

Article

Executable Reference Trajectory Construction and Conflict-Aware Residual Reinforcement Learning for Urban Multi-UAV Navigation

Xiangzhi Zhou ¹ , Siqin Li ¹, Qianjin Xia ^{1,*} and Shanmei Li ^{2,*}

¹ COMAC Aviation College, Ordos Institute of Technology, Ordos 017000, China; xz-zhou@outlook.com (X.Z.)

² College of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China

* Correspondence: xqj@oit.edu.cn (Q.X.); sm-li@cauc.edu.cn (S.L.)

Abstract

Urban multi-UAV navigation in dense building environments requires not only collision-free geometric paths but also executable flight processes under motion constraints and inter-UAV safety requirements. A static path that is feasible in a geometric map may still fail during closed-loop execution because of velocity limits, acceleration constraints, local path-association errors, and coupled multi-UAV interactions. Meanwhile, end-to-end reinforcement learning often suffers from unstable training, weak geometric interpretability, poor early-stage safety, and high sample complexity. To address these issues, this paper proposes a hierarchical planning-and-learning framework that connects static reference path generation, executable reference tracking, successful demonstration distillation, and conflict-aware residual reinforcement learning. First, three-dimensional reference paths are generated offline in an OpenStreetMap-based urban scene represented by cuboid buildings. Second, a damped reference-tracking mechanism transforms these static paths into closed-loop executable reference processes through local path association, monotonic progress updating, path recapture, look-ahead guidance, and bounded action construction. Third, successful pure-reference executions are distilled for behavior-cloning initialization. Finally, a bounded residual TD3 module is introduced as a local conflict-correction mechanism around the verified executable reference baseline. Experiments in an urban scene containing 754 buildings show that simplified tracking strategies fail to execute the static paths reliably, whereas the proposed full-damped reference-tracking controller achieves a 91.67% all-success rate and eliminates building collision episodes in the tracking-ablation test. Speed-sensitivity experiments at 10, 15, and 20 m/s show the same 91.67% all-success rate, indicating that the conclusion is not dependent on a single speed setting. In constructed conflict-stress tests, the conflict-aware residual TD3 module increases the all-success rate from 33.33% to 80.09%, reduces inter-UAV collision episodes from 66.67% to 11.57%, and improves the hard-safety satisfaction rate from 33.33% to 87.04%. These results show that the main contribution of the proposed framework lies in converting static geometric paths into executable reference trajectories and further enabling bounded residual correction under inter-UAV conflict conditions.

Keywords: multi-UAV navigation; urban path planning; executable reference trajectory; reference tracking; residual reinforcement learning; TD3



Academic Editor: Hailong Huang

Received: 6 May 2026

Revised: 8 July 2026

Accepted: 9 July 2026

Published: 13 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Urban low-altitude airspace is becoming an important operating domain for logistics delivery, infrastructure inspection, emergency response, traffic monitoring, and other unmanned aerial vehicle (UAV) applications [1–3]. Compared with open-area UAV operations, urban navigation is more difficult because tall buildings fragment the available flight space, narrow corridors limit maneuverability, and multiple UAVs may operate simultaneously in the same low-altitude region. In such environments, a navigation method must satisfy not only goal-reaching and obstacle-avoidance requirements but also inter-UAV separation, execution stability, and computational feasibility.

A key difficulty in urban UAV navigation is that geometric path feasibility does not automatically imply execution feasibility. Many path-planning methods output a static geometric path, usually represented as a discrete waypoint sequence or a polyline in three-dimensional space. Such a path may be collision-free in a static map, but this does not guarantee that a UAV can follow it safely in closed-loop execution. During flight, the UAV is subject to velocity limits, acceleration constraints, action bounds, inertia-like motion trends, and local tracking errors. Directly chasing waypoints may lead to overshoot, retreat along the path, abrupt segment switching, large lateral deviation, or even collision with buildings. This problem becomes more severe in dense urban scenes, where the clearance between buildings may be small and the path curvature may change sharply.

The issue is further complicated in multi-UAV scenarios. Even when each UAV has an individually feasible static path, simultaneous execution may create inter-UAV conflicts. Static paths generated independently for multiple UAVs do not necessarily guarantee safe temporal separation. Therefore, urban multi-UAV navigation requires a complete chain that connects geometric path generation, executable trajectory construction, closed-loop reference tracking, and local conflict correction.

Reinforcement learning has been widely studied for UAV navigation, obstacle avoidance, and multi-agent coordination because it can learn closed-loop decision policies from interaction data [4–7]. However, end-to-end reinforcement learning remains difficult to apply directly to dense urban multi-UAV navigation. Without geometric priors, the policy must simultaneously discover feasible corridors, learn goal-reaching behavior, avoid buildings, maintain inter-UAV separation, and satisfy motion constraints. This usually leads to unstable actor–critic training, high sample demand, difficult reward design, weak interpretability, and unsafe early exploration. These problems are particularly undesirable in safety-critical low-altitude applications.

A more reliable strategy is to combine geometric planning, reference tracking, demonstration-guided initialization, and residual reinforcement learning. In such a hierarchical framework, geometric planning provides global spatial structure, reference tracking transforms static paths into executable flight processes, successful demonstrations initialize the policy near verified safe behavior, and reinforcement learning is used only for bounded local correction. This reduces the burden on the learned policy and avoids asking reinforcement learning to rediscover the entire navigation pipeline from scratch.

The central question addressed in this paper is therefore not simply whether reinforcement learning can replace path planning. Rather, the question is how to construct an interpretable and executable navigation chain from static geometric paths to closed-loop multi-UAV trajectories, and how to introduce reinforcement learning without destroying the verified reference baseline. In nominal tasks, a well-designed reference-tracking controller may already provide strong performance. Therefore, the value of residual reinforcement learning should be examined especially under conflict-stress conditions, where pure reference execution may produce inter-UAV conflicts.

