on a graph; rules can map to infeasible states	Safe time intervals explicitly account for acceleration and deceleration	Provides completeness and optimality for its problem; this paper performs finite-horizon revalidation only
PE-Planner / DYNUS [13, 19]	Dynamics or spatiotemporal corridors; urban-configuration interfaces are outside the main focus	Dynamics, disturbance estimation, or uncertainty enters prediction	Joint trajectory and execution conditions; this paper organizes two-dimensional candidates
MADER / Robust MADER [12, 22]	Multi-UAV trajectory sharing; rule configuration requires separate specification	Commitments and communication delay; information assumptions differ from the observation-only, nonreciprocal traffic setting	Acceptance under cooperative conditions; this paper does not use communicated commitments
SUPER / gatekeeper [1, 20]	Known free space or online constraints	Planning and backup control jointly constrain subsequent motion	Explicit safe trajectories or continued safety verification; this paper retains explicit finite-braking failure handling

3 Problem Formulation

3.1 Environment, Rules, and Observations

Let $\mathcal{W} \subset \mathbb{R}^2$ denote the mission region, $\mathcal{O}_s$ the set of building solids, and $\mathscr{R}$ the rule configuration, which is compiled into the exclusion set $\mathcal{O}_r(\mathscr{R})$. The global hard-constraint set is defined by Eq. (1):

$$\mathcal{O}(\mathscr{R}) = \mathcal{O}_s \cup \mathcal{O}_r(\mathscr{R}) \cup (\mathbb{R}^2 \backslash \mathcal{W}). \quad (1)$$

The UAV is represented as a disk of radius $r_a = 8$ m. Traffic object i is represented by center $p_i(t)$ and radius r_i. Physical surface-to-surface separation is defined by Eq. (2):

$$d_i(t) = \| p_a(t) - p_i(t) \| - r_a - r_i. \quad (2)$$

Several planner thresholds are specified as the distance from the obstacle edge to the UAV center, denoted by $c_i = d_i + r_a$. All experimental minimum-separation results reported below use the body-surface definition in Eq. (2). Obstacle-edge-to-UAV-center thresholds cannot be interpreted directly as actual surface-to-surface separation.

Ground truth is used to advance the simulation and for independent evaluation. The planner consumes only observation snapshots containing object IDs, timestamps, positions, and velocity estimates. The sensing radius is 150 m, with a refresh rate of 10 Hz. Objects no longer observed are extrapolated for a 1 s validity period and then deleted. Waiting commitments are bound to object IDs and cannot continue to access objects updated from ground truth. This setting represents limited-range, noisy observations; it does not implement a camera or lidar perception pipeline.

The map uses a 5 m grid over a 2000×2000 m region. The planning distance field $D(p)$ provides a conservative lower bound on UAV-center clearance from occupied grid-cell solids and the region boundary. It is constructed by subtracting half a cell diagonal from the Euclidean distance transform and applying a Lipschitz lower bound on the distance function. Final collision evaluation independently computes the distances between the swept UAV disk over each time step and solid grid squares or dynamic disks, rather than checking discrete samples alone.

3.2 Constrained Motion Model

The UAV state is denoted by $z = (x, y, \psi, v, \omega)$, and the control input $u = (v_c, \omega_c)$ specifies target speed and angular velocity. For a time step Δt, the shared integrator approaches these targets at bounded rates according to Eq. (3):

$$v^+ = \mathrm{clip}(v + \mathrm{clip}(v_c - v, -a_v\Delta t, a_v\Delta t), 0, v_{\max}),$$
$$\omega^+ = \omega + \mathrm{clip}(\omega_c - \omega, -a_\omega\Delta t, a_\omega\Delta t). \quad (3)$$

Here, a_v = 6 m/s², a_ω = 1 rad/s², and v_max = 33.3 m/s. The maximum speed retains the speed scale proposed for the project. The linear acceleration, angular acceleration, nominal radius of 45 m, and low-speed lift assistance are simulation assumptions in the frozen code and have not been calibrated to a specific aircraft or flight tests. Finite rates explicitly represent the processes required for deceleration and turning and ensure that prediction and execution use the same parameters. Integration advances along an arc using the averages of the initial and final speeds and angular velocities, with the straight-line limit used near zero angular velocity. An existing angular velocity is not instantaneously clipped to the envelope; a candidate is rejected if its rollout violates the envelope. Actual changes in speed and angular velocity between steps are audited separately.

These parameters directly affect the acceptance set. For example, straight-line constant deceleration from 30 m/s to 8 m/s at 6 m/s² takes approximately 3.67 s and covers 40.33 m more than an instantaneous speed reduction. A lower acceleration limit increases this additional distance, while a lower angular-acceleration limit delays attainment of the target angular velocity. Section 5.5 examines this finite process, and Section 6.3 discusses the implications of the uncalibrated motion envelope. The existing degradation experiments do not replace a systematic sensitivity analysis of the acceleration parameters.

The motion envelope uses speed-dependent limits. Let $R_m = 45$ m and $A = v_{\max}^2/R_m$, with transition speed $v_m = \sqrt{AR_m}$. The low-speed regime is governed primarily by the lateral-acceleration limit; the high-speed regime additionally imposes a radius limit. The angular-velocity envelope is given in Eq. (4):

$$\Omega(v) = \begin{cases} \min(\omega_{\max}, A/\max(v,\varepsilon)), & v < v_m, \\ \min(\omega_{\max}, v/R_m, A/\max(v,\varepsilon)), & v \geq v_m, \end{cases} \quad \omega_{\max} = 1 \text{ rad/s}. \quad (4)$$

At nonzero speed, curvature and lateral acceleration satisfy $\kappa = \omega/v$ and $a_{\mathrm{lat}} = v^2\kappa$. These relationships connect geometric curvature to execution speed. The 45 m value is a nominal path-design scale and should not be interpreted as a hard minimum turning radius for all low-speed execution trajectories. The current parameters give $v_m = v_{\max}$, so the operating speed range in the main experiments primarily uses the low-speed branch of Eq. (4); the high-speed branch has no separate experimental coverage. Waiting candidates have a minimum target speed of 8 m/s, whereas the braking target can be zero; both assume that lift assistance is available. Equation (4) describes the speed-dependent envelope used for planning. Actual transition flight and stall conditions are discussed in Section 6.3.

3.3 Mission, Feasibility, and Evaluation Targets

The mission is to reach a neighborhood of the goal from a specified start. Each step first checks collisions with physical objects and rule violations, then checks whether the ground-truth position is less than 25 m from the goal, and finally checks the 240 s mission limit. A collision in the same step therefore takes precedence over arrival. Search failure, infeasible smoothing, mission timeout, and collision are recorded separately; a returned search path is not equivalent to a completed mission.

This paper distinguishes three levels of feasibility. Spatial feasibility requires a path to avoid the set in Eq. (1) and satisfy the applicable curvature and clearance thresholds. Predictive feasibility requires a rollout under the current estimates and finite horizon to pass motion and dynamic-separation checks. Mission success is determined from complete ground-truth execution. Observation error, changes in target motion, and the finite prediction horizon can still create differences between the latter two levels. Feasibility labels describe computational conditions and do not constitute a proof of collision-free closed-loop operation.

4 Rule-Decoupled and Predictive Hierarchical Path Planning (RDPHPP)

RDPHPP connects reference-path generation, dynamic-response selection, and local-action execution. The global module outputs a reference path checked for curvature and clearance, together with its applicable speed limit. The prediction module outputs continuation along the current route, a yielding commitment tied to an object ID and expiry time, or a revalidated detour reference. The local module reads the route and speed limit and selects reachable controls. Prediction updates use the latest observations and execution states at the scheduled checks; invalid commitments are revoked and references are regenerated when necessary. The layers

share exclusion geometry and an acceleration-limited integrator. Candidate adoption includes further checks of the current position, connection conditions, and target validity period. The following sections specify the inputs, outputs, and rejection conditions of each layer.

4.1 Rule Configuration and Constraint Propagation

The rule layer compiles disk, polygonal, or grid-rectangular exclusion zones into an occupancy mask. The city map and exclusion configuration are interface inputs. Search, smoothing, and local control read the same compiled result, while path quality is computed through a separate cost interface. When the city map or exclusion configuration changes, rebuilding the occupancy geometry and distance field propagates the constraints without rewriting the direction-augmented search expansion logic or exclusion checks within each layer. Section 5.2 examines this propagation through rule replacement on fixed maps; both maps use a common planning implementation. The development cost of transfer to a new city is not measured separately. The rule semantics validated here are limited to exclusion; altitude restrictions, one-way air routes, and aircraft priorities remain outside this scope.

Figure 2 presents the information flow between layers. Dynamic observations enter prediction and local avoidance and are not written back into the permanent rule set. Detour search may temporarily add an obstacle's future swept region, but this added set can be withdrawn if search fails; buildings and exclusion rules are always retained. Prediction can therefore guide spatial search without treating short-term occupancy as a permanent exclusion zone. After swept-region guidance is withdrawn, candidates must still undergo dynamic acceptance checks again.

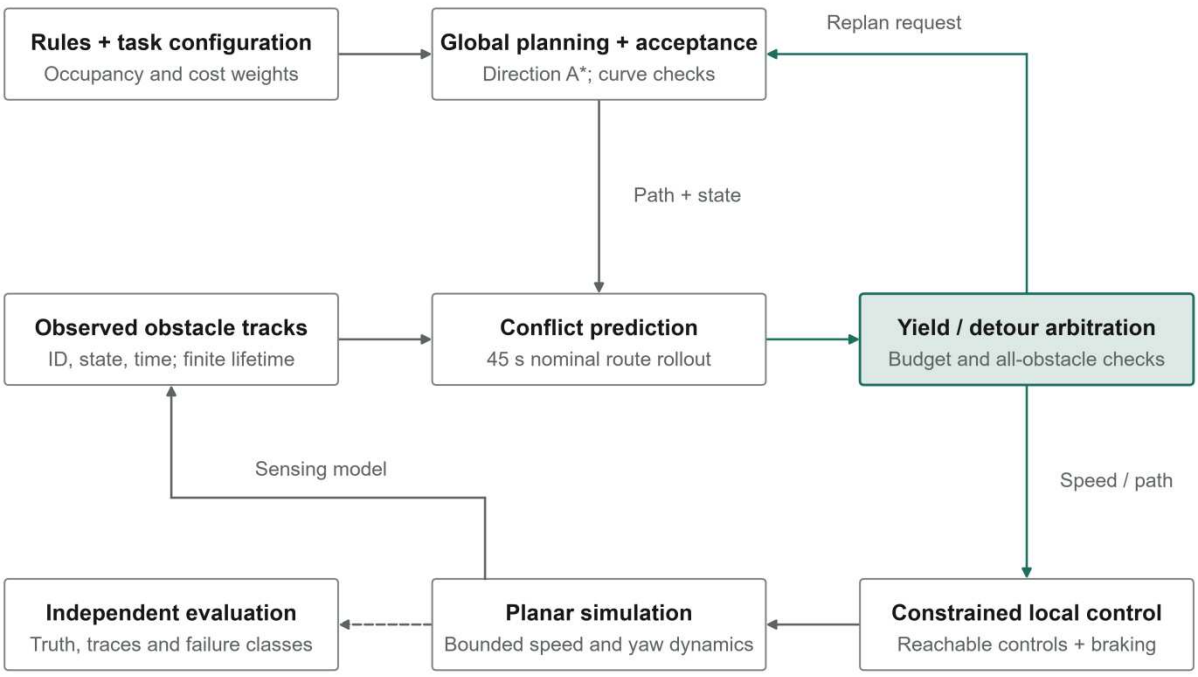


Figure 2. Information flow among rules, geometric paths, observations, and controls. Global and local modules share rule geometry. Observation snapshots are separated from simulation ground truth, and final mission evaluation uses independent ground-truth checks.

Static UAV-center clearance thresholds are 12 m for global paths, 10 m for long-horizon candidate acceptance, and 9 m for local control, corresponding to additional body-surface clearances of 4, 2, and 1 m, respectively. Smaller local clearances are allowed across layers to preserve maneuverability, but crossing physical solids or exclusion zones is prohibited. A larger global clearance does not guarantee that the same clearance is maintained throughout execution.

4.2 Direction-Augmented Search and Path Acceptance

Because turning cost depends on the incoming direction, the search state is $q = (r, c, d)$, where (r, c) is the grid position and $d \in \{0, \ldots, 7\}$ is the incoming direction. The initial direction is unspecified, and the first edge incurs no turning cost. States with different directions at the same position retain separate costs and parent pointers. Diagonal transitions cannot cut corners between adjacent orthogonally occupied cells.

The search cost of path P is given by Eq. (5):

$$J(P) = \sum_{e \in P} \left[a_\ell \ell_e + \beta (\Delta\theta_e)^2 + \delta \ell_e \max\left(0, \frac{R_s - D(q_e)}{R_s}\right)\right], \quad (5)$$

Here, ℓ_e is the edge length in meters, $\Delta\theta_e$ is the wrapped turning angle, q_e is the successor position, and $R_s = 40$ m. The frozen parameters are $a_\ell = 1.35$, $\beta = 20$, and $\delta = 1.2$. Equation (5) is a surrogate cost for distance, turning, and obstacle proximity, with the coefficients providing the corresponding dimensional conversions. The distance and "energy" terms in the historical configuration were collinear and are combined here into a_ℓ. The search objective is therefore not interpreted as independent optimization of energy consumption in joules.

The heuristic uses the octile distance in meters given in Eq. (6):

$$h(q) = a_\ell \rho \left[\Delta r + \Delta c + (\sqrt{2} - 2)\min(\Delta r, \Delta c)\right], \quad \rho = 5 \text{ m}. \quad (6)$$

The main experiments use $f = g + h$. When the other terms are nonnegative, omitting turning and proximity penalties yields a lower bound on the remaining cost in the discrete graph [8]. This property does not extend to smoothed trajectories, results returned under a finite budget, or the complete closed-loop system. The initial planning budget is 10 s, and the budget for full replanning during flight is 2 s. If no acceptable path is found within the budget, the corresponding failure state is returned.

Nearly collinear points are first removed from the raw path, and the path is simplified before quadratic Bézier curves are used to connect its corners. After normal smoothing, the unaveraged discrete peak curvature and conservative clearance are checked, preventing mean curvature from masking local peaks. Acceptance requires $\max_j |\kappa_j| \le 1.05/R$, where 1.05 is an explicitly stated numerical tolerance. The sampling scale is approximately 2.5 m. Clearance along continuous segments is further checked using a midpoint distance lower bound and the half-chord length. Bézier and circular-arc joins ensure tangential continuity at the corresponding connections; second-order continuity of the entire path is not claimed.

If the normal search, smoothing, and feedback procedure still produces no acceptable path and the original planning budget has not expired, circular fillet recovery is initiated from the original state. The ideal tangential setback distance for a turning angle $\Delta\theta$ is given by Eq. (7):

$$d_f = R_f \tan\left(\frac{|\Delta\theta|}{2}\right), \qquad R_f = 1.01R. \quad (7)$$

Recovery is rejected if adjacent segments are too short, an arc crosses occupied space, or the peak-curvature check fails. All retries share the original deadline; circular fillets do not reset the budget. Initial planning may try smaller radii with an additional speed limit, whereas replanning during flight must satisfy the current route and connection conditions. Geometric recovery thus expands the set of accepted candidates rather than forcing a path to be returned.

4.3 Time-Aligned Conflict Prediction

For a valid observation $\hat{p}_i, \hat{v}_i$, constant-velocity extrapolation gives $\hat{p}_i(t + \tau) = \hat{p}_i(t) + \hat{v}_i(t)\tau$. The nominal UAV trajectory does not estimate arrival times from arc length and a fixed flight speed. Instead, it is rolled out along each candidate route from the current five-dimensional estimated state using the shared integrator. Initial angular velocity, finite deceleration, and subsequent acceleration are therefore included in prediction. This remains a nominal tracking model and is not equivalent to the future closed-loop DWA trajectory.

The long-horizon prediction window is 45 s, with a normal refresh period of 1 s. Let r_k be the relative-position vector at a sample and Δr_k its increment to the next sample. The closest-approach fraction along the resulting chord is computed using Eq. (8):

$$u_k = \text{clip}\left(-\frac{r_k^T \Delta r_k}{\max(\|\Delta r_k\|^2, \varepsilon)}, 0, 1\right). \quad (8)$$

After subtracting the obstacle radius and an upper bound η_k on the UAV's turning deviation from the chord, if Eq. (9) holds,

$$\| r_k + u_k \Delta r_k \| - r_i - \eta_k < R_0 + \sigma t_{k+1}, \qquad R_0 = 25 \text{ m}, \quad \sigma = 0.5 \text{ m/s}, \tag{9}$$

the prediction interval is classified as conflicting. The deviation compensation is $\eta_k = \max(v_k, v_{k+1}) \max(|\omega_k|, |\omega_{k+1}|) \Delta t_k^2 / 8$. The inflation rate increases the margin for more distant predictions; it is not a probabilistic safety bound derived from an observation distribution. For each object, the closest-approach time in the first hazardous interval is retained, and objects are sorted by that time and then by ID. This time is not the analytic time of the first threshold crossing.

When the earliest predicted conflict is more than 15 s away, monitoring continues. At more than 5 s and at most 15 s, continuation, speed yielding, and detours are evaluated. At 5 s or less, no new waiting commitment is created, and detours and local avoidance are attempted. RDPHPP checks candidates against all valid observations; responding first to one object does not permit other traffic to be ignored.

4.4 Speed-Yielding and Detour Candidates

Let p_c be the conflict reference point. Solving Eq. (10)

$$\| \hat{p}_i + \hat{v}_i t - p_c \|^2 \leq (r_i + 30)^2 \tag{10}$$

gives the time window during which the object occupies the reference region. The candidate hold deadline is the exit time plus 4 s. No new yielding candidate is generated if the exit time is nonfinite, the hold deadline exceeds the prediction horizon, or the conflict is already urgent.

The yielding speed is determined from the finite-deceleration distance. With current speed v_0, target speed v_w, and hold duration T_h, the finite-deceleration travel distance is computed by Eq. (11):

$$\begin{aligned} t_b &= \min\{T_h, \max[0, (v_0 - v_w)/a_v]\}, \\ d(v_w, T_h) &= v_0 t_b - 1/2\, a_v t_b^2 + v_w (T_h - t_b). \end{aligned} \tag{11}$$

The largest possible v_w is sought subject to not exceeding the along-route distance to the conflict, with a lower bound of 8 m/s. If even the lowest candidate speed cannot yield in time, the candidate is infeasible. The waiting budget measures the travel-distance loss relative to cruise as an equivalent time: $L = \max(0, T_h - d/v_{\text{cruise}})$, subject to $L \leq B$. Initially, $B = 12$ s; this does not mean a maximum stationary waiting time of 12 s. During a commitment, its validity is continuously rechecked using the bound object ID and current observations. Budget exhaustion, target expiry, or trajectory infeasibility revokes the commitment.

A detour first attempts to rejoin the original route approximately 200 m ahead and, if unsuccessful, attempts a new search to the goal. Traffic trajectories near the predicted conflict are projected over the relevant time window into a two-dimensional swept mask that temporarily guides search by blocking space. This mask may be overly conservative and can be withdrawn for a retry if no solution is found, but building and exclusion-zone constraints are never relaxed. Any detour adopted by RDPHPP must still pass the dynamic conflict check based on Eq. (9) and the motion checks. This mechanism combines spatial search with temporal acceptance checks; it is not exact spatiotemporal A*.

Continuation, speed-yielding, and detour candidates all start from the same current state, use the acceleration-limited integrator in Eq. (3), and are checked step by step against the speed-regime motion envelope in Eq. (4). A spatially connected detour can still be rejected because of turning conditions, and a yielding maneuver that delays arrival can still be rejected because the deceleration distance is insufficient. Candidates that pass the common motion checks are then checked for static clearance and for dynamic conflicts using Eq. (9). Only candidates satisfying all checks are compared using Eq. (12),

$$S_c = 0.5 \frac{T_c}{T_0} + 0.5 \frac{E_c}{E_0} + 0.05 n_c \tag{12}$$

which combines estimated time T_c, modeled energy consumption E_c, and whether replanning occurs, $n_c \in \{0,1\}$. The common normalization scales T_0 and E_0 are obtained from the current remaining path. Time and energy beyond the prediction window are approximated from arc length; this extension provides no safety certificate for the unpredicted remainder. Equation (12) compares only a finite candidate set and does not imply optimal

actual energy consumption over the full mission. Continuing a valid waiting commitment does not require detours to be enumerated again at every update.

Figure 3 connects candidate comparison to execution. Continuation, yielding, and detours use the same observation snapshot and initial UAV state and undergo common checks before cost comparison. Yielding outputs a target speed, object ID, and hold deadline; a detour outputs a new reference. If adoption is delayed, both are revalidated against the state at adoption. A failed revalidation discards the proposal; an invalid existing commitment revokes its associated speed limit. The near-term conflict branch in the figure applies to conflicts more than 5 s and at most 15 s away. Algorithm 1 gives the urgent branch and handling of an empty candidate set.

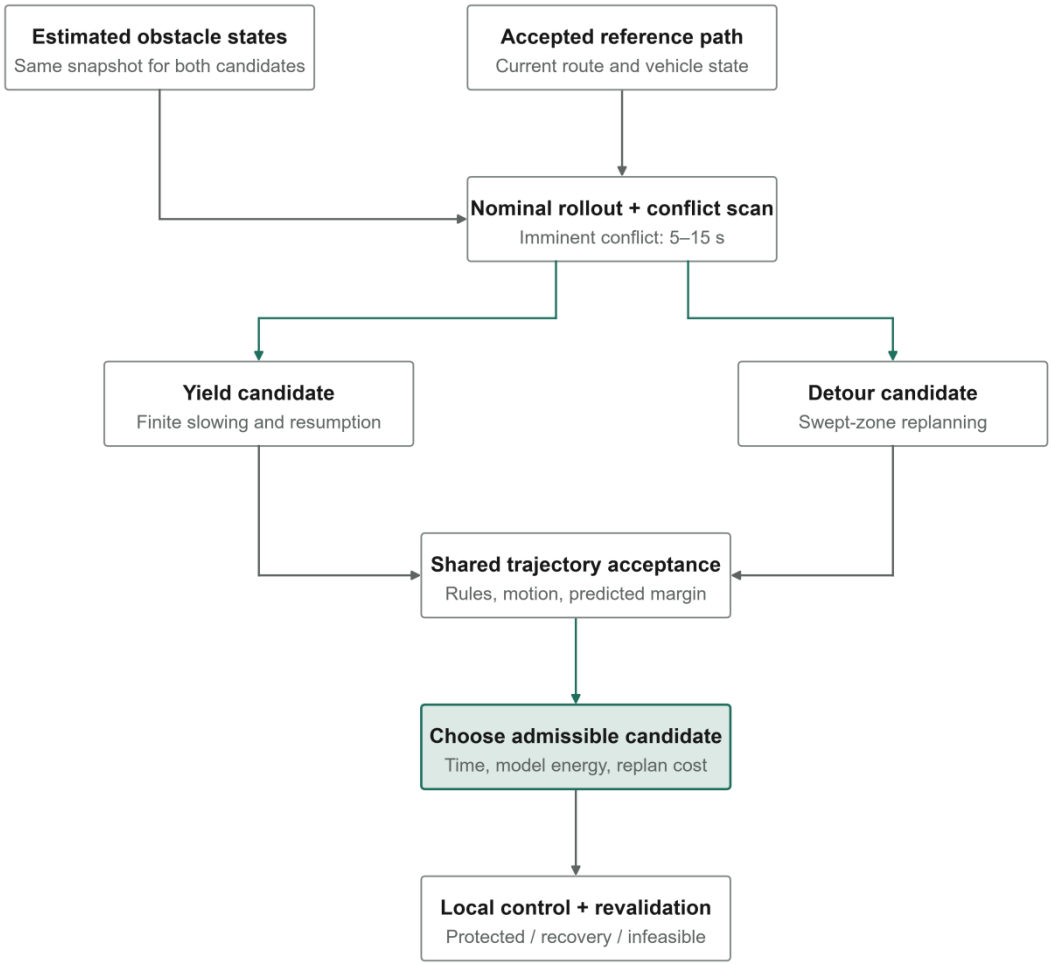


Figure 3. Candidate generation, common acceptance checks, and connection to execution for near-term conflicts in RDPHPP. Candidates enter cost comparison only after passing static, motion, and dynamic checks, and are revalidated against new observations during local execution. Algorithm 1 gives the full branches by urgency.

4.5 Speed-Regime DWA, Finite Braking, and Failure Handling

The execution layer combines DWA screening of reachable controls with the speed-regime motion envelope in Eq. (4). Local control is updated every 0.2 s, and the prediction integration step is 0.1 s. Normal candidates sample target speed and angular velocity, yielding at most 144 distinct targets after deduplication. All actual speed changes obey Eq. (3), and the angular-velocity limit for the corresponding speed regime is checked. The frozen parameters mainly cover the low-speed branch, as detailed in Section 3.2. Targets include zero speed, which represents finite braking. The prediction horizon is $H = \max(3.5, v/a_v + 0.2)$ s to cover the current braking timescale. Each normal candidate is rolled out with a constant target; a stopping trajectory is not appended to every rollout.

Candidates must satisfy a static UAV-center clearance of 9 m, a dynamic obstacle-edge-to-UAV-center distance of 25 m, and the motion envelope. Feasible targets are then scored by heading, clearance, progress, speed preference, and turning magnitude. Dynamic obstacles are extrapolated from current valid observations. The time-growing inflation term used for long-horizon prediction is not directly added to this fixed local threshold.

When the normal candidate set is empty, the system checks the complete finite-braking trajectory against a 13 m obstacle-edge-to-UAV-center threshold. Only when braking is also unsafe and observed dynamic traffic is present does it expand the target-speed set. It first searches with the normal threshold and, if the set is still empty, attempts conditional local recovery with the 13 m threshold, corresponding to 5 m of body-surface separation. Recovery candidates prioritize improving the worst predicted clearance within the horizon. If no solution remains, the system outputs a braking target and explicitly records infeasibility; this cannot be described as safe hovering. The minimum separation in actual execution can still fall below the prediction threshold.

Algorithm 1 gives the online RDPHPP procedure, including all candidate-comparison branches. Inputs are the frozen map, exclusion rules, start and goal, motion parameters, and observations at each time step. Outputs are the control sequence, mission termination state, and decision records. Search and smoothing retain distinct timeout, failure, and success states.

Algorithm 1. Online RDPHPP planning and execution.

Step 1: Initialize. Read the map, exclusion configuration, start and goal, and motion parameters. Compile the constraint geometry using Eq. (1), and perform direction-augmented search with Eqs. (5) and (6). Attempt normal smoothing and, if needed, circular fillet recovery sequentially within the 10 s initial deadline. On success, save the reference route and proceed to Step 2. If search finds no solution, times out, or smoothing remains unacceptable, record the corresponding state, return initialization failure, and terminate the mission.

Step 2: Update observations and revalidate adoption. At every simulation step, read the current observation snapshot, update targets by ID, and delete objects not observed within the last 1 s. For delayed yielding or detour proposals due for adoption, check the current position, route connection, target validity period, remaining budget, and current temporal feasibility. Adopt a proposal if it passes; otherwise discard it and record the reason. Proceed to Step 3 whether no proposal is due, adoption succeeds, or revalidation fails.

Step 3: Schedule prediction checks. If the 1 s prediction refresh is not due, retain the current valid high-level response and proceed to Step 7. If a refresh is due and a waiting commitment exists, revalidate its target, hold deadline, budget, and remaining waiting trajectory. If all remain valid, retain the commitment and proceed to Step 7. If no commitment exists, it expires normally, or any condition fails, release the associated speed limit, record the release or revocation reason, and proceed to Step 4.

Step 4: Detect conflicts. Use Eq. (3) to roll out the nominal route for up to 45 s or until arrival at the goal. Check all valid traffic using Eqs. (8) and (9), and sort conflicts by time and then ID. If no conflict exists or the earliest conflict is more than 15 s away, retain the route and proceed to Step 7. Otherwise, initialize the candidate set, independently assess continuation, and add it if accepted. If the earliest conflict is at most 5 s away, mark the situation as urgent and prohibit new waiting commitments. Both urgent and nonurgent branches proceed to Step 5.

Step 5: Generate yielding and detour candidates. If the conflict is nonurgent, waiting is permitted, and the budget is positive, solve Eq. (10) for the occupancy exit time and set the hold deadline to the exit time plus 4 s. Only if the deadline is within 45 s, use Eq. (11) and 24 bisection iterations to find the largest target speed satisfying the along-route condition, with a lower bound of 8 m/s and subject to the reference speed limit. Reject yielding if the minimum speed still fails the condition, the budget is insufficient, or trajectory revalidation fails; otherwise add it to the candidate set. Skip this branch for urgent conflicts or when waiting is prohibited. If more than 6 s has elapsed since the previous replanning attempt and no proposal awaits adoption, attempt a detour: first search for a rejoining point approximately 200 m ahead, then search to the goal if that fails. If temporary swept-region guidance prevents a solution, withdraw it and retry while retaining all building and exclusion constraints. Add a detour only if it passes search, smoothing, motion, and dynamic checks within

the shared 2 s budget; otherwise record the failure or timeout reason. Skip detours until the cooldown expires. After all applicable generation branches finish, proceed to Step 6.

Step 6: Compare and adopt. Retain only candidates that pass all checks and have finite costs under Eq. (12). If the set is empty, retain the current route, record that no high-level candidate is feasible, and proceed to Step 7. Otherwise, select the minimum-cost candidate, breaking ties first by the number of replanning operations and then by candidate name. For continuation, retain the route; for waiting, save the target ID, speed, and deadline; for a detour, adopt the new route. With zero adoption delay, all three outcomes proceed to Step 7. If adoption is delayed, queue the yielding or detour proposal and proceed to Step 7 for the current step. Actual adoption when due depends on revalidation in Step 2; stale acceptance labels cannot be adopted directly.

Step 7: Select local control. If the 0.2 s control refresh is not due, retain the previous control target and proceed to Step 8. Otherwise, construct the reachable control set, roll out candidates using Eqs. (3) and (4), and check static clearance and dynamic separation. If the normal set is nonempty, select an action using Eq. (C1) in Appendix C and proceed to Step 8. If it is empty, check finite braking against the 13 m dynamic obstacle-edge-to-UAV-center threshold; if accepted, adopt the braking target and proceed to Step 8. If braking is unsafe and valid observed traffic is present, expand the local recovery target set and screen it first with the 25 m threshold and then with the 13 m threshold. If a candidate passes, adopt the recovery target and proceed to Step 8. If braking and all applicable recovery attempts fail, output targets of zero speed and zero angular velocity, record local infeasibility, and proceed to Step 8. These zero targets are still executed with finite acceleration.