This paper proposes a hierarchical planning-and-learning framework for urban multi-UAV navigation. The framework first generates static three-dimensional reference paths in an urban building environment. It then converts the static paths into executable reference processes using a damped reference-tracking controller with local path association, monotonic progress updating, path recapture, look-ahead guidance, and bounded action construction. Successful pure-reference executions are used for behavior-cloning initialization. Finally, a conflict-aware residual TD3 module provides bounded local corrections around the verified reference action.

The main contributions of this paper are summarized as follows.

- (i) Executable reference trajectory construction. This paper formulates urban multi-UAV navigation as a transformation from static geometric paths to dynamically executable reference processes. The proposed framework explicitly separates geometric path generation, reference execution, and residual correction.
- (ii) Damped reference-tracking mechanism and ablation validation. A full reference-tracking controller is developed by integrating local path association, monotonic progress updating, path recapture, look-ahead guidance, and bounded damped action construction. Tracking-ablation experiments show that simplified tracking controllers fail to execute static paths reliably, while the full controller achieves a 91.67% all-success rate and eliminates building collision episodes.
- (iii) Conflict-aware residual TD3 refinement. A bounded residual TD3 module is introduced on top of the verified executable reference baseline. In constructed head-on and crossing conflict-stress tests, the residual policy substantially reduces inter-UAV collision episodes compared with pure reference tracking.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 formulates the urban multi-UAV navigation problem. Section 4 presents the proposed hierarchical framework. Section 5 describes the experimental setup and results. Section 6 discusses the role of reference tracking, the value and limitations of residual learning, and runtime scope. Section 7 concludes the paper. An overview of the urban scenario and the generated static reference paths is shown in Figure 1.

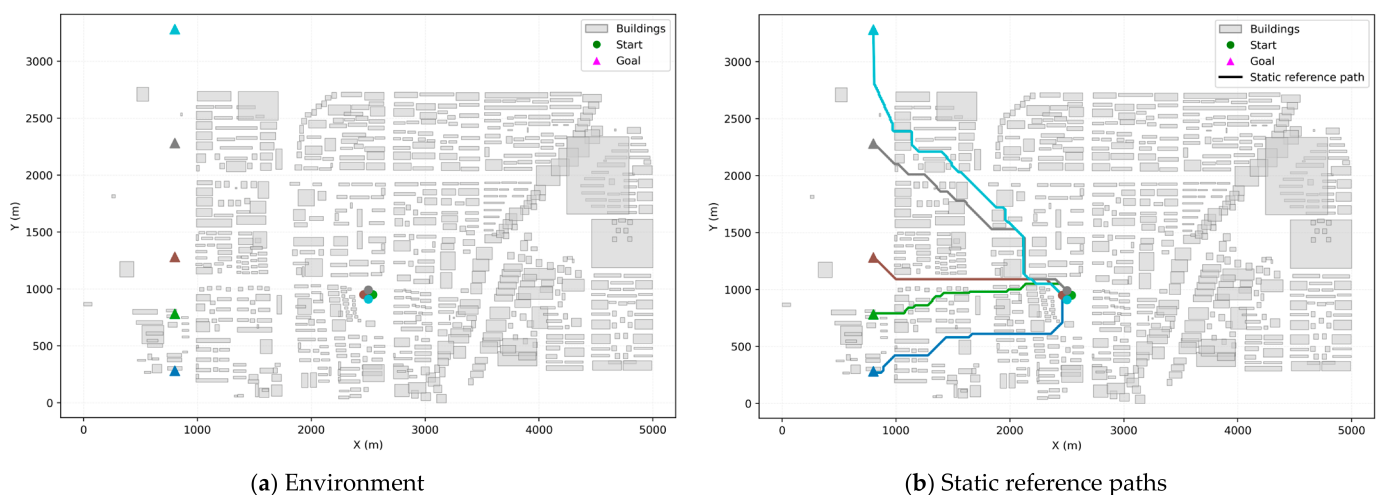


Figure 1. Overview of the urban multi-UAV navigation scenario and static reference paths. Gray rectangles denote building footprints, green markers denote start positions, magenta markers denote goal positions, and colored solid lines denote static reference paths.

2. Related Work

2.1. UAV Path Planning in Urban Environments

UAV path planning aims to generate feasible routes from start positions to goal positions while avoiding obstacles and satisfying operational constraints. In urban low-altitude environments, the feasible space is strongly constrained by buildings, altitude restrictions, and limited aerial corridors. Existing methods can be broadly classified into graph-search methods, sampling-based methods, optimization-based methods, and learning-based methods.

Graph-search methods such as Dijkstra, A*, D*, and their variants are widely used because of their clear structure and strong geometric interpretability. By discretizing the environment into grids or graphs, these methods can explicitly account for occupied regions and generate reproducible obstacle-avoiding paths. However, their computational burden may increase rapidly in high-resolution three-dimensional maps, and the output is usually a discrete path that requires further processing before execution.

Sampling-based methods such as rapidly exploring random trees and RRT* can handle high-dimensional spaces and non-convex constraints [4,6]. They are flexible in continuous environments, but their convergence may be slow in narrow urban corridors, and the generated paths may be jagged or dynamically inconsistent without additional smoothing or trajectory optimization.

Optimization-based trajectory planners, including B-spline planning, model predictive control, and direct trajectory optimization, can generate smoother and dynamically constrained trajectories [8–11]. These methods are powerful when the environment and constraints are well modeled, but they may become computationally expensive in dense multi-UAV scenarios. In addition, centralized optimization can suffer from scalability issues when many UAVs interact.

Many of these methods focus mainly on geometric feasibility. They can generate paths that avoid buildings, but they do not necessarily guarantee that the paths can be executed by closed-loop UAV motion under speed, acceleration, and tracking constraints. This gap between geometric planning and executable navigation is one of the main motivations of this paper.