Step 8: Advance and check termination. Advance the ground-truth UAV and traffic states using a 0.1 s integration step. Independently compute collisions and rule violations between steps, then check arrival and the 240 s mission limit. On collision or rule violation, return the corresponding failure state and terminate. If neither failure occurs and the goal is reached, return success and terminate. If the goal has not been reached and the time limit has expired, return mission timeout and terminate. Otherwise, proceed to Step 2. Save all planning attempts, candidate rejection reasons, and control labels. Collision checks take precedence over arrival.

Step 3 defines high-level refresh branches, and Step 7 defines control refresh branches. Steps 2 and 8 continue between refreshes to maintain observation updates and ground-truth advancement. Initialization failure terminates immediately. An empty high-level candidate set leads to continued local-control checks, while local infeasibility retains an explicit label. This schedule allows the algorithm's inputs, branch outcomes, and loop transitions to be mapped directly to the implementation. Nominal reference rollouts and future closed-loop DWA trajectories are still evaluated separately.

5 Experimental Results and Analysis

5.1 Experimental Setup

5.1.1 Simulation Platform and Scenarios

The experiments used a two-dimensional kinematic simulator implemented in Python. NumPy handled array and trajectory calculations, SciPy supported distance transforms and connected-component processing, and Matplotlib provided plots, animations, and the interactive interface. The implementation covers map and rule assembly, traffic and observation generation, hierarchical planning, local control, logging, and independent ground-truth evaluation. The interactive program was used to inspect routes and events, whereas a separately managed batch program read presaved scenarios and produced paired results. The frozen computing environment comprised Windows, 16 logical CPUs, Python 3.11.9, NumPy 1.26.4, and SciPy 1.17.1; performance tasks ran in a single process (Appendix A). Figure 5 shows the actual interface, while all experimental statistics were calculated from the frozen batch logs.

Figure 4 shows the frozen Shenzhen and Guangzhou maps loaded by the simulator. Both are two-dimensional grids covering 2000×2000 m at a resolution of 5 m. The base maps were derived from OpenStreetMap [7] and include manually adjusted corridors and exclusion zones specified in the scenario configurations. The displayed

geometry is an experimental input rather than real-time urban airspace data; the configured exclusion zones do not represent verified real-world restrictions. Coordinates denote local positions in meters, and colors distinguish occupied structures from exclusion masks.

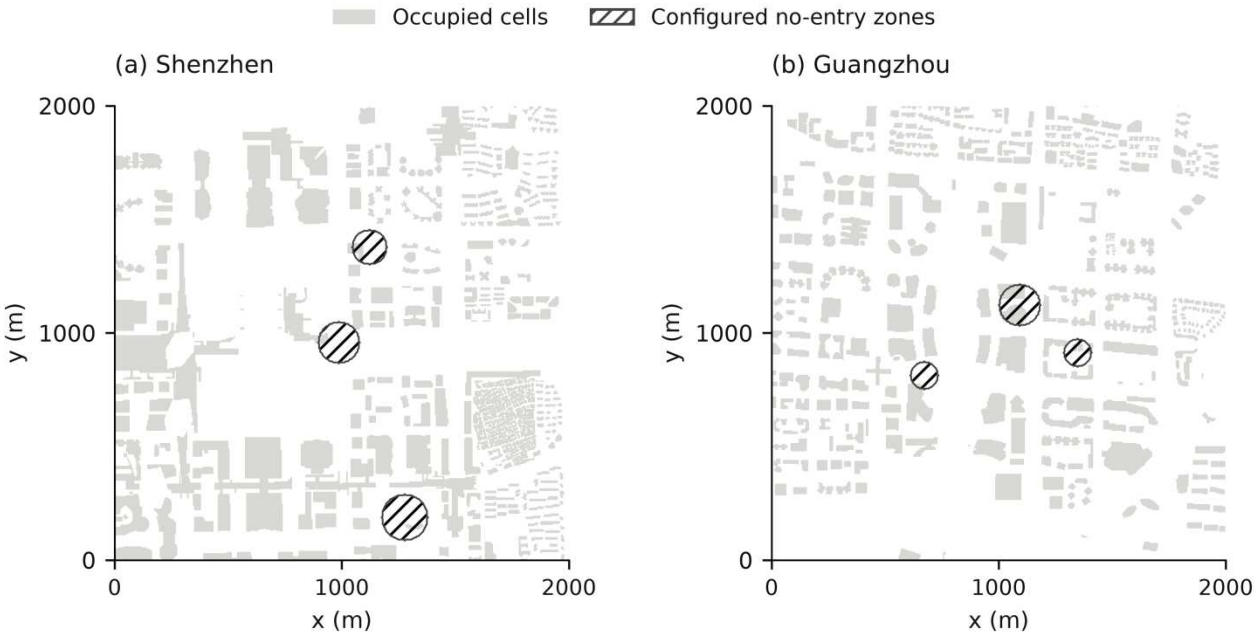
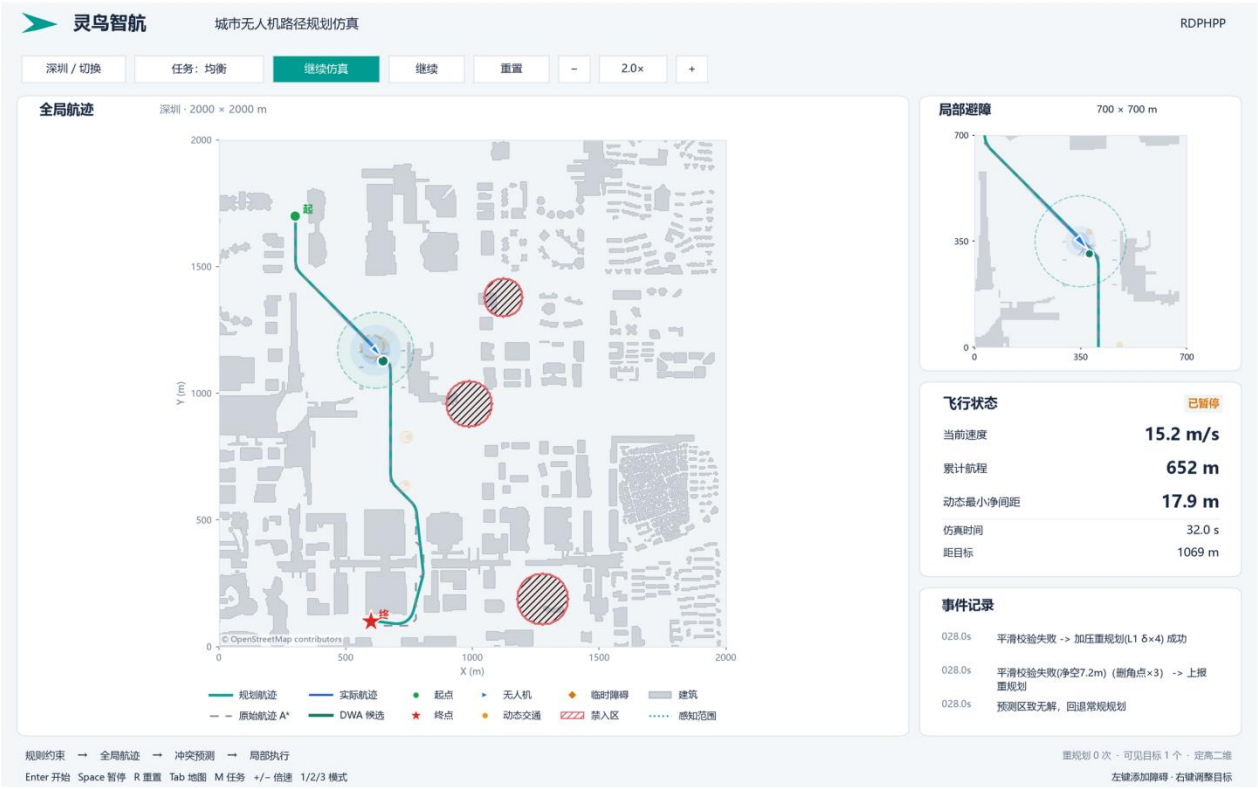


Figure 4. Frozen maps used in the formal experiments: (a) Shenzhen; (b) Guangzhou. Gray cells are occupied, hatched regions are configured exclusion masks, and white regions are traversable. The two masks may overlap. The figure was plotted directly from the map arrays in the formal run directory and includes manual scenario adjustments; it is neither satellite imagery nor a generative illustration.

Figure 5 presents the interactive simulation interface built with Matplotlib. The global view displays reference routes, exclusion zones, and executed trajectories, while the local view shows nearby traffic. Flight states and event logs support inspection of smoothing rejection, replanning, and response switching. The screenshot comes from a saved demonstration and illustrates how the platform is inspected. All five experimental groups use frozen scenarios and raw logs; demonstration readings from the interface are excluded from the statistics.

(a) Actual simulation interface



(b) Flight status



(c) Event log



Figure 5. Actual simulation interface and enlarged views: (a) global and local views; (b) flight state; (c) event log. The interface uses the project's urban maps. The screenshot illustrates the interactive tool and is not a statistical sample from the frozen experiments. The original interface is localized in Chinese. The enlarged status panel reports flight mode, speed, remaining distance, lateral acceleration, elapsed time, and altitude; the adjacent panel shows the event log. The screenshot is retained as recorded rather than presented as an English-language interface.

The main dynamic matrix crossed two maps, 1/5/10 obstacles, and three conflict types—crossing, converging, and overtaking—with 30 scenarios per combination, yielding 540 scenarios. All methods received identical maps, start–goal pairs, traffic motion, and noise streams. Because directed encounters facilitate control of conflict conditions, an additional 30 independent traffic scenarios per map were generated without positioning traffic relative to the proposed method's route, to assess dependence on encounter generation. Scenario validity was determined independently; samples were not selected according to whether RDPHPP succeeded.

The urban experiments followed a single frozen protocol containing 1110 unique scenarios and 4710 method tasks. Scenarios were reused across the rule, static, dynamic, mechanism, and degradation comparisons, so method-task counts and independent-scenario counts are reported separately. Development, calibration, and test

inputs were saved separately under the original protocol, and parameters were locked before the formal test scenarios were prepared. Both cities had been used during system development; no external test on an unseen map was conducted. An archived implementation was rerun with identical inputs and shared parameters for implementation-version comparison; it differs from the Legacy comparison implemented in the current engine. The fixed-input circular-fillet switch analysis was a post hoc mechanism diagnostic, saved separately without altering the original main matrix. Appendix A details the audit protocol. Section 5.5 uses a separate controlled-request protocol saved subsequently, comprising 252 physical scenarios and 2268 execution records. It tests the shared finite-motion model and protection conditions and is not pooled with the original urban protocol to calculate completion rates. The execution implementations for the two protocols are archived separately, and task performance is not paired across versions.

5.1.2 Comparison Methods and Shared Parameters

The proposed method is consistently termed RDPHPP. In the static comparison, the "RDPHPP global module" denotes the proposed direction-augmented multi-cost search followed by smoothing and acceptance checking. The dynamic comparisons include Reactive (reactive A* + DWA), which disables long-horizon predictive arbitration while retaining local trajectory rollout; No-wait, which retains predictive replanning but disables waiting; Legacy, which adapts the earlier decision logic to the current engine; and GVO-adapt, a nonreciprocal GVO adaptation operating over the shared reachable control set [23]. All methods share the applicable motion envelope, local braking, and recovery conditions.

GVO-adapt selects a target whose terminal velocity is close to the desired direction according to Eq. (C2) in Appendix C, then rechecks the entire candidate trajectory using the shared integrator. Its sampling, local goals, and coordination with global planning are defined by this platform [23]. The static comparisons use project implementations of conventional distance-based A* and RRT*, an asymptotically optimal variant of rapidly-exploring random trees [8, 11]. The RDPHPP global module in Table 4 comprises the direction-augmented multi-cost search, smoothing, and acceptance checks. The module differences defined here delimit all subsequent comparisons.

Table 2 relates the five experimental groups to the research questions in the introduction. Rule replacement and static acceptance checks test whether the admissible space yields usable references. Dynamic-task and efficiency experiments assess task completion and its costs after reference adoption. Controlled execution directly checks whether the conditions predicted under finite deceleration are realized. Degradation experiments examine the limits of this connection under motion errors and delays. Seven mechanism ablations are retained as supplementary analyses in Appendix E. These five groups provide a post hoc organization of existing results; the subsequent controlled diagnostic and the original urban tests are reported separately.

Table 2. Objectives, content, and scale of the five experimental groups.

Experiment and section	Objective	Design and comparison scale
I: Rules and static paths, Section 5.2	Test rule propagation and geometric acceptance of references	60 A/B rule pairs; 60 start–goal pairs × 3 search methods; 60 additional fixed-input circular-fillet switch diagnostics
II: Dynamic tasks, Section 5.3	Compare the complete workflow and distinguish the effects of long-horizon prediction, waiting, and version changes	540 main scenarios × 5 methods; 60 independent traffic scenarios × 5 methods; separately paired archived-version comparison
III: Efficiency and separation, Section 5.4	Assess time, distance, model energy, and surface-to-surface separation in addition to completion rate	Reuse joint-success pairs from the main experiment; summarize all 990 successful RDPHPP dynamic runs
IV: Time alignment and execution, Section 5.5	Check finite-deceleration decisions and the effect of the state at adoption	252 physical scenarios × 2 target speeds; 3 adoption delays; 2268 execution records
V: Degradation and overhead, Section 5.6	Test the effects of motion errors, delays, and computational timeliness	5 conditions × 90 scenarios × 2 methods; 48 time-step comparison pairs; timing of individual planning and control calls

Table 3 summarizes the shared parameters and their roles. Motion, sensing, and power parameters are simulation settings and have not been calibrated on a physical composite-wing UAV platform. Sections 3.2 and 4.2 define the speed-dependent motion envelope and curvature acceptance conditions, respectively.

Table 3. Main parameters of the frozen experiments.

Category	Parameter and value	Role
Map and vehicle	Area: 2000 × 2000 m; grid resolution: 5 m; vehicle radius: 8 m	Occupancy representation and independent swept-collision evaluation
Motion	Maximum speed: 33.3 m/s; acceleration limit: 6 m/s²; angular acceleration limit: 1 rad/s²; angular velocity limit: 1 rad/s	Shared integrator for prediction and execution
Path	Nominal radius: 45 m; curvature tolerance: 1.05; static clearance to the UAV center: 12 m	Global smoothing acceptance
Search	Distance coefficient: 1.35; turning coefficient: 20; obstacle-proximity coefficient: 1.2; obstacle-proximity scale: 40 m	Surrogate cost in Eq. (5)
Prediction	Horizon: 45 s; update interval: 1 s; emergency/response thresholds: 5/15 s	Conflict detection and candidate generation
Prediction protection	Base obstacle-edge-to-UAV-center distance: 25 m; margin growth rate: 0.5 m/s; long-horizon static clearance to the UAV center: 10 m	Acceptance of time-aligned candidates
Speed yielding	Minimum target speed: 8 m/s; loss budget: 12 s; exit buffer: 4 s	Finite-deceleration yielding and commitment revocation
Sensing	Range: 150 m; update rate: 10 Hz; observation validity after loss: 1 s	ID-based snapshot updates and bounded extrapolation
Local control	Update interval: 0.2 s; normal dynamic obstacle-edge-to-UAV-center threshold: 25 m; conditional recovery threshold: 13 m; static clearance to the UAV center: 9 m	Reachable controls, braking, and recovery
Simulation and termination	Integration step: 0.1 s; arrival radius: 25 m; task time limit: 240 s	Ground-truth propagation and task evaluation
Planning budget	Initialization: 10 s; en-route replanning: 2 s	Shared deadline covering smoothing and recovery

Nominal observation standard deviations were 1 m per coordinate for ego position, 0.5° for heading, 0.5 m per coordinate for traffic position, and 0.2 m/s per coordinate for traffic velocity. The degradation conditions are specified in Experiment V so that the nominal and robustness settings can be inspected separately.