2.2. Multi-UAV Coordination and Conflict Avoidance

Multi-UAV navigation requires both individual path feasibility and fleet-level coordination. Existing approaches include centralized planning, prioritized planning, conflict-based search, velocity-obstacle methods, reciprocal collision avoidance, distributed model predictive control, and multi-agent reinforcement learning [12–15].

Centralized planning can explicitly model inter-UAV coupling, but the computational complexity increases rapidly with the number of UAVs. Prioritized planning assigns an execution order or right-of-way rule to reduce conflicts, but its performance depends heavily on the priority design. Velocity-obstacle and reciprocal avoidance methods are effective for local collision avoidance but usually require additional mechanisms to handle dense building constraints. Model predictive control can account for short-horizon constraints, but repeated online optimization may be costly.

In this paper, static paths are generated independently for each UAV and guarantee building-level geometric feasibility. Inter-UAV conflicts are not assumed to be fully resolved at the static path-generation stage. Instead, they are evaluated during reference execution and further addressed through conflict-aware residual correction.

2.3. Reference Tracking and Executable Trajectories Generation

Reference tracking is the bridge between path planning and actual execution. A planned path becomes meaningful only when it can be transformed into reference actions that drive the UAV through the environment. Traditional tracking methods use target points, heading errors, lateral deviations, curvature, or preview information to guide vehicles along desired paths [16–18].

For UAVs in dense urban environments, reference tracking is not merely a low-level implementation detail. If reference tracking is poorly designed, even a collision-free path may fail during execution. For example, directly chasing the nearest waypoint may cause oscillation, abrupt switching, or retreat to previous path segments. In three-dimensional urban scenes, such behavior can easily lead to building collision or failure to reach the goal.

Therefore, the present study treats executable reference tracking as a central component of the navigation framework. The proposed controller converts a static path into a closed-loop reference process using local association, monotonic progress, path recapture, and damped action construction.

2.4. Reinforcement Learning for UAV Navigation

Deep reinforcement learning has been applied to UAV path planning, obstacle avoidance, formation control, and multi-agent navigation [19–23]. Methods such as DDPG, TD3, SAC, PPO, MADDPG, and MAPPO have been investigated in various continuous-control and multi-agent tasks.

Despite its potential, end-to-end reinforcement learning is difficult in dense urban multi-UAV navigation. The agent must learn global routing, local obstacle avoidance, inter-UAV separation, and motion feasibility simultaneously. Reward terms for progress, safety, smoothness, and reference preservation may conflict. In addition, random exploration in early training often produces unsafe behavior, and actor–critic methods may suffer from value-function overestimation or instability.

These limitations motivate the use of reinforcement learning as a residual correction mechanism rather than as a full replacement for planning and tracking.

2.5. Residual Reinforcement Learning and Demonstration-Guided Learning

Residual reinforcement learning allows a learned policy to output a correction term on top of an existing controller or reference policy. This design can preserve the structure and stability of the baseline while enabling adaptation to local conditions. Demonstration-guided learning, including behavior cloning and learning from demonstration, can further improve training stability by initializing the policy near successful behavior [24–26].

For urban multi-UAV navigation, residual learning is meaningful only if the baseline behavior is executable. If the reference execution process is unstable, a residual policy cannot reliably correct it. Therefore, this paper first constructs a verified executable reference baseline, then uses successful executions for behavior-cloning initialization, and finally trains a bounded residual TD3 policy for local conflict correction.

3. Problem Statement and Formulation

Urban multi-UAV navigation is considered in a bounded three-dimensional low-altitude airspace containing dense building obstacles. The purpose of this section is to define the environment, the multi-UAV navigation task, the adopted motion model, and the safety constraints that must be satisfied during closed-loop execution. Unlike purely geometric path-planning problems, the problem studied here emphasizes the transformation from static collision-free paths to executable flight processes under motion and safety constraints.

3.1. Urban Environment Representation

The navigation environment is represented as a bounded three-dimensional space denoted by

$$X = [x^{\min}, x^{\max}] \times [y^{\min}, y^{\max}] \times [z^{\min}, z^{\max}] \quad (1)$$

Within this space, urban buildings are modeled as a set of static cuboid obstacles,

$$B = \{B_m\}_{m=1}^M \quad (2)$$

where M is the total number of buildings. The m -th building is represented by its horizontal footprint and height range:

$$B_m = [x^{\min}, x^{\max}] \times [y^{\min}, y^{\max}] \times [z^{\min}, z^{\max}] \quad (3)$$

The free navigation space is therefore defined as the part of X that is not occupied by buildings:

$$F = \frac{X}{\cup_{m=1}^M B_m} \quad (4)$$

A UAV trajectory is considered geometrically feasible with respect to buildings only if all its sampled positions remain inside F . In this study, the static reference path-generation stage uses this building representation to provide a global obstacle-avoiding geometric prior. However, a geometrically feasible static path is not yet treated as an executable trajectory because the UAV still needs to follow it under motion constraints and closed-loop tracking errors.

3.2. Multi-UAV Navigation Task

Assume that N UAVs operate simultaneously in the same urban airspace. For UAV i , the start position is denoted by $s_i = [x_i^s, y_i^s, z_i^s]^T$ and the goal position is denoted by $g_i = [x_i^g, y_i^g, z_i^g]^T$. The complete start and goal sets are denoted by S and G , respectively. The navigation task is to generate an executable flight process for each UAV such that all UAVs reach their corresponding goals while avoiding buildings, remaining inside the permitted airspace, and maintaining sufficient separation from other UAVs.

The input to the problem includes the building set B , the start set S , the goal set G , the number of UAVs N , the simulation time step Δt , the maximum speed $v_{\max}$, the maximum acceleration $a_{\max}$, the inter-UAV collision threshold d_{coll} , the hard-safety threshold d_{hard} , the soft-safety threshold d_{safer} and the goal-reaching tolerance ϵ_g .