5.1.3 Metrics, Statistics, and Run Procedure

The full-sample task success rate is the proportion of runs that reached the goal without triggering a failure. Collision rates, rule violations, and motion-constraint violations are recorded separately. Minimum dynamic surface-to-surface separation is the minimum value of Eq. (2) over the complete ground-truth execution. Proportions are reported with Wilson 95% intervals [24]. Binary comparisons use the exact paired McNemar test [15], with separate Holm corrections for the prespecified primary, mechanism, and robustness comparison families [10], at a significance level of 0.05. Stratified results by map, density, and conflict type are descriptive only.

Completion time, travel distance, and model energy are compared in joint-success pairs, with the sample size reported separately for each comparator. Continuous outcomes use two-sided Wilcoxon signed-rank tests. Zero differences are excluded from the rank sums; if all paired differences are zero, the test returns $p = 1$. For at most 30 pairs, a permutation procedure with a fixed random stream is used; larger samples use the asymptotic calculation with continuity correction. Effect sizes are reported as rank-biserial correlations, and bootstrap 95% intervals for mean differences are obtained from 9999 paired resamples. Mean-difference intervals and rank tests address different quantities, so conclusions follow the prespecified tests and corrections; nonsignificance does not establish equivalence. Full statistics and resource use before failure are reported in Tables D1 and D2 of Appendix D, respectively.

Model energy is integrated as $E = \sum_k P(\bar{v}_k, \kappa_k, a_{t,k})\Delta t_k$, using the power model in Eq. (B1) of Appendix B. This metric comes from a shared assumed model; it is neither measured battery consumption nor the A* surrogate cost. Early termination of failed tasks cannot be counted as a time or energy benefit.

Each run first validates the frozen scenario and parameters, after which every method reads the same inputs. The planner consumes observation snapshots, while the ground-truth engine propagates traffic and independently checks collisions, violations, arrival, and timeout. Trajectories, failure reasons, timings of all planning attempts, and decision events are then saved and summarized in pairs by scenario ID. Control periods are prescribed by the simulation scheduler, and single-process solver wall times are recorded separately; the environment does not advance automatically while a solver function runs. In the fixed decision-delay experiments, actions are queued until specified simulation times, traffic continues to move during the delay, and actions are revalidated when due. This setup does not embed actual CPU computation latency in the feedback loop.

5.2 Experiment I: Rule Replacement and Static Path Executability

Dynamic responses first require references that comply with the current exclusion rules and accommodate constrained turns. Experiment I checks rule propagation and reference construction separately. For rule replacement, the maps and 60 start–goal pairs are fixed, configuration A is replaced with configuration B containing additional exclusion zones, and search and cost parameters remain unchanged; compliance with the exclusions is then checked independently. The static comparison runs distance-based A*, the RDPHPP global module, and RRT* on 30 start–goal pairs per map, separately counting raw connections and references that pass common curvature and clearance checks. The fixed-input circular-fillet diagnostic holds the raw polyline, map, and thresholds constant and changes only the recovery switch, isolating the effect of the alternative connection.

Start–goal pairs were generated under the frozen protocol independently of the planners being compared. Free cells were first selected using a distance field corrected by subtracting half the grid-cell diagonal, requiring at least 15 m clearance. A 30 m band along the map boundary was excluded, and the largest four-connected component was retained. A start was sampled from cell centers with at least 40 m clearance, and a goal was sampled from candidates at least 900 m from that start. The random stream was determined jointly by the test split, seed, map, and experiment type. The formal test used a base seed of 930000 and 30 pairs per map. An independent breadth-first connection generated the reference geometry; samples were not filtered according to the success of any compared method. This strategy ensures geometric connectivity and long-range inputs within the selected region, but does not guarantee coverage of all small connected components or levels of turning difficulty. Full coordinates and seeds are saved with the experimental data.

After rule replacement, 14 of the 60 input pairs required rerouting, 4 became infeasible, and 42 retained their original outcome. All 116 successful paths across the two configurations passed independent swept-path checks against the exclusion regions. The added exclusion zones reduce the traversable space, so rerouting and rejection are the expected consequences of constraint propagation. This experiment directly supports the rule interface: changing the configuration changes planning outputs without changing search expansion logic, and acceptance checks respect the updated feasible domain.

Table 4 shows that distance-based A* and RDPHPP both found 60 raw connections, but only 30 and 57, respectively, passed final smoothing acceptance. RRT* found 59 connections, of which 46 passed. Raw search success was similar across the three methods, whereas the proportion of acceptable references differed substantially. Under motion constraints, connectivity and reference executability are therefore distinct evaluation levels, and subsequent comparisons should use references that pass geometric acceptance.

Table 4. Static search and smoothing acceptance for 60 start–goal pairs per method.

Method	Raw search successes	Smoothing acceptance successes	Acceptance rate
Distance-based A*	60	30	50.00%
RDPHPP global module (proposed)	60	57	95.00%
RRT*	59	46	76.67%

On identical inputs, the archived implementation accepted 40/60 static references, compared with 57/60 for the current RDPHPP. The smoothed coordinates of the 40 jointly successful references were identical point by point. This version diagnostic is retained in Figure 6(b) and Appendix F. To isolate the circular-fillet effect further, the fixed-input switch diagnostic held the raw paths and thresholds constant: acceptance increased from 40/60 with recovery disabled to 57/60 with it enabled. There were 17 references that succeeded only when recovery was enabled, and the enabled outputs reproduced the successful path coordinates of the formal version. Figure 6(b) thus provides more direct evidence of geometric recovery than the overall version comparison.

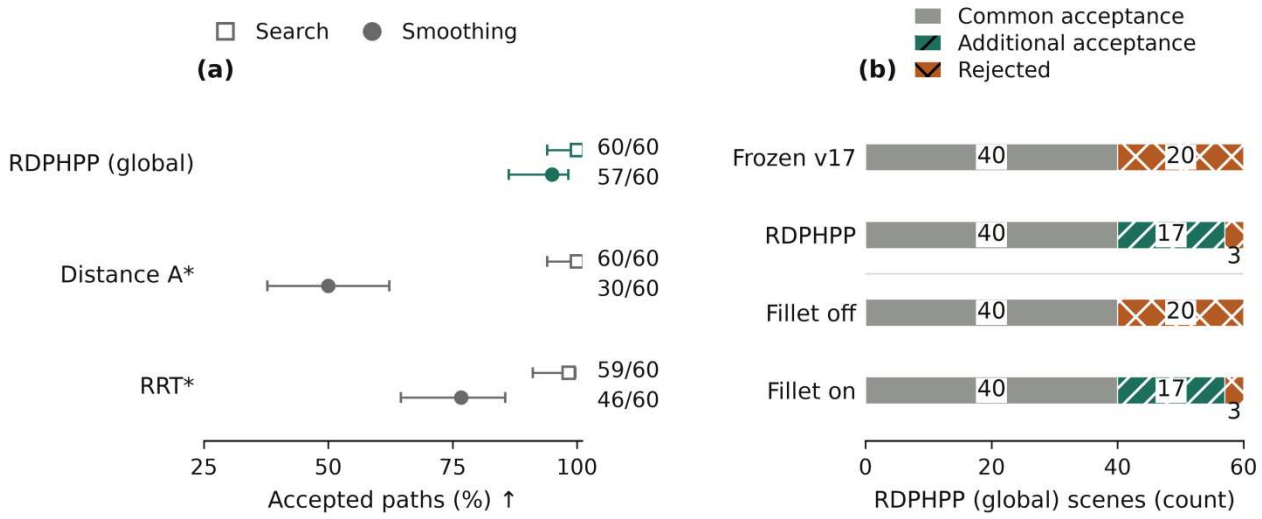


Figure 6. Static path generation and acceptance. Search and smoothing success use the same 60 start–goal pairs as their denominator. The archived-implementation comparison and fixed-input switch diagnostic are identified separately. The latter is a post hoc mechanism analysis.

The acceptance gains in Figure 6 arise from geometric preparation during search and alternative candidates during connection. Distance-based A* prioritizes shorter polylines, which can leave insufficient tangent setback near corners and still cause rejection for peak curvature or clearance after smoothing. RDPHPP retains distinct incoming directions at the same grid cell. Sharp-turn costs can change the approach direction, while obstacle-proximity costs encourage paths away from occupied boundaries, leaving room for curved connections. If the standard connection fails, circular-fillet recovery attempts an alternative corner construction under the same thresholds. Accepting 57 of 60 connections, compared with 30 for distance-based A*, indicates that selecting paths with subsequent connection in mind is better suited to these constrained-reference tasks than connectivity alone. The 17 additional acceptances with fixed polylines further support the independent geometric role of the circular fillet. The overall search and postprocessing differences were not isolated for each cost coefficient, and the remaining 3 rejections show that connection space and the shared time budget still limit acceptance.

Rule replacement verifies that the admissible space follows the configuration, whereas the static comparison and fixed-input diagnostic verify an increase in acceptable references. These provide separate evidence for the system interface and geometric construction. The following closed-loop experiments test whether tasks can be completed and what actual separation is achieved after dynamic control is introduced.

5.3 Experiment II: Main Dynamic Tasks and Independent Traffic

Even with acceptable reference geometry, moving traffic can alter feasible execution timing. Experiment II compares RDPHPP, Reactive, No-wait, Legacy, and GVO-adapt on 540 main scenarios. Each method receives identical maps, start–goal pairs, traffic motion, and observations, and collisions, timeouts, and initial planning failures are retained. Reactive disables long-horizon predictive arbitration, and No-wait disables waiting, allowing the incremental effects of these responses to be examined. A further 60 traffic scenarios, generated without directed placement relative to the ego reference, test the effect of encounter generation. The overall comparison with the archived implementation is reported in Appendix F, with its raw results and trajectory cases retained.

Table 5 reports task outcomes for the five methods, and Figure 7 retains both the main comparison and clearly marked supplementary version data. RDPHPP completed 515/540 tasks, yielding a success rate of 95.37% with a Wilson 95% interval of 93.26%–96.84%; its 25 collisions correspond to 4.63%. Success was 15.00 percentage points higher than for GVO-adapt, with a significant paired difference after correction. Differences from Reactive and Legacy were not significant. No-wait and RDPHPP had identical binary outcomes in every scenario. The complete workflow thus performed better in this GVO-adapt comparison, but completion rates did not establish an independent benefit of waiting or long-horizon prediction over their corresponding ablations.

Table 5. Main dynamic comparison on 540 shared scenarios per method. The p-values retain the frozen joint Holm correction for five comparisons: the four comparisons in the main table and the archived-version comparison in Appendix F belong to the original test family. Moving a comparison to the appendix does not reduce that family.

Method	Successes	Success rate	Collisions	Collision rate	Adjusted p
RDPHPP	515	95.37%	25	4.63%	—
No-wait	515	95.37%	25	4.63%	1.000
Reactive	506	93.70%	34	6.30%	0.235
Legacy	508	94.07%	32	5.93%	0.459
GVO-adapt	434	80.37%	65	12.04%	6.18×10^{-15}

All five current methods recorded zero motion-constraint violations. The independent rule check found 1 violation for GVO-adapt and none for the other four methods. Rule violations and physical collisions are recorded separately, so success and collision rates need not sum to 100%.

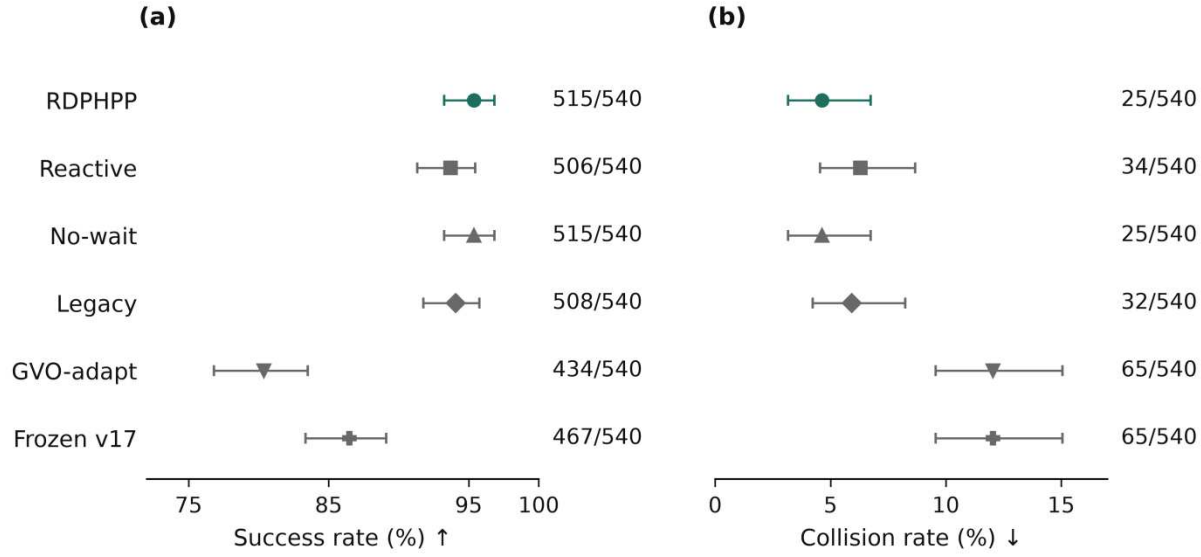


Figure 7. Main dynamic success and collision rates, with 540 scenarios per method and Wilson 95% intervals. The first five configurations form the main comparison. Frozen v17 is the supplementary version

diagnostic in Appendix F; the original figure is retained for a complete record, and its results are interpreted separately from the current Legacy adaptation.

Appendix F reports the archived implementation's complete success, collision, and bidirectional paired results. Figures 7 and 8 retain its data and cases of improvement and degradation to support review of implementation changes. Mechanism analysis for the main method instead relies on the specified switch experiments and the controlled diagnostic in Section 5.5.

Stratified results show 252/270 successes in Shenzhen and 263/270 in Guangzhou; 177/180, 173/180, and 165/180 with 1, 5, and 10 obstacles; and 177/180, 172/180, and 166/180 for crossing, converging, and overtaking encounters. More failures remained at higher densities and in overtaking encounters, but these differences are descriptive. In the independent traffic set, RDPHPP, Reactive, No-wait, and Legacy each achieved 59/60, whereas GVO-adapt achieved 48/60. RDPHPP's performance therefore does not depend entirely on directed encounter generation in the main matrix. However, the independent traffic sample is smaller and produced no arbitration events in which waiting and detouring were both feasible, limiting its mechanism coverage.