The output of the planning-and-execution framework consists of two levels. The first level is the static reference path set, denoted by Π^{ref} , where each π_i^{ref} is a building-avoiding geometric path for UAV i . The second level is the closed-loop execution process, represented by the executed action sequence A^{exec} :

$$\Pi^{\text{ref}} = \left\{ \pi_i^{\text{ref}} \right\}_{i=1}^N \quad (5)$$

$$A^{\text{exec}} = \left\{ a_{i,t}^{\text{exec}} \right\}_{i=1,t=0}^{N,T} \quad (6)$$

The distinction between Π^{ref} and A^{exec} is important. The former describes where the UAV should generally fly in the geometric sense, whereas the latter determines how the UAV actually moves under the adopted dynamics and controller. Therefore, a static reference path is treated only as a geometric planning result, while the executed action sequence represents the actual closed-loop flight process.

3.3. Motion Model

Each UAV is modeled as a three-dimensional point-mass system at the navigation level. Let $p_{i,t} \in \mathbb{R}^3$, $v_{i,t} \in \mathbb{R}^3$ and $u_{i,t} \in [-1, 1]^3$ denote the position, velocity, and normalized control input of UAV i at time step t , respectively. The physical acceleration is obtained by scaling the normalized action:

$$a_{i,t} = a_{\max} u_{i,t} \quad (7)$$

The velocity is updated according to

$$v_{i,t+1} = \text{clip}(v_{i,t} + a_{i,t}\Delta t, -v_{\max}, v_{\max}) \quad (8)$$

and the position is updated as

$$p_{i,t+1} = p_{i,t} + v_{i,t+1}\Delta t \quad (9)$$

The function $\text{clip}(\cdot)$ limits each velocity component within the admissible range. This simplified point-mass model does not attempt to describe full attitude dynamics, actuator response, or low-level flight-control loops. Instead, it provides a navigation-level abstraction for evaluating whether a planned path can be transformed into a dynamically executable flight process under bounded velocity and acceleration.

This abstraction is sufficient for the purpose of this paper because the main research question is not low-level flight control, but the relationship among static reference path generation, executable reference tracking, and residual local correction in urban multi-UAV navigation.

3.4. Feasibility and Safety Requirements

During execution, each UAV must satisfy three types of requirements: environmental feasibility, motion feasibility, and multi-UAV safety.

Environmental feasibility requires that the UAV remain inside the navigation boundary and avoid all building obstacles. Therefore, for all UAVs and all time steps,

$$p_{i,t} \in X \quad (10)$$

$$p_{i,t} \notin \bigcup_{m=1}^M B_m \quad (11)$$

Motion feasibility is imposed by the bounded control and velocity model described above. The normalized action must remain in the admissible action space,

$$u_{i,t} \in [-1, 1]^3 \quad (12)$$

and the velocity is limited by the clipping operation in the state-update equation.

For multi-UAV safety, the distance between any pair of UAVs must remain above the collision threshold. A binary inter-UAV collision is recorded if

$$\|p_{i,t} - p_{j,t}\|_2 \geq d_{\text{coll}}, \forall i < j, t \quad (13)$$

In addition to this binary collision criterion, two larger safety thresholds are used to evaluate the quality of separation. The hard-safety threshold d_{hard} represents the minimum acceptable operational safety margin:

$$\|p_{i,t} - p_{j,t}\|_2 \geq d_{\text{hard}}, \forall i < j, t \quad (14)$$

The soft-safety threshold d_{safe} is used as a desired separation margin in reward shaping and performance evaluation:

$$\|p_{i,t} - p_{j,t}\|_2 \geq d_{safe}, \forall i < j, t \quad (15)$$

Violating d_{safe} does not necessarily mean that the episode fails, but it indicates that the UAVs are operating in a risk-sensitive proximity region. By contrast, violating d_{coll} is treated as a collision event.

A UAV is considered to have completed its task successfully if it reaches its goal within the prescribed tolerance:

$$\|p_{i,T_i} - g_i\|_2 \leq \epsilon_g \quad (16)$$

An episode is considered fully successful only when all UAVs reach their goals without building collision, inter-UAV collision, or boundary violation.

3.5. Objective of the Hierarchical Navigation Problem

The objective of the proposed framework is not only to find static geometric paths, but also to generate closed-loop executable trajectories that preserve safety and task completion. Conceptually, the problem can be written as finding reference paths and executable actions that maximize task success and safety while limiting path deviation and control effort:

$$\max_{\Pi^{ref}, A^{exec}} J = J_{task} - \lambda_b J_{building} - \lambda_u J_{inter} - \lambda_r J_{ref} - \lambda_c J_{control} \quad (17)$$

Here, J_{task} represents goal-reaching performance, $J_{building}$ penalizes building collision or insufficient building clearance, J_{inter} penalizes inter-UAV collision and unsafe proximity, J_{ref} penalizes excessive deviation from the reference path, and $J_{control}$ penalizes large or unstable control actions. The coefficients λ_b , λ_u , λ_r , and λ_c balance the relative importance of different terms.

This objective is implemented through the reference-tracking controller and the residual reinforcement learning reward design described in Section 4. The formulation emphasizes that the static path, the reference-tracking process, and the learned residual correction are parts of the same navigation pipeline rather than isolated modules.

3.6. Scope of Static Path Generation and Residual Correction

It is important to clarify the scope of each component in the proposed framework. The static path-generation stage is responsible for producing building-avoiding geometric paths. In the current implementation, these paths are generated independently for each UAV. Therefore, the static paths guarantee only individual building-level geometric feasibility; they do not by themselves guarantee inter-UAV temporal separation during simultaneous execution.

The reference-tracking stage then converts these static paths into executable closed-loop reference processes. This stage addresses the gap between geometric path feasibility and execution feasibility. However, even a well-designed reference tracker may still encounter inter-UAV conflicts when multiple UAVs execute their paths simultaneously.

The residual reinforcement learning stage is introduced for this reason. It is not intended to replace the static planner or the reference tracker. Instead, it provides bounded local corrections around the verified reference action, especially under head-on or crossing conflict conditions. Thus, the overall problem addressed in this paper can be understood as a hierarchical navigation problem: static paths provide geometric reachability, reference tracking provides executable baseline behavior, and residual learning provides conflict-aware local refinement.

4. Proposed Method

Urban multi-UAV navigation in dense building environments requires a method that can preserve the global geometric structure of planned routes while producing dynamically executable actions under closed-loop interaction. The proposed framework follows a hierarchical planning-and-learning design. A static planner first generates building-avoiding reference paths, a conservative tracking controller then converts these paths into executable reference actions, and a residual reinforcement learning module finally provides bounded local corrections around the reference behavior. The executed action of UAV i at time step t is written as

$$a_{i,t}^{exec} = a_{i,t}^{ref} + \Delta a_{i,t} \quad (18)$$

where $a_{i,t}^{ref}$ is generated by the reference tracker and $\Delta a_{i,t}$ is the learned residual correction. This decomposition allows the geometric planner and reference tracker to provide a stable execution baseline, while the learning component is restricted to local refinement under interaction or conflict conditions.

4.1. Hierarchical Planning-And-Learning Framework

The overall framework consists of static reference path generation, executable reference tracking, demonstration-guided initialization, and residual TD3 refinement. These stages form a continuous navigation chain from geometric planning to closed-loop execution. The static planner works in the three-dimensional urban map and generates one building-avoiding reference path for each UAV. The reference tracker receives this path and the current UAV state, associates the UAV with a local path segment, and outputs a bounded action that drives the UAV along the reference path. Successful pure-reference executions are then collected as demonstration data for actor initialization. After this initialization, residual TD3 is trained to produce local corrections around the reference action. The overall hierarchical workflow is shown in Figure 2.

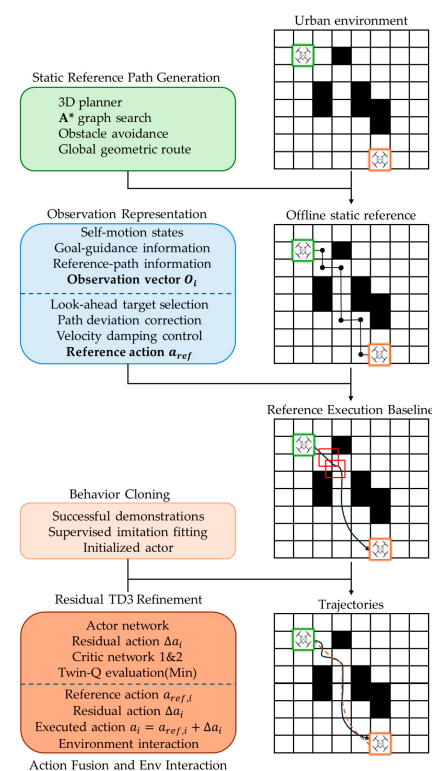


Figure 2. Overall hierarchical framework of static reference paths, reference execution, successful demonstration distillation, and residual reinforcement learning. A* denotes the A-star graph-search

algorithm. White squares denote free grid cells, black squares denote obstacle-occupied cells, black dots denote reference nodes, dashed lines denote the reference execution baseline, and red squares highlight local conflict.

The framework separates global structure from local correction. Static path generation determines the main spatial route, while reference tracking determines how the UAV follows that route under the adopted motion model. The learned residual policy does not directly output the entire control action. Instead, it modifies the reference action within a bounded range. During execution, the tracker and the actor are evaluated at every time step, and the final action is obtained by fusing the reference action and the residual correction. Static path generation, demonstration collection, behavior cloning, and residual TD3 training are performed offline; online execution only requires reference tracking, local safety-feature calculation, actor inference, and action fusion.

4.2. Static Reference Path Generation and Executable Tracking

The static reference path is generated by a three-dimensional grid-based heuristic search. The urban space is discretized into grid nodes, and nodes located inside buildings or outside the navigation boundary are removed from the feasible graph. A 26-neighbor connection structure is adopted so that the generated path can include horizontal, vertical, and diagonal movements. For a search node n , the A* evaluation function is

$$f(n) = g(n) + h(n) \quad (19)$$

where $h(n)$ is the Euclidean heuristic distance from n to the goal, and $g(n)$ is the accumulated cost. The accumulated cost includes path length, altitude variation, turning behavior, and building proximity:

$$g(n) = g_{dist}(n) + \lambda_z g_{alt}(n) + \lambda_\theta g_{turn}(n) + \lambda_b g_{near}(n) \quad (20)$$

Here, g_{dist} denotes the distance cost, g_{alt} penalizes unnecessary altitude changes, g_{turn} penalizes sharp direction changes, and g_{near} penalizes flying too close to buildings. After a candidate path is obtained, point-level and segment-level collision checks are applied. Point-level checking ensures that all reference nodes remain outside buildings, while segment-level checking prevents a line segment between two collision-free nodes from passing through a building interior. The output reference path for UAV i is denoted by

$$\pi_i^{ref} = \{r_{i,0}, r_{i,1}, \dots, r_{i,K}\} \quad (21)$$

where $r_{i,k} \in \mathbb{R}^3$ is the k -th reference point.