GVO-adapt and RDPHPP organize decisions differently over the same reachable actions and revalidation conditions. GVO-adapt evaluates local targets by terminal-velocity proximity to the desired direction. RDPHPP additionally compares spatial detours and finite-deceleration responses using longer-horizon rollouts, adjusting routes or passage times in advance. This design difference is consistent with RDPHPP's higher completion count and indicates an overall benefit of broadening the response set for these encounters. Because global coordination and action evaluation change together, the specific source of the benefit requires switch diagnostics. The nonsignificant completion-rate difference from Reactive also suggests that local responses may already resolve most tasks. Experiment IV directly checks whether finite-deceleration decisions agree with execution.

Success counts decreased from 177 to 173 and 165 as density increased from 1 to 5 and 10 obstacles. A plausible explanation is that increasing occupancy leaves fewer candidates satisfying both geometric and dynamic conditions. The 166/180 completion count in overtaking encounters was lower than 177/180 for crossing and 172/180 for converging encounters; sustained relative motion may reduce opportunities for speed yielding. These explanations concern the design and descriptive strata; the specific causal chains of failures still require trajectory-by-trajectory inspection. Identical outcomes for No-wait and RDPHPP also need to be interpreted alongside response coverage: enabling waiting may not change task endpoints when detours and local control already resolve most solvable conflicts, or when both high-level options are rarely available together.

Following the previously disclosed figure-selection rule, Figure 8 uses the smallest scenario ID from each of three sets: successes unique to RDPHPP, successes unique to the archived implementation, and static recovery successes. These illustrate improvement, degradation, and geometric recovery. Retaining version differences in both directions explains why implementation changes did not produce monotonic improvement. All 25 RDPHPP collision tasks recorded local infeasibility at termination, indicating that no action satisfying the current predictive conditions was found in the final state. This label preserves the failure exit, but the terminal record alone cannot attribute a collision to an earlier planning step; that requires the complete observation, candidate, and control history.

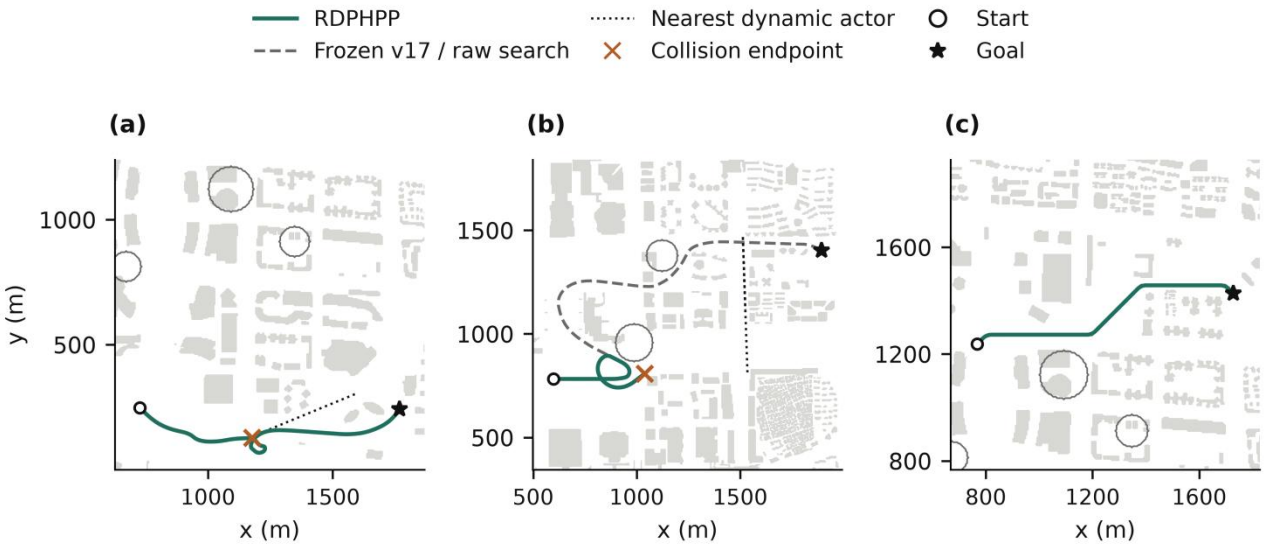


Figure 8. Improvement, degradation, and static recovery cases from stored trajectories. Each case is selected by the smallest scenario ID in its outcome set; set sizes are 50, 2, and 17, respectively. Light-gray fill denotes occupied regions, and thin gray contours denote exclusion boundaries. Circles and stars mark starts and goals; crosses mark collision termination locations. Dashed lines show archived execution trajectories in (a, b) and the raw search polyline rejected by the earlier smoothing procedure in (c). Dotted lines show the dynamic object trajectory selected as nearest at archived termination; they do not imply simultaneous arrival of the two trajectories at arbitrary locations. All trajectories come from raw records.

The original urban protocol contained 52 events in which yielding and detouring were both feasible: yielding was selected 4 times and detouring 48 times. These events include rolling checks and repeated tasks. This response coverage provides one possible explanation for the identical task outcomes of No-wait and RDPHPP. Complete mechanism switches and event counts are reported in Appendix E.

5.4 Experiment III: Completion Efficiency and Actual Surface-to-Surface Separation

Task completion must be assessed alongside the resources required to reach the goal and the actual separation from obstacles. The efficiency comparison reuses the logs of Experiment II. For each comparator, completion time, travel distance, and model energy are compared only in joint-success pairs, preventing early termination of failures from being counted as time savings. Separation is evaluated independently from 1050 RDPHPP ground-truth runs spanning the main dynamic set, independent traffic, and five degradation conditions, using the minimum dynamic surface-to-surface separation of the 990 successful runs. These runs include scenario reuse: 990 is a successful-run count, not a count of independent encounters.

Table 6 uses joint-success pairs, with differences calculated as RDPHPP minus the comparator; positive values indicate greater resource use. Relative to Reactive, mean travel distance increased by 35.68 m, with a bootstrap 95% interval for the mean difference of [23.85, 49.39] m and an adjusted $p = 6.76 \times 10^{-11}$. Completion time and model energy did not differ significantly. Long-horizon planning can change routes in advance and can therefore incur additional detour distance. The overall completion results in Experiment II must be assessed alongside this distance cost; success rate alone does not characterize efficiency.

Table 6. Mean paired differences in joint-success tasks. Energy is model-estimated. Significance follows the frozen multiplicity-adjusted tests for each metric, rather than the magnitude of the means.

Comparator	Paired n	Time difference / s	Distance difference / m	Model energy difference / kJ	Adjusted test conclusion
Reactive	500	−0.54	+35.68	+1.93	Only distance differed significantly
No-wait	515	−0.24	−7.41	−1.52	None of the three differences was significant
Legacy	499	−0.70	+13.24	+0.13	None of the three differences was significant
GVO-adapt	419	+21.14	−74.01	+160.14	All three differences were significant

Relative to GVO-adapt, RDPHPP traveled 74.01 m less on average but required 21.14 s more and consumed an additional 160.14 kJ of model energy; all three differences were significant. A shorter route need not reduce flight time because operating speed, deceleration, and recovery also determine duration. The low-speed power branch in Appendix B has a higher base term, so longer low-speed operation may offset distance savings. This explanation follows the structure of the power model; the logs have not yet isolated low-speed duration and turning expenditure segment by segment. Relative to No-wait, adjusted tests for time, distance, and model energy were all nonsignificant, indicating no consistent efficiency benefit from the current waiting configuration. Supplementary efficiency statistics for the archived version are reported in Appendix F.

Of the 1050 RDPHPP dynamic runs, 990 completed their tasks. Across successful runs, minimum surface-to-surface separation had a median of 38.44 m and a 5th percentile of 8.11 m; 11 runs fell below 5 m and 1 below 1 m, with a minimum of 0.538 m. Most successful trajectories maintained substantial separation, while the lower tail approached the physical collision boundary. Figure 9 displays both the distribution and extreme values so that completion rate and actual protection can be assessed separately.

(a)

Comparator	Pairs, n	Distance (m) ↓	Time (s) ↓	Model energy (kJ) ↓
Reactive	500	1548.7 / 1513.1 p = 6.8e-11	85.0 / 85.5 p = 1	363.8 / 361.9 p = 0.257
No-wait	515	1558.3 / 1565.7 p = 0.650	86.4 / 86.6 p = 1	369.7 / 371.2 p = 1
Legacy	499	1553.3 / 1540.1 p = 1	84.9 / 85.6 p = 0.635	363.9 / 363.8 p = 1
GVO-adapt	419	1558.8 / 1632.8 p = 9.5e-24	85.2 / 64.0 p = 8.8e-52	365.2 / 205.0 p = 4.0e-63

RDPHPP / comparator — Holm-adjusted paired p below each mean pair

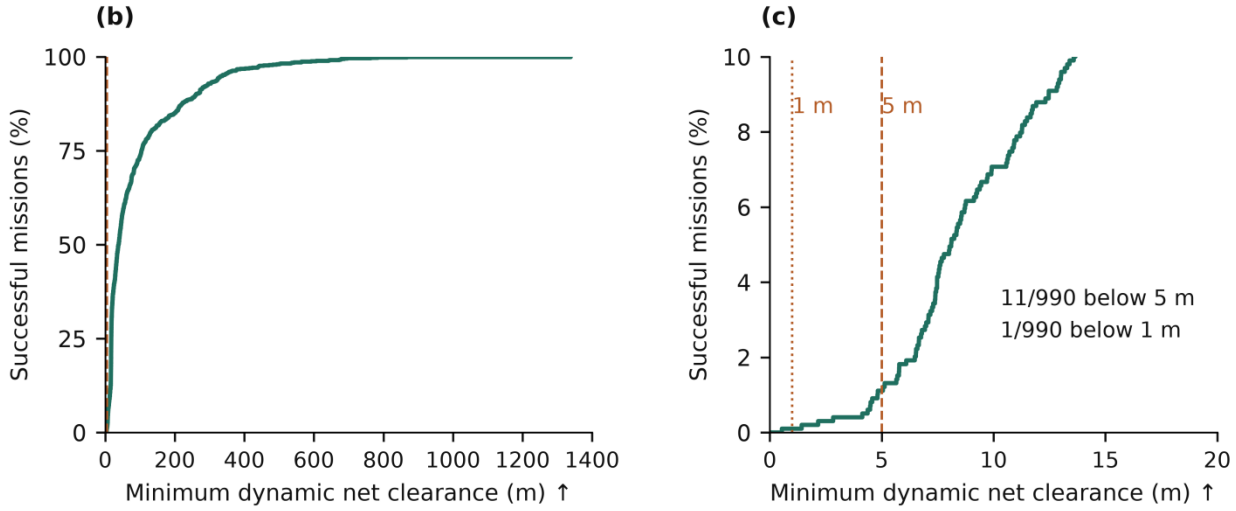


Figure 9. Completion-efficiency costs and separation distribution in successful tasks. Efficiency uses joint-success pairs, with sample sizes indicated. The safety distribution covers all 990 successful RDPHPP dynamic runs. Normal and recovery protection thresholds are prediction conditions; actual surface-to-surface separation must be calculated independently from ground-truth trajectories.

Figure 9 shows that response benefits and costs depend on the comparator: distance increases relative to Reactive, whereas relative to GVO-adapt it decreases at the cost of additional time and model energy. Higher completion rates therefore require interpretation across multiple metrics. The minimum surface-to-surface separation of 0.538 m in successful runs shows that predictive thresholds did not become a uniform lower bound on actual trajectories. Nominal reference rollouts, DWA actions, noisy observations, and subsequent recovery can produce different approach histories. Reference and candidate acceptance establish conditions for execution, while ground-truth separation tests their closed-loop consequences.

5.5 Experiment IV: Time-Aligned Candidates and Actual Execution

The main dynamic comparison changes references, decisions, and local execution together, so completion rates alone cannot establish the accuracy of finite-deceleration prediction. To isolate this timing question, this section uses a completed controlled-request diagnostic. High-level arbitration is disabled, prescribed yielding requests are injected, and the archived Simulator and DWA execute them with finite acceleration while retaining local vetoes, recovery, and route changes. The finite model and instantaneous approximation predict the same requests and are compared with the saved execution trajectories. Prediction labels do not gate request injection, avoiding execution restricted to self-accepted candidates. This diagnostic is a subsequent mechanism validation, with inputs and implementation archived separately from the main urban matrix.

The controlled reference is a 1500 m straight line with no static obstacles and perfect observations; both disks have a radius of 8 m. Initial speeds of 12, 24, and 30 m/s, target speeds of 8 and 12 m/s, crossing-point distances of 60, 120, and 220 m, crossing times of 1–14 s, and traffic speeds of 6 and 10 m/s define 504 conditions: two target speeds for each of 252 physical scenarios. Both predictions examine the same 14 s window. The finite model decelerates from the true initial speed according to Eq. (3), whereas the instantaneous approximation sets the predicted initial speed directly to the target speed. Actual Simulator and DWA execution both use finite acceleration.

Execution checks assess both a fixed 17 m surface-to-surface separation and a protection margin growing at 0.5 m/s. The fixed separation corresponds to the $25 - r_a = 17$ m convention in Eq. (9). Table 7 separately records violations of the growing separation margin, violations of the fixed separation, and physical collisions, distinguishing candidate commitments from physical outcomes. The 17 m separation and 0.5 m/s growth rate are design conditions for this diagnostic; their appropriate magnitudes are not independently calibrated by this comparison.

Table 7. Prediction acceptance and actual execution with zero adoption delay, across 504 conditions.

Metric	Finite-deceleration prediction	Instantaneous speed-reduction approximation
Predicted acceptances	256	264
Growing separation-margin violations after acceptance	0	27
Fixed 17 m separation violations after acceptance	0	1
Physical collisions after acceptance	0	0
Accepted, with growing margin and original request maintained through 14 s	256	237

With zero adoption delay, all 256 conditions accepted by the finite model maintained the growing separation margin. The instantaneous approximation accepted 264 conditions, of which 27 violated that margin across 25 physical scenarios. The acceptance sets differed in membership as well as size: 31 conditions were accepted only by the instantaneous approximation, and 27 of these failed during execution; 23 were accepted only by the finite model, and all maintained the original conditions. Table 7 therefore reveals both false acceptances and additional rejections that the instantaneous approximation can cause, supporting inclusion of the actual deceleration process in candidate evaluation. The 504 conditions comprise two target speeds for each of 252 physical inputs. Counts are descriptive by condition and are not treated as 504 independent physical scenarios for significance inference.

Under finite deceleration, the ego UAV continues to advance at higher speed before reaching its target speed. For example, decelerating from 30 to 8 m/s with an acceleration limit of 6 m/s² takes approximately 3.67 s and adds about 40.33 m of travel relative to an instantaneous reduction, enough to change the location and timing of a crossing encounter. Replacing the initial speed instantaneously omits this motion and can cause both false acceptances and additional rejections of otherwise valid timing. Of the 27 growing-margin violations in Table 7, only 1 also violated the fixed 17 m separation, and no physical collisions occurred. The directly demonstrated benefit is therefore consistency between acceptance conditions and actual motion.