The reference path is then converted into an executable process by a damped tracking controller. For each UAV, the tracker maintains a progress index $q_{i,t}$, which records the current advancement along the path. Instead of searching for the nearest point over the full reference path, the tracker searches within a local window $S(q_{i,t}, W)$ around the current progress index. The projection point is computed as

$$r_{i,t}^{proj} = \arg \min_{r \in S(q_{i,t}, W)} \|p_{i,t} - r\|_2 \quad (22)$$

and the corresponding reference deviation is

$$d_{i,t}^{ref} = \|p_{i,t} - r_{i,t}^{proj}\|_2 \quad (23)$$

The target point is selected according to the deviation from the reference path. If the deviation is larger than the recapture threshold η , the local projection point is used as the target; otherwise, a look-ahead point is used to maintain forward progress:

$$r_{i,t}^{tar} \begin{cases} r_{i,t}^{proj}, d_{i,t}^{ref} > \eta \\ r_{i,t}^{look}, d_{i,t}^{ref} \leq \eta \end{cases} \quad (24)$$

The tracking error is

$$e_{i,t} = r_{i,t}^{tar} - p_{i,t} \quad (25)$$

A damped action is generated from the position error and recent motion trend:

$$a_{i,t}^{ref} = K_p e_{i,t} - K_d (p_{i,t} - p_{i,t-1}) \quad (26)$$

where K_p is the proportional gain and K_d is the damping gain. The horizontal and vertical components are clipped separately to reflect their different motion characteristics:

$$a_{i,t}^{ref,xy} = clip(a_{i,t}^{ref,xy}, -a_{max}^{xy}, a_{max}^{xy}) \quad (27)$$

$$a_{i,t}^{ref,z} = clip(a_{i,t}^{ref,z}, -a_{max}^z, a_{max}^z) \quad (28)$$

The final reference action is obtained by combining the clipped horizontal and vertical components. This controller preserves monotonic path progress, avoids attraction to already passed segments, and reduces oscillation near turns through damping. As a result, the static geometric path is transformed into a closed-loop reference process that can be executed under the adopted motion model. The geometric construction of executable reference tracking and the corresponding reference-action construction are illustrated in Figure 3.

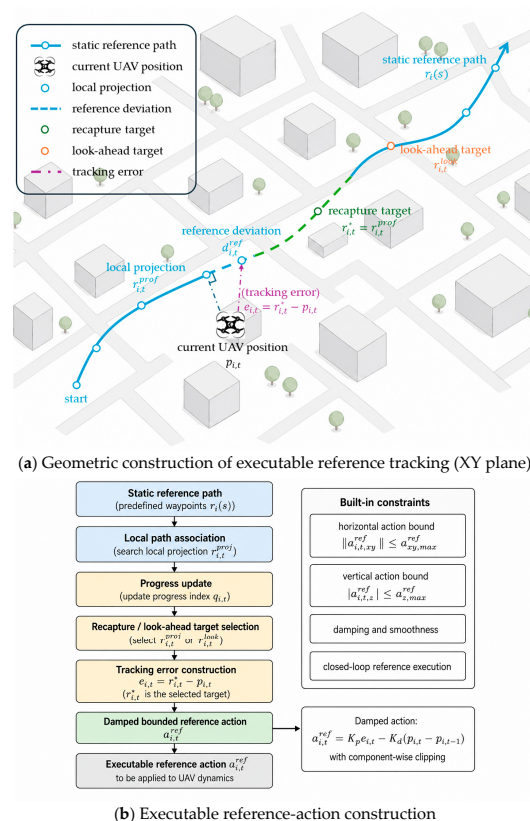


Figure 3. Executable reference tracking mechanism.

4.3. Demonstration-Guided Residual TD3

After constructing the reference tracker, pure reference replay is performed by executing

$$a_{i,t}^{exec} = a_{i,t}^{ref} \quad (29)$$

Episodes in which all UAVs reach their goals without building collision, inter-UAV collision, or boundary violation are stored as successful demonstrations. Each demonstration transition contains the observation, the reference action, the next observation, and the termination flag:

$$(o_t, a_t^{ref}, o_{t+1}, d_t) \quad (30)$$

The successful demonstration dataset is denoted by D_{succ} . The actor π_θ is initialized by minimizing the behavior-cloning loss

$$L_{BC} = E_{(o_t, a_t^{ref}) \sim D_{succ}} [\|\pi_\theta(o_t) - a_t^{ref}\|_2^2] \quad (31)$$

This initialization places the actor near the executable reference-action distribution before reinforcement learning begins. During residual learning, the actor output is scaled and bounded as

$$\Delta a_{i,t} = clip(\beta \pi_\theta(o_{i,t}), -\delta, \delta) \quad (32)$$

where β is the residual scaling coefficient and δ is the residual bound. The executed action is

$$a_{i,t}^{exec} = clip(a_{i,t}^{ref} + \Delta a_{i,t}, -1, 1) \quad (33)$$

The residual policy network architecture and the actor–critic network used in residual TD3 are illustrated in Figure 4.

The critic evaluates the fused executed action rather than the raw residual output. Therefore, the two critics are expressed as

$$Q_1(o_t, a_t^{exec}) \quad (34)$$

$$Q_2(o_t, a_t^{exec}) \quad (35)$$

The next-step target action is constructed in the same fused-action form,

$$a_{t+1}^{exec} = a_{t+1}^{ref} + \Delta a_{t+1} \quad (36)$$

The TD3 target value is

$$y_t = r_t + \gamma(1 - d_t) \min_{k=1,2} Q'_k(o_{t+1}, a_{t+1}^{exec}) \quad (37)$$

where Q' is the target critic and γ is the discount factor. The critic loss is

$$L_Q = E[\sum_{k=1}^2 (Q_k(o_t, a_t^{exec}) - y_t)^2] \quad (38)$$

The actor is updated by maximizing the value of the fused action while retaining behavior-cloning regularization:

$$L_\pi = -E[Q_1(o_t, a_t^{exec})] + \lambda_{BC} L_{BC} \quad (39)$$

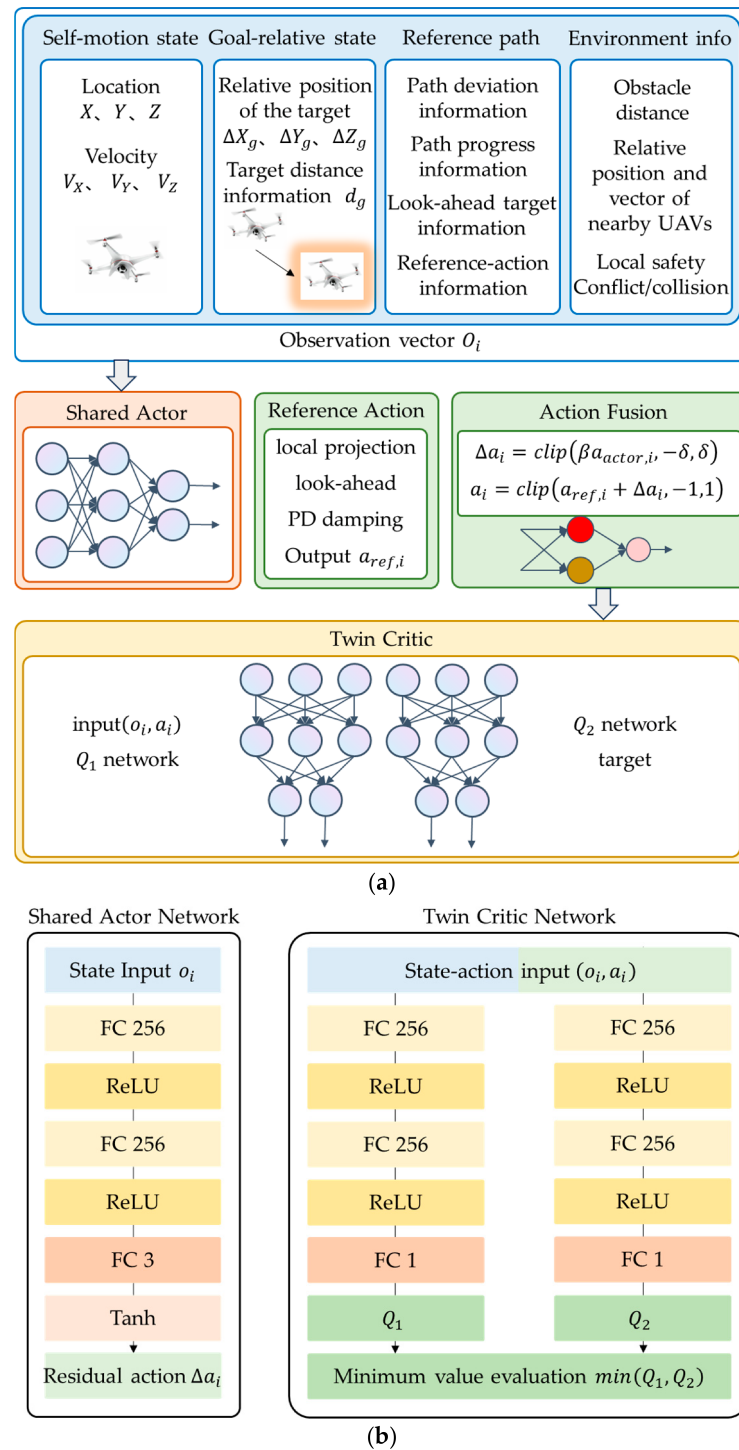


Figure 4. (a) Architecture of the Residual Policy Network. (b) Actor and critic network.

To capture inter-UAV interaction, the observation includes relative-state features of nearby UAVs. For a UAV pair (i, j) , the relative position and velocity are

$$p_{ij,t} = p_{j,t} - p_{i,t} \quad (40)$$

$$v_{ij,t} = v_{j,t} - v_{i,t} \quad (41)$$

The closing speed is calculated as

$$c_{ij,t} = -\frac{p_{ij,t}^T v_{ij,t}}{\|p_{ij,t}\|_2 + \epsilon} \quad (42)$$

where ϵ is a small positive constant. A positive value of $c_{ij,t}$ indicates that the two UAVs are approaching each other. Additional features, including time-to-conflict, predicted closest point of approach, and priority information, are used in the conflict-aware setting. A priority-based safety teacher provides right-of-way guidance in symmetric head-on and crossing cases. The teacher supplies informative residual actions during training and is also evaluated as an explicit baseline.

4.4. Reward, Training Procedure, and Computational Cost

The reward function combines goal-reaching, reference-preservation, safety, and smoothness terms:

$$r_t = r_t^{goal} + r_t^{ref} + r_t^{safe} + r_t^{smooth} \quad (43)$$

The goal term rewards progress toward the destination and terminal success. The reference term encourages forward progress along the planned path and penalizes excessive deviation from the reference. The safety term penalizes building collision, inter-UAV collision, unsafe proximity, and boundary violation. The smoothness term penalizes excessive action magnitude, abrupt action changes, and unnecessary vertical oscillation. Safety penalties are assigned the largest magnitude, reference deviation and unsafe proximity are treated as intermediate penalties, and smoothness terms are used as regularization. The reward weights are scaled according to the relative roles of task completion, safety, reference preservation, and control quality.

Training starts with static path generation and pure reference replay. Successful reference executions are collected for behavior-cloning initialization. Residual TD3 is then trained on a mixture of nominal tasks and constructed conflict-stress tasks. The stress tasks include head-on and crossing cases, in which pure reference tracking may lead to inter-UAV conflicts. During each training step, the reference tracker produces a_t^{ref} , the actor produces Δa_t , and the environment executes

$$a_t^{exec} = a_t^{ref} \Delta a_t \quad (44)$$

The replay buffer stores observations, reference actions, residual actions, fused actions, rewards, next observations, and termination flags. The critics and target critics are updated using fused actions, and the actor is updated using the TD3 objective with behavior-cloning regularization. Checkpoints are selected using a fixed prioritization set containing both nominal and conflict-stress scenarios. The validation score prioritizes task success, collision avoidance, hard-safety satisfaction, and controlled reference deviation.

The computational cost of the framework is divided between offline and online stages. The grid-based A* search has complexity

$$O(|E| \log |V|) \quad (45)$$

where $|V|$ and $|E|$ are the numbers of grid nodes and edges. Segment-level collision checking has approximate complexity $O(KM)$, where K is the number of path segments and M is the number of buildings. During execution, the reference tracker searches only a local path window, giving complexity $O(NW)$, where N is the number of UAVs and W is the local window size. Pairwise inter-UAV safety checking has complexity $O(N^2)$, and actor inference has complexity $O(NP_\pi)$, where P_π is the number of actor-network parameters.