Actual control does not always maintain the prescribed yielding request. Of the 504 zero-delay executions, 27 exited the original request before the check window ended: 15 because of route changes and 12 because of local vetoes. Mapping each predictor's decisions to “yield if accepted, continue if rejected” produced task completion for all 504 conditions under either predictor. This mapping reuses existing execution trajectories; it adds no independent runs and is not a rerun of the complete high-level policy. Local adjustments can change the consequences of an original request, so different counts of candidate-condition violations are compatible with identical final completion counts.

The $t = 0$ prediction labels were then retained while adoption was delayed by 0.2 or 1 s. Each delay setting contains 504 yielding executions and 252 shared continuation executions. Zero-delay runs cover complete tasks, whereas the two delayed settings check only the first 14 s, yielding 2268 execution records in total. As

Figure 10 shows, the numbers of growing-margin violations among conditions originally accepted by the finite model were 0, 1, and 11; the corresponding counts for the instantaneous approximation were 27, 28, and 36. Reaching the end of a delayed check window is not counted as a task timeout, and these counts are not treated as independent urban task samples.

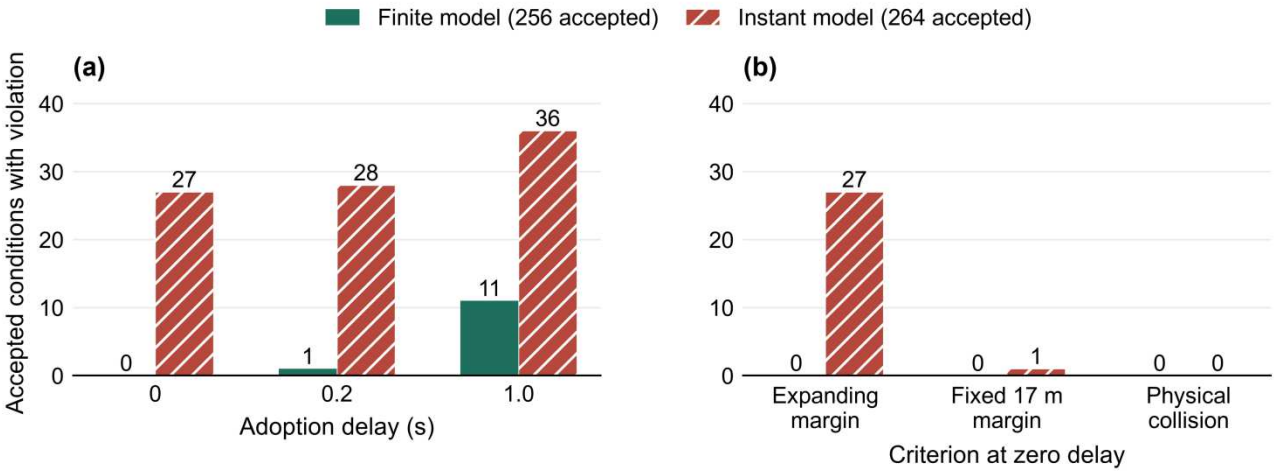


Figure 10. Preservation of original prediction labels during actual execution and under adoption delay. (a) Growing separation-margin violations among accepted conditions, with each model's acceptance denominator indicated. (b) Growing-margin, fixed 17 m separation, and physical-collision outcomes at zero delay. The three delay settings reuse physical inputs and t = 0 labels.

Figure 10 shows that the finite model improves motion assessment at zero delay while also revealing that candidate labels have limited validity for adoption. Retaining t = 0 labels with a 1 s adoption delay produced 11 growing-margin violations even for the finite model because the actual state had changed by adoption. A fresh rollout from the current state is therefore necessary for adoption revalidation. This diagnostic did not enable a new adoption-revalidation step: its counts support the design rationale for revalidation, while the task-level benefit still requires a separate controlled experiment. All 2268 joint histories were independently recomputed from saved controls, with a maximum motion-equation residual of approximately 9.15×10^{-13} m. Motion-envelope and constant-velocity traffic-input checks also passed, providing auditable execution evidence for this candidate-level finding.

5.6 Experiment V: Degradation, Numerical Sensitivity, and Computational Overhead

Constant-velocity prediction, observation errors, and the timeliness of action adoption jointly constrain dynamic responses. This section varies these conditions separately to assess whether the task performance in Section 5.3 persists and whether integration step size or computational overhead limits deployment. The 90 base scenarios were each tested with traffic acceleration, traffic turning, noise and occlusion, a 0.2 s decision delay, or a 1 s decision delay. RDPHPP and Reactive were paired within each condition, with 90 runs per method. Perturbations were applied individually, and methods shared the corresponding inputs. Statistical tests compare methods within the same condition.

Under acceleration, traffic speed is updated each step as $u^{+} = \mathrm{clip}_{[6,25]}[u + 0.8\sin(t/8 + \phi_i)\Delta t]$. Under turning, direction changes each step by $(4\pi/180)\sin(t/10 + \phi_i)\Delta t$. The phase ϕ_i is sampled from $[-\pi, \pi]$ during scenario generation and then frozen. At geometric boundaries, velocity is reversed and position is held for that step; the driving amplitudes therefore do not bound derivatives over complete trajectories that include boundary handling. The noise condition triples each standard deviation in Section 5.1 and enables building line-of-sight occlusion and a per-object detection-loss probability of 0.15. Sensing range, update rate, and observation validity after loss remain unchanged. The environment continues to advance during delays, and actions are revalidated using new observations when due.

As a supplementary check, the integration step was reduced from 0.1 to 0.05 s in 48 paired runs while the control period remained 0.2 s, and changes in task classification and surface-to-surface separation were

compared. Computational overhead uses all attempts in the frozen logs, including failed replanning attempts, and is reported through quantiles, maxima, and budget-overrun counts.

Table 8 and Figure 11(b) show slightly fewer RDPHPP completions under traffic acceleration and noise with occlusion, and slightly more under turning and both delay settings. None of the five method comparisons was significant after correction. With a 1 s delay, RDPHPP completed 76/90 tasks, 8 fewer than the 84/90 in the nominal mechanism subset. Traffic continues moving while actions await adoption, potentially consuming the remaining braking distance and turning space. Revalidation can reject stale proposals but cannot recover reaction time already lost. This explains why delays may reduce completions even when revalidation and finite-acceleration modeling are both present. The 8-completion difference between nominal and delayed conditions is descriptive; no additional cross-condition significance test was performed.

Table 8. Successful runs in the robustness experiments, with 90 runs per method and condition.

Condition	RDPHPP	Reactive
Traffic acceleration	81	82
Traffic turning	88	87
Noise and occlusion	85	86
0.2 s decision delay	86	83
1 s decision delay	76	72

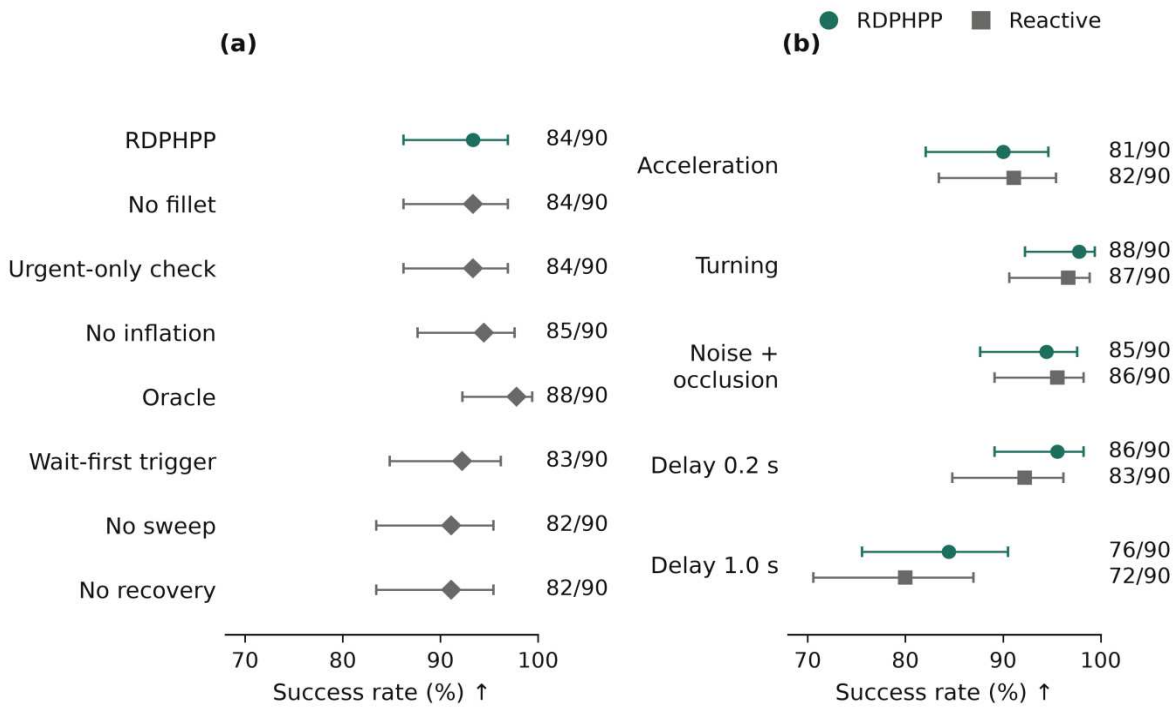


Figure 11. Mechanism comparisons and degradation conditions in the original urban protocol. (a) Eight configurations from Appendix E. (b) RDPHPP and Reactive under five degradation conditions. Each configuration has a denominator of 90; horizontal lines show Wilson 95% intervals. No success-rate comparison was significant after correction.

Traffic acceleration and turning cause constant-velocity extrapolations to depart from ground truth, while noise and occlusion change the initial state used for prediction. Local control continuously reads new observations and can adjust some actions promptly, consistent with both methods completing most tasks under degradation. None of the five differences was significant, so these data do not demonstrate that long-horizon prediction outperforms Reactive under every degradation condition. The seven mechanism comparisons in Figure 11(a) were also nonsignificant. They are retained to identify aspects requiring broader coverage, rather than to rank module importance by point estimates.

Across the 48 time-step comparison pairs, halving the step from 0.1 to 0.05 s without changing the control period caused no reversal of task success classifications. Differences in minimum surface-to-surface separation had a median of 0.292 m and a maximum of 33.83 m; 10 pairs differed by more than 2 m. Finer integration changes subsequent states and candidate selection, allowing trajectories to diverge while still reaching the same endpoint. These results support the stability of task classifications to this step-size change while quantifying the sensitivity of separation estimates. Continuous-time trajectories and numerical convergence require finer investigation. The original step-diagnostic summary.json describes control updates at every integration step, whereas audit.json and the hash-matched frozen driver specify a fixed 0.2 s control period. The interpretation here follows the latter records and the controller scheduling code. The original files are retained unchanged, with the consistency check supplied in Experiment 5.

Table C1 in Appendix C summarizes complete in-mission replanning times. Across 2448 RDPHPP attempts, the median, 95th percentile, 99th percentile, and maximum were approximately 60, 558, 1453, and 2065 ms; 14 attempts exceeded the 2 s budget. Across the five methods, 35 over-budget attempts affected 21 tasks. Budget checks therefore do not enforce strict deadlines.

The 123386.4 ms local-control wall-clock maximum was traced to a single-obstacle overtaking task with a configured 1 s decision delay; its original task ID and records are supplied in the Experiment 5 directory. The task had a local-control median of 0.837 ms and a 99th percentile of 1.943 ms, a total wall-clock runtime of 124.549 s, and a simulated duration of 87.7 s. It reached the goal without collision and involved no in-mission replanning, yielding commitment, or recovery. The archived records retain aggregate local timings but no per-call timestamps, CPU timings, compilation, garbage-collection, or scheduling traces. The simulation time of the slow call and the cause of the pause therefore cannot be determined.

Three supplementary warmed replays using the frozen source and identical inputs reproduced all 878 archived trajectory frames and the events exactly. Each replay made 439 local-control calls with 136 candidates per call, no expanded recovery, and no additional Numba compilation signatures after warm-up. The respective local maxima were 1.580, 1.183, and 1.056 ms. Failure to reproduce the extreme value neither explains nor replaces the original record; these diagnostic measurements are stored separately from the formal statistics. Because the fixed-step simulated world does not advance while the solver runs, the extreme wall-clock time does not change this offline run's recorded arrival or collision outcome. Traffic in physical flight would continue moving, so the record precludes a worst-case real-time guarantee. The 0.2 s control period specifies simulation scheduling and does not establish a 5 Hz closed loop on hardware.

6 Discussion

RDPHPP's contributions must be assessed at their corresponding levels. Section 5.2 tests rule propagation through configuration replacement and reference construction through shared geometric acceptance. Section 5.5 directly checks execution conditions for finite-deceleration candidates. Sections 5.3, 5.4, and 5.6 evaluate completion, efficiency, and degradation when these components are combined. The following discussion interprets the method's value, relates it to existing designs, and examines dependence on modeling and information conditions.

6.1 Methodological Strengths of RDPHPP

1) An independent rule interface propagates configuration changes into a shared feasible domain.

The value of the interface is supported directly by configuration replacement in Section 5.2, rather than inferred indirectly from search acceptance rates. With start–goal pairs, search code, and costs held constant, replacing only the exclusion configuration yielded 14 rerouted pairs, 4 newly infeasible pairs, and 42 unchanged outcomes. All 116 successful paths satisfied the corresponding exclusion checks. These results show that input configuration changes can alter admissible routes while acceptance uses the same updated geometry, addressing the maintenance problem created when rule checks are distributed throughout algorithm internals. The evidence verifies propagation behavior; maintenance time saved when changing cities has not been measured.

A separate set of metrics supports search and smoothing. Both A* variants in Table 4 found 60 connections, but RDPHPP accepted 57 references compared with 30 for distance-based A*. The fixed-polyline circular-fillet

diagnostic increased acceptance from 40 to 57. Direction states and obstacle-proximity costs prepare space for connections, while conditional circular-fillet recovery adds geometric candidates after standard connection fails. This advantage occurs at the reference-construction level, reducing the transfer of curvature- or clearance-invalid paths to dynamic response. It complements rule-configuration propagation, but the two are supported by separate experiments.

2) Time alignment improves execution consistency in finite-deceleration candidate evaluation.

Fixed ETAs or instantaneous target-speed changes omit continued forward motion during deceleration. RDPHPP uses the same integrator as execution and starts its rollout from the same current speed, allowing yielding time and along-route distance to be checked together. Even if an obstacle is known to leave the conflict region, a candidate is rejected when the minimum speed cannot delay passage sufficiently. Thus, permission to wait becomes a concrete condition: the current state must permit the required deceleration and yielding.

Section 5.5 directly supports this evaluation benefit. Finite prediction accepted 256 conditions with 0 growing-margin violations; the instantaneous approximation accepted 264 with 27 violations. All 23 conditions accepted only by the finite model maintained the original protection. The finite-motion model both excluded the observed false acceptances and retained valid conditions missed by the instantaneous approximation. Both task mappings produced identical completion counts and zero physical collisions, showing that the consistency gain concerns candidate commitments, while final task outcomes also depend on local vetoes and route changes.

Candidate states must also remain current. With a 1 s adoption delay, 11 conditions originally accepted by the finite model violated the protection requirement, showing that even a correct motion model depends on the state at generation. Revalidating the current position, the target validity period, and remaining trajectory before adoption incorporates this timeliness requirement into the interface. The experiment motivates revalidation; its independent closed-loop benefit requires further isolation.

3) Shared acceptance and revocable responses make candidate adoption and failure handling explicit.

Continuation, yielding, and detour candidates must satisfy motion, clearance, and dynamic timing conditions before their costs in Eq. (12) are compared. Waiting is bound to a target, deadline, and budget, with speed restrictions released when the commitment becomes invalid. Dynamic swept regions guide spatial search only temporarily; they can be withdrawn for a retry when search fails, and every new candidate is dynamically rechecked. The existence of a spatial path and an obstacle's departure time therefore support candidate generation, while adoption still depends on the complete trajectory from the current state. This addresses the use of inconsistent feasibility conditions across responses.

Section 5.3 provides task-level evidence for the combined response workflow: RDPHPP's success rate exceeded GVO-adapt's by 15.00 percentage points, with a significant corrected difference. Finite braking, conditional recovery, and failure labels when normal control finds no feasible action are also retained in the execution logs. Section 5.4 quantifies the costs: mean distance increased by 35.68 m relative to Reactive, and completion time and model energy increased by 21.14 s and 160.14 kJ relative to GVO-adapt. Success-rate differences from Reactive, No-wait, and Legacy were nonsignificant, as were the mechanism comparisons in Appendix E. Overall performance and individual mechanism effects must therefore be interpreted separately. Shared acceptance provides inspectable adoption conditions; closed-loop safety is still evaluated from ground-truth trajectories.

No-wait and RDPHPP had identical outcomes across all 540 scenarios. Response coverage helps interpret this result: among 9437 rolling arbitration checks, only 52 found yielding and detouring simultaneously feasible, and 4 selected yielding. The current matrix rarely covers conditions requiring a choice between the two high-level responses, and detours or local actions may also substitute for waiting. Both explanations are consistent with the records but have not been isolated. Efficiency differences were likewise nonsignificant. Future work should define naturally arising tradeoff scenarios on the development set and then freeze a confirmation set, retaining cases in which waiting has no opportunity or has a negative benefit.

6.2 Relationship to Existing Research

Aiello et al. coordinate energy-oriented global search and local obstacle avoidance for fixed-wing missions [2]. RDPHPP follows the cooperation between a global reference and local responses, separates urban exclusion configurations from search costs, and introduces explicit acceptance interfaces for references and dynamic proposals. Configuration replacement in Section 5.2 shows that rule changes directly alter admissible routes, while 57/60 static acceptances support the practical value of turn- and clearance-aware reference construction. The tradeoff is that stronger acceptance conditions may reject spatially connected paths. The A*–DWA fusion of He et al. and the neighborhood search and circular fillet smoothing of Xu et al. improve tracking and geometry, respectively [9, 25], showing that module combinations and circular connections already have established foundations. This paper additionally compares finite-deceleration responses and records adoption failures. Its contribution lies in connecting constraints and validating that connection at separate levels. These published methods were not numerically reproduced on this platform; performance conclusions rely on the implemented comparisons in Section 5.

GVO evaluates collision candidates in a control space subject to motion constraints, making it suitable for incorporating dynamics into action evaluation [23]. GVO-adapt and RDPHPP in this project share action sampling and trajectory revalidation. GVO-adapt selects targets by terminal-velocity proximity to the desired direction, whereas RDPHPP adds long-horizon route adjustments and finite-deceleration commitments. The 81 additional completions and significant paired result in Section 5.3 support the overall value of this response organization for the present inputs. Section 5.4 also shows increases of 21.14 s in time and 160.14 kJ in model energy, indicating that the benefit comes with costs in speed profiles and computational responses. This comparison concerns the complete coordination scheme: the 81 additional completions cannot all be attributed to waiting, and the adapted implementation does not represent every GVO configuration.

SIPP-IP accounts for acceleration and deceleration within safe time intervals and provides completeness and optimality analyses for the corresponding search problem [3]. RDPHPP reuses two-dimensional spatial search and checks candidates over a finite horizon, facilitating shared rule masks and global components. This organization adds timing checks with limited structural changes but may miss feasible solutions requiring other waiting locations or arrival times. Section 5.5 validates execution consistency for generated yielding requests; coverage of the spatiotemporal solution space has not been measured. DYNUS further combines spatiotemporal corridors with safe, exploratory, and emergency trajectories [13], demonstrating a more comprehensive three-dimensional organization of prediction and fallback. This paper adopts a simpler two-dimensional candidate workflow, whose coverage and computational costs require further comparison on shared inputs.

MADER and Robust MADER coordinate multiple vehicles through committed trajectories and communication; Robust MADER explicitly handles communication delays [12, 22]. RDPHPP addresses traffic that neither shares plans nor actively yields. The ego UAV relies only on limited-range observations, avoiding trajectory negotiation while remaining exposed to target-prediction errors and insufficient response time. The decline from 84 nominal-subset completions to 76 under a 1 s decision delay in Section 5.6 is consistent with this information limitation; it concerns action-adoption delay, not communication validation. SUPER retains a safe trajectory in known free space, while gatekeeper analyzes persistent safety using committed trajectories and a fallback mechanism [1, 20]. Relative to these safety constructions, the failure exits here improve state interpretability, but the 25 main dynamic collisions and the minimum surface-to-surface separation of 0.538 m in successful runs show that finite-horizon acceptance has not established persistent safety conditions.

6.3 Limitations and Future Work

1) The constant-altitude two-dimensional model does not capture physical composite-wing transition flight.

The constant-altitude two-dimensional model abstracts transition flight as low-speed lift assistance, omitting tilting, stall, aerodynamic coupling, and wind disturbances. Consequently, the reachable low-speed action set may be broader than that of a physical platform, and sustained lift after deceleration and subsequent recovery to cruise have not been verified. Parameter calibration may change the acceptance sets for both yielding and detouring. In the Section 3.2 example, decelerating from 30 to 8 m/s at 6 m/s² adds 40.33 m relative to the

instantaneous approximation. Changing only the limit to 3 m/s² gives 80.67 m under the same straight-line constant-deceleration estimate, illustrating how strongly maneuver parameters can affect timing decisions. This calculation illustrates model dependence; it is not an additional task-level sensitivity experiment. The transition speed in Eq. (4) equals the maximum execution speed, so existing tasks mainly cover the low-speed branch. Power has also not been calibrated on a physical vehicle. Static acceptance and candidate consistency therefore constitute evidence internal to the model. Physical composite-wing transition and energy magnitudes require further assessment through nonlinear flight modeling and platform validation of the kind studied by Lu et al. [14]. Both city maps were used during development, and generalization to unseen maps remains untested.

2) Constant-velocity prediction and limited observations constrain available reaction lead time.

A 45 s rollout window does not provide 45 s of warning: the 150 m sensing range, occlusion, and relative closing speeds determine the actual available lead time, while traffic acceleration and turning progressively invalidate constant-velocity predictions. Section 5.6 reports 81/90, 88/90, and 85/90 completions under acceleration, turning, and noise with occlusion, respectively, with nonsignificant method differences in all five groups. A 1 s decision delay yielded 76/90 completions, 8 fewer than the nominal 84/90, demonstrating an observable effect of state timeliness. The experiments use prescribed perturbation amplitudes and detection-loss rates and do not cover arbitrary abrupt maneuvers. The perfect-observation straight-line diagnostic also does not quantify prediction errors from urban occlusion. Qiu et al. incorporate dynamics and disturbance estimation into local optimization, while DYNUS explicitly addresses uncertainty [13, 19]. These studies can inform improved prediction, which should be evaluated through acceptance sets, protection violations, and task endpoints together so that predictor changes are assessed for their execution consequences.

3) Acceptance conditions do not provide a closed-loop safety guarantee.

The 25 collisions in Section 5.3 and the minimum surface-to-surface separation of 0.538 m in Section 5.4 show a remaining gap between an acceptable reference, an accepted candidate, and closed-loop collision avoidance. Nominal rollouts differ from actual DWA trajectories, and conditional recovery can change the original request. Both the 17 m separation and growing margin are design conditions. Although halving the integration step did not change task classifications, the maximum separation difference remained 33.83 m, indicating sensitivity of states and action selection to discretization. Future work could draw on gatekeeper's fallback and persistent-safety verification or introduce safe corridors and safety filters [1], incorporating prediction error, finite braking, adoption delay, and infeasible states into common conditions.

4) Computational tails and decision delays have not been validated in a closed loop on target hardware.

In Section 5.6, RDPHPP completed 76/90 tasks under a 1 s decision delay, compared with 84/90 in the nominal subset. Fourteen of 2448 replanning attempts exceeded the 2 s budget. The identity and timing boundaries of the original 123.4 s local-control extreme have been verified. Three frozen-source replays reproduced the trajectory but not the pause; the original instrumentation is insufficient to identify its cause. Separating solver wall time from environment advancement preserves the recorded arrival under the offline protocol but does not establish the outcome in physical flight or worst-case real-time execution. Further work should record per-call CPU time, wall time, and system events on target hardware such as Jetson/Orin, let traffic advance with actual computation time, and test deadlines, timeout exits, and fallback actions. Multi-vehicle rule coordination would additionally require communication and commitment mechanisms.

7 Conclusion

This paper addresses the constraint-connection problem posed in the introduction through independent rule propagation, curvature- and clearance-aware reference construction, predictive arbitration using a shared finite-motion model, and local adoption revalidation. Under configurable exclusions, constant-velocity traffic prediction, and bounded acceleration, admissible space, acceptable references, and dynamic yielding and detour responses can be connected into an inspectable planning workflow. The principal contributions and findings of RDPHPP are as follows.

(1) At the system level, an independent rule interface propagates exclusion-configuration changes into search and acceptance. Among 60 configuration-replacement pairs, 14 required rerouting and 4 became infeasible,

while all 116 successful paths complied with the corresponding rules. This verifies constraint propagation without changing search logic.

(2) At the reference-construction level, direction-augmented multi-cost search and conditional circular-fillet recovery improve geometric acceptance. RDPHPP accepted 57/60 references, compared with 30/60 for distance-based A*. The fixed-input switch increased acceptance from 40/60 to 57/60, directly supporting the role of alternative connections in supplementing standard smoothing.

(3) At the prediction level, a shared finite-acceleration integrator incorporates speed reduction into candidate evaluation. In controlled execution, growing-margin violations changed from 27/264 for the instantaneous approximation to 0/256 for the finite model, which also retained 23 valid conditions accepted only by that model. A 1 s adoption delay still caused 11 violations, demonstrating the need to revalidate the adoption state.

(4) At the execution level, DWA with a speed-dependent envelope, adoption revalidation, and finite-braking failure exits make candidate adoption, local adjustment, and infeasible states individually inspectable. These mechanisms provide a coherent handling workflow. The 25 main dynamic collisions and the minimum surface-to-surface separation of 0.538 m along successful trajectories also identify the remaining gap between finite-horizon acceptance and persistent closed-loop safety.

(5) At the validation level, paired two-city tasks, independent traffic, and controlled diagnostics distinguish overall performance from mechanism evidence. RDPHPP completed 515/540 main tasks, improving success by 15.00 percentage points over GVO-adapt with a significant difference, accompanied by time and model-energy costs. Completion-rate differences from Reactive and No-wait were nonsignificant. The paper thus provides an implementation and evidence connecting rules, references, and finite-prediction execution, while the independent task benefit of waiting requires broader coverage.

Further validation will focus on physical composite-wing transition and power calibration, candidate revalidation under observation errors, actual solver delays on target hardware, and persistent-safety fallback conditions. Three-dimensional tasks and unseen maps will be added while preserving the auditability of these interfaces.

Declarations

Funding

This work was supported by GCC under Grant No. 2025KYQD64

Availability of Data and Materials

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request. Partial data has been provided to reviewers through the submission system.

Author Contributions

Hanyu Liu: Methodology (algorithm design), Software, Writing – Original Draft (main).

Meng Wang: Supervision, Writing – Review & Editing, Funding Acquisition

Ying Li: Methodology (experimental design), Writing – Original Draft (partial).

Ethics Approval

Not applicable. This study reports computational UAV path-planning simulations and does not involve human participants, animals, or personal data collected through real-world flight experiments.

AI-Assisted Work

OpenAI tools were used in the research to assist in literature review, partial code writing, and review. The author is responsible for the research content, data, citations, and final manuscript, and verifies the aforementioned auxiliary content before submission.

References

[1] Agrawal, D. R., Chen, R., & Panagou, D. (2024). gatekeeper: Online Safety Verification and Control for Nonlinear Systems in Dynamic Environments. IEEE Transactions on Robotics, 40, 4358–4375. https://doi.org/10.1109/TRO.2024.3454415.

[2] Aiello, G., Valavanis, K. P., & Rizzo, A. (2022). Fixed-Wing UAV Energy Efficient 3D Path Planning in Cluttered Environments. Journal of Intelligent & Robotic Systems, 105(3), Article 60. https://doi.org/10.1007/s10846-022-01608-1.

[3] Ali, Z. A., & Yakovlev, K. (2023). Safe Interval Path Planning with Kinodynamic Constraints. Proceedings of the AAAI conference on artificial intelligence, 37, 12330–12337. https://doi.org/10.1609/aaai.v37i10.26453.

[4] Candan, F., Beke, A., Mahfouf, M., & Mihaylova, L. (2024). A Real-time Fuzzy Interacting Multiple-Model Velocity Obstacle Avoidance Approach for Unmanned Aerial Vehicles. Journal of Intelligent & Robotic Systems, 110(2), Article 61. https://doi.org/10.1007/s10846-024-02075-6.

[5] Fiorini, P., & Shiller, Z. (1998). Motion Planning in Dynamic Environments Using Velocity Obstacles. The International Journal of Robotics Research, 17(7), 760–772. https://doi.org/10.1177/027836499801700706.

[6] Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23–33. https://doi.org/10.1109/100.580977.

[7] Haklay, M., & Weber, P. (2008). OpenStreetMap: User-Generated Street Maps. IEEE Pervasive Computing, 7(4), 12–18. https://doi.org/10.1109/MPRV.2008.80.

[8] Hart, P., Nilsson, N., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100–107. https://doi.org/10.1109/TSSC.1968.300136.

[9] He, Y., Hou, T., & Wang, M. (2024). A new method for unmanned aerial vehicle path planning in complex environments. Scientific Reports, 14, Article 9257. https://doi.org/10.1038/s41598-024-60051-4.

[10] Holm, S. (1979). A Simple Sequentially Rejective Multiple Test Procedure. Scandinavian Journal of Statistics, 6(2), 65–70. https://www.jstor.org/stable/4615733.

[11] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 30(7), 846–894. https://doi.org/10.1177/0278364911406761.

[12] Kondo, K., Figueroa, R., Rached, J., Tordesillas, J., Lusk, P. C., & How, J. P. (2024). Robust MADER: Decentralized Multiagent Trajectory Planner Robust to Communication Delay in Dynamic Environments. IEEE Robotics and Automation Letters, 9(2), 1476–1483. https://doi.org/10.1109/LRA.2023.3342561.

[13] Kondo, K., Peterson, M., Rober, N., Rached Viso, J., Jia, L., Chen, J., Merton, H., & How, J. P. (2025). DYNUS: Uncertainty-aware Trajectory Planner in Dynamic Unknown Environments. arXiv preprint arXiv:2504.16734v3. https://doi.org/10.48550/arXiv.2504.16734.

[14] Lu, G., Ren, Y., Zhu, F., Li, H., Xue, R., Cai, Y., Lyu, X., & Zhang, F. (2024). Autonomous Tail-Sitter Flights in Unknown Environments. arXiv preprint arXiv:2411.15003v2. https://doi.org/10.48550/arXiv.2411.15003.

[15] McNemar, Q. (1947). Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. Psychometrika, 12(2), 153–157. https://doi.org/10.1007/BF02295996.

[16] Mugnai, M., Teppati Losè, M., Satler, M., & Avizzano, C. A. (2024). Towards Autonomous Firefighting UAVs: Online Planners for Obstacle Avoidance and Payload Delivery. Journal of Intelligent & Robotic Systems, 110(1), Article 10. https://doi.org/10.1007/s10846-023-02042-7.

[17] Phillips, M., & Likhachev, M. (2011). SIPP: Safe interval path planning for dynamic environments. 2011 IEEE international conference on robotics and automation, 5628–5635. https://doi.org/10.1109/ICRA.2011.5980306.

[18] Primatesta, S., Guglieri, G., & Rizzo, A. (2019). A Risk-Aware Path Planning Strategy for UAVs in Urban Environments. Journal of Intelligent & Robotic Systems, 95(2), 629–643. https://doi.org/10.1007/s10846-018-0924-3.

[19] Qiu, J., Liu, Q., Qin, J., Cheng, D., Tian, Y., & Ma, Q. (2024). PE-Planner: A Performance-Enhanced Quadrotor Motion Planner for Autonomous Flight in Complex and Dynamic Environments. IEEE Robotics and Automation Letters, 9(6), 5879–5886. https://doi.org/10.1109/LRA.2024.3396110.

[20] Ren, Y., Zhu, F., Lu, G., Cai, Y., Yin, L., Kong, F., Lin, J., Chen, N., & Zhang, F. (2025). Safety-assured high-speed navigation for MAVs. Science Robotics, 10(98), eado6187. https://doi.org/10.1126/scirobotics.ado6187.

[21] Şenbaşlar, B., Hönig, W., & Ayanian, N. (2023). RLSS: real-time, decentralized, cooperative, networkless multi-robot trajectory planning using linear spatial separations. Autonomous Robots, 47(7), 921–946. https://doi.org/10.1007/s10514-023-10104-w.

[22] Tordesillas, J., & How, J. P. (2022). MADER: Trajectory Planner in Multiagent and Dynamic Environments. IEEE Transactions on Robotics, 38(1), 463–476. https://doi.org/10.1109/TRO.2021.3080235.

[23] Wilkie, D., van den Berg, J., & Manocha, D. (2009). Generalized velocity obstacles. 2009 IEEE/RSJ international conference on intelligent robots and systems, 5573–5578. https://doi.org/10.1109/IROS.2009.5354175.

[24] Wilson, E. B. (1927). Probable Inference, the Law of Succession, and Statistical Inference. Journal of the American Statistical Association, 22(158), 209–212. https://doi.org/10.1080/01621459.1927.10502953.

[25] Xu, L., Xi, M., Gao, R., Ye, Z., & He, Z. (2025). Dynamic path planning of UAV with least inflection point based on adaptive neighborhood A* algorithm and multi-strategy fusion. Scientific Reports, 15, Article 8563. https://doi.org/10.1038/s41598-025-92406-w.

[26] Yang, K., & Sukkarieh, S. (2010). An Analytical Continuous-Curvature Path-Smoothing Algorithm. IEEE Transactions on Robotics, 26(3), 561–568. https://doi.org/10.1109/TRO.2010.2042990.

[27] Zong, Z., Li, D., Dong, X., Cui, Y., Yang, B., Xiang, J., & Tu, Z. (2025). Risk-Aware Enabled Path Planning for Drones Flight in Unknown Environment. Journal of Intelligent & Robotic Systems, 111(2), Article 47. https://doi.org/10.1007/s10846-025-02243-2.

Appendix A Data verification and statistical definitions

The experimental materials are organized into five directories corresponding to Sections 5.2–5.6. Each directory contains original records, processed tables, the figures used in that section, and an index linking the records to the manuscript. Experiment 1 contains rule configurations A/B, static runs, and the fixed-input fillet diagnostic; Experiment 2 contains the main dynamic and independent-traffic runs; Experiment 3 contains the reused efficiency and surface-to-surface separation trajectories; Experiment 4 contains controlled requests, prediction labels, and joint execution histories; and Experiment 5 contains degraded conditions, integration-step checks, computational records, and the supplementary timing diagnostics conducted in this revision. Shared maps, frozen source code, and configuration provenance are supplied in the Shared directory, and the archived implementation is supplied separately in Appendix_F. For ease of review, reused records are provided in each relevant experiment directory, remain identified by their original task IDs, and are not counted as additional independent samples. The recorded environment was Windows build 26200, AMD64 Family 25 Model 117 Stepping 2, with 16 logical CPUs; the full commercial processor model was not retained.

Data-integrity checks covered missing tasks, duplicates, invalid records, and execution errors. The original audit examined 1295 trajectories, including all 305 collision runs and all dynamic RDPHPP runs; inter-step dynamic separations agreed with the stored records. Dynamic collision, static collision, timeout, infeasible smoothing, rule violation, planning timeout, and initial infeasibility are retained as distinct termination states.

Search return, smoothing acceptance, candidate adoption, and mission completion are recorded separately. Resource use before early failure is not interpreted as a completion-efficiency benefit.

Cases were selected by the smallest scenario ID within the specified outcome set. Improvement, degradation, and static-recovery cases are all retained. Figure data include the denominators, original trajectory paths, and selection rules. Directory descriptions and SHA-256 manifests support verification of the delivered files; these checks assess consistency among the records, equations, and statistics.

Appendix B Model-based energy consumption

The frozen power model is given by Eq. (B1):

$$P(v,\kappa,a_t)=\begin{cases}\max(C_0+C_1v^3+C_2\kappa^2v^3+C_3|a_t|v,C_0), & v\geq v_k,\\ \max(P_0+P_1v+P_2\kappa^2\max(v,0.5)+C_3|a_t|v,0.3P_0), & v<v_k.\end{cases} \quad \text{(B1)}$$

In SI units, $C_0=200$ W, $C_1=0.025725$, $C_2=37.3168665$, $C_3=33.3333333$, $P_0=3425.1562507$ W, $P_1=-68.1444224$, and $P_2=41477.6971523$; the units of each coefficient follow from the requirement that its term has units of watts. The breakpoint is $v_k=\sqrt{24.7\times 45}=33.3391662$ m/s and is not exactly equal to $A=v_{\max}^2/45$ in the motion envelope. Execution speeds do not exceed 33.3 m/s, so all dynamic evaluations use the low-speed branch. Static-path energy evaluated at the default model breakpoint speed is not directly pooled with energy from executed dynamic trajectories.

At each step, curvature and tangential acceleration are estimated from the mean speed, heading change, and speed difference, and power is integrated over time. The speed denominator in the low-speed curvature calculation is bounded below by 0.5 m/s. Deceleration contributes through the absolute-acceleration term; energy recovery is not modeled. The coefficients establish a common power scale for comparison but have not been calibrated on a physical aircraft. Energy is therefore treated as a model-based metric and interpreted alongside time, distance, and the speed profile.

Appendix C Local objectives and implementation parameters

The local dynamic window is 0.5 s. The speed set contains seven uniformly spaced targets in $[\max(0,v-6\times 0.5),\min(v_{\max},v+6\times 0.5)]$ and a zero target; the angular-rate set contains 17 targets in $[\max(-1,\omega-0.5),\min(1,\omega+0.5)]$ and a zero target. Duplicate values are removed. A yielding commitment or an initially speed-limited reference may further reduce the target-speed upper bound. The expanded recovery set adds more distant target speeds without changing the acceleration bound between actual integration steps.

Let c denote the minimum static/dynamic center clearance over a local rollout, α the terminal heading error toward the local goal, p the reduction in goal distance divided by $v_{\max}H$, and v_H the terminal speed. With $b(c)=\mathrm{clip}[(c-r_a-1)/(60-r_a-1),0,1]$, the speed preference is $v_d=4+(v_{\max}-4)b(c)$. Among accepted targets, DWA minimizes Eq. (C1):

$$J_{\mathrm{DWA}}=0.35(\alpha/\pi-0.5p)+0.30\{1-\mathrm{clip}(c/60,0,1)\}+0.20\frac{|v_H-v_d|}{\max(v_{\max}-12,1)}+0.15\frac{|\omega_c|}{\omega_{\max}}. \quad \text{(C1)}$$

The last term penalizes the magnitude of the target angular rate, rather than a difference between consecutive controls. GVO-adapt uses Eq. (C2) on the same feasible set:

$$J_{\mathrm{GVO}}=\frac{\|v_H(\cos\psi_H,\sin\psi_H)-v_{\max}(\cos\psi_g,\sin\psi_g)\|^2}{v_{\max}^2}-0.25p+0.05\frac{|\omega_c|}{\omega_{\max}}, \quad \text{(C2)}$$

Here, ψ_g is the direction from the current state toward the local goal. The two controllers share target sampling and trajectory revalidation; they differ in their evaluation of accepted actions and their interaction with global planning.

Table C1 Wall-clock times of complete in-mission replanning attempts. Quantiles are calculated over attempts. The last column is the largest single local-control time across all runs and is not part of the replanning distribution.

Method	Attempts	p50 / ms	p95 / ms	p99 / ms	Maximum / ms	Over-budget attempts	Local maximum / ms
RDPHPP	2448	60.4	558.5	1452.6	2064.7	14	123386.4
Reactive	557	25.0	478.7	972.0	2006.0	1	46.8
No-wait	1377	63.8	718.3	1884.5	2084.9	11	45.8
Legacy	1068	50.7	643.2	1513.6	2046.0	9	52.5
GVO-adapt	2546	9.7	142.3	456.9	1408.6	0	43.5

Appendix D Paired effect sizes and resource use before failure

Table D1 reports the complete paired statistics for jointly successful runs in the main dynamic study.Table D2 summarizes resource use before termination among failed main dynamic runs.

Table D1 Complete paired statistics for jointly successful runs in the main dynamic study. Differences are RDPHPP minus the comparator. Intervals are 95% confidence intervals for the mean difference from 9999 paired bootstrap resamples. r_{rb} is the rank-biserial correlation, and p_H is the Holm-adjusted value for the prespecified family of continuous-metric tests. A positive effect indicates a larger value of the corresponding metric for RDPHPP.

Comparator	Metric	Paired n	Mean difference	95% interval	r_{rb}	p_H
Reactive	Time / s	500	-0.54	[-1.69, 0.56]	-0.013	1.000
Reactive	Distance / m	500	+35.68	[23.85, 49.39]	+0.489	6.8e-11
Reactive	Model energy / kJ	500	+1.93	[-2.58, 6.42]	+0.153	0.257
No-wait	Time / s	515	-0.24	[-0.77, 0.17]	+0.198	1.000
No-wait	Distance / m	515	-7.41	[-16.43, -0.92]	-0.350	0.650
No-wait	Model energy / kJ	515	-1.52	[-3.76, 0.21]	-0.099	1.000
Legacy	Time / s	499	-0.70	[-1.74, 0.34]	-0.123	0.635
Legacy	Distance / m	499	+13.24	[-0.92, 28.42]	+0.069	1.000
Legacy	Model energy / kJ	499	+0.13	[-4.53, 4.53]	+0.040	1.000
GVO-adapt	Time / s	419	+21.14	[18.92, 23.34]	+0.862	8.8e-52
GVO-adapt	Distance / m	419	-74.01	[-107.22, -43.69]	-0.579	9.5e-24
GVO-adapt	Model energy / kJ	419	+160.14	[149.43, 170.41]	+0.954	4e-63

Table D2 Mean resource use before termination among failed main dynamic runs. Each method contributes its own failures; these samples are not paired between methods. Runs reaching the time limit retain resource use up to that limit. The table describes failure processes and is not used to infer completion efficiency.

Method	Failed runs	Time before termination / s	Distance before termination / m	Model energy before termination / kJ
RDPHPP	25	64.54	771.61	254.74
Reactive	34	68.27	813.05	269.89
No-wait	25	64.54	771.61	254.74
Legacy	32	73.44	910.81	292.92
GVO-adapt	106	149.04	3727.58	517.08

Appendix E Mechanism switches and response coverage

This appendix reports seven mechanism comparisons under the original urban protocol. The first five prespecified scenarios from each map, density, and conflict-type combination in the main matrix form a shared set of 90 scenarios. Each variant changes only the implementation switch specified in Table E1.

Table E1 Mechanism comparisons under the original urban protocol, with 90 shared inputs per configuration. None of the seven exact paired success comparisons is significant after Holm correction; all adjusted p values equal 1.

Configuration	Successful runs	Success rate
RDPHPP	84	93.33%
No fillet recovery	84	93.33%
Urgent-only dynamic route check	84	93.33%
No predictive margin inflation	85	94.44%
Oracle observations	88	97.78%
Wait-first trigger	83	92.22%
No swept-region injection	82	91.11%
No conditional recovery	82	91.11%

Table E1 and Fig. 11a show the same counts. The no-fillet variant and RDPHPP each have one exclusive success, demonstrating that equal completion counts can mask different execution outcomes. None of the seven adjusted comparisons is significant. The mission-level contribution of individual modules requires further evaluation with more extensive mechanism coverage.

Across 1050 dynamic runs, 9437 rolling arbitration events were recorded: yielding alone was feasible in 80, detouring alone in 560, both in 52, and neither in 8745. When both were feasible, yielding was selected four times and detouring 48 times. “Neither feasible” refers only to these two high-level candidates; a local action may still be available. Events include reused scenarios and are not treated as independent samples.

Appendix F Supplementary comparison with an archived implementation

The archived v17 implementation was rerun on the same frozen scenarios with shared parameters. It differs from Legacy, which replaces only the decision logic within the current engine. Because the version comparison involves changes across the implementation, it is reported here rather than treated as a single-mechanism ablation. Static, dynamic, and efficiency results retain their original records. The original Holm family of five primary success comparisons is unchanged.

Table F1 summarizes static acceptance and dynamic task outcomes for the archived implementation and RDPHPP on shared inputs.

Table F1 Archived implementation and RDPHPP on shared inputs: 60 static cases and 540 dynamic scenarios. Success-rate p values retain the original joint five-comparison correction.

Implementation	Static acceptance	Dynamic success	Dynamic collision	Adjusted p versus RDPHPP
RDPHPP	57/60	515/540 (95.37%)	25/540 (4.63%)	—
Archived v17	40/60	467/540 (86.48%)	65/540 (12.04%)	2.45×10^{-12}

The paired outcomes comprise 50 RDPHPP-only successes and two archived-only successes, so improvement is not monotonic across individual tasks. Figure 8 retains cases in both directions, selected by the smallest scenario ID in each set; Fig. 6b retains the static version comparison. Section 5.2 separately verifies that the fixed-input fillet switch accounts for 17 additional static acceptances. The overall dynamic change is not attributed to that single module.

Table F2 reports the paired time, distance, and model-energy differences for dynamic runs in which both implementations succeed.

Table F2 Mean paired differences over the 465 dynamic runs in which both implementations succeed; differences are RDPHPP minus archived v17.

Pairs	Time difference / s	Distance difference / m	Model energy difference / kJ	Statistical interpretation
465	−0.04	−0.01	+0.14	All three intervals for the mean difference include zero; equivalence is not established.