TD3 training updates have approximate complexity $O(B(P_\pi + P_Q))$, where B is the batch size and P_Q is the number of critic-network parameters. Static path generation, demonstration collection, and residual TD3 training are performed offline, while online execution only requires local reference tracking, pairwise safety checking, and actor inference.

The complete training and execution procedure is summarized in Algorithm 1.

Algorithm 1. Hierarchical reference tracking and residual TD3 training

Input: urban building map, start and goal sets, reference tracker, conflict-stress generator.

Output: trained residual policy.

1. Generate static reference paths for all UAVs.
 2. Execute pure reference replay using the reference tracker and collect successful demonstrations D_{succ} .
 3. Initialize the actor by minimizing the behavior-cloning loss on D_{succ} .
 4. Construct nominal, head-on, and crossing training scenarios.
 5. For each training episode, compute a_i^{ref} infer Δa_i execute $a_i^{exec} = a_i^{ref} + \Delta a_i$, and store the transition.
 6. Update the twin critics using fused executed actions.
 7. Update the actor using the TD3 objective with behavior-cloning regularization.
 8. Periodically evaluate the policy on fixed nominal and conflict-stress validation tasks.
 9. Save the checkpoint with the best safety-aware validation score.
-

5. Experiments and Results

5.1. Experimental Setup and Evaluation Protocol

The proposed framework was evaluated in an OpenStreetMap-based urban environment containing 754 cuboid buildings. The environment represents a dense low-altitude urban navigation scene in which UAVs must fly through narrow building-constrained spaces while maintaining safe separation from other UAVs. Five UAVs were considered in the main experiments. Each UAV was assigned a start position and a goal position corresponding to an urban dispatch or delivery task. All UAVs followed the three-dimensional point-mass model defined in Section 3, and all methods were evaluated under the same map, task definitions, motion limits, and safety thresholds.

The default maximum speed was set to 20 m/s. To examine the influence of the speed setting, additional speed-sensitivity tests were conducted at 10 m/s and 15 m/s. The inter-UAV collision threshold was set to 4 m, the hard-safety threshold was set to 8 m, and the soft-safety threshold was set to 25 m. An episode was regarded as fully successful only if all UAVs reached their goals without building collision, inter-UAV collision, or boundary violation. The maximum episode length was 4000 simulation steps. The main simulation parameters are summarized in Table 1.

The compared methods were selected to examine different parts of the proposed navigation chain. Naive waypoint tracking, global projection tracking, monotonic projection tracking, and recapture-based projection tracking were used to test whether static paths can be executed without the full reference-tracking mechanism. The full-damped reference tracker and pure reference replay were used to evaluate the executable reference baseline. BC-only, residual DDPG, the priority-based safety teacher, and conflict-aware residual TD3 were then compared to examine the effect of demonstration-guided residual learning. The main evaluation metrics included all-success rate, Individual-UAV success rate, final three-dimensional error, reference deviation, collision episode rate, building collision rate, inter-UAV collision rate, minimum inter-UAV separation, and hard-safety satisfaction rate.

Table 1. Main simulation parameters.

Parameter	Value
Number of buildings	754
Number of UAVs	5
Motion model	3D point mass
Default maximum speed	20 m/s
Speed-sensitivity settings	10, 15, 20 m/s
Inter-UAV collision threshold	4 m
Hard-safety threshold	8 m
Soft-safety threshold	25 m
Maximum episode length	4000 steps

5.2. Executable Reference Tracking Analysis

The first group of experiments examined whether the static reference paths generated by the planner could be directly executed by simple tracking strategies. All tracking variants used the same static paths and the same UAV motion model; they differed only in how the current UAV state was associated with the reference path and how the reference action was generated. The results are shown in Table 2.

Table 2. Reference-tracking ablation results.

Method	All-Success Rate	Individual-UAV Success Rate	Collision Episode Rate	Building Collision Rate	Mean Final 3D Error (m)	Mean Reference Deviation (m)
Naive waypoint	0.00%	0.00%	100.00%	100.00%	1457.36	35.72
Global projection P	0.00%	3.33%	100.00%	100.00%	1211.77	57.92
Monotonic P	0.00%	3.33%	100.00%	100.00%	1207.14	58.62
Recapture P	0.00%	5.00%	100.00%	100.00%	1157.59	51.61
Full-damped reference tracking	91.67%	96.67%	8.33%	0.00%	63.50	3.84

The simplified controllers failed to execute the static paths reliably. Naive waypoint tracking produced building collision episodes in all evaluated tasks, which indicates that a collision-free geometric path cannot be treated as a directly executable trajectory. Global projection and monotonic projection slightly improved UAV-level success, but both still failed at the all-UAV episode level and retained a 100% building collision rate. Recapture-based projection reduced part of the tracking error but remained insufficient for stable execution. Representative execution trajectories of different reference-tracking variants are shown in Figure 5.

By contrast, the full-damped reference tracker achieved a 91.67% all-success rate and a 96.67% individual-UAV success rate, while reducing the mean final three-dimensional error to 63.50 m and the mean reference deviation to 3.84 m. No building collision episode occurred under the full tracker. A paired task-level comparison further showed that the full controller improved the success outcome in 11 out of 12 matched trials compared with each simplified baseline. These results show that the executable flight process is produced by the tracking mechanism rather than by the static path alone.

The remaining failure of the full tracker was caused by inter-UAV conflict rather than building collision. This distinction is important because it separates two different problems. The reference tracker resolves the gap between geometric path feasibility and single-UAV executable tracking, but simultaneous execution of independently generated paths can still produce multi-UAV conflicts. This motivates the conflict-aware residual correction evaluated later in this section.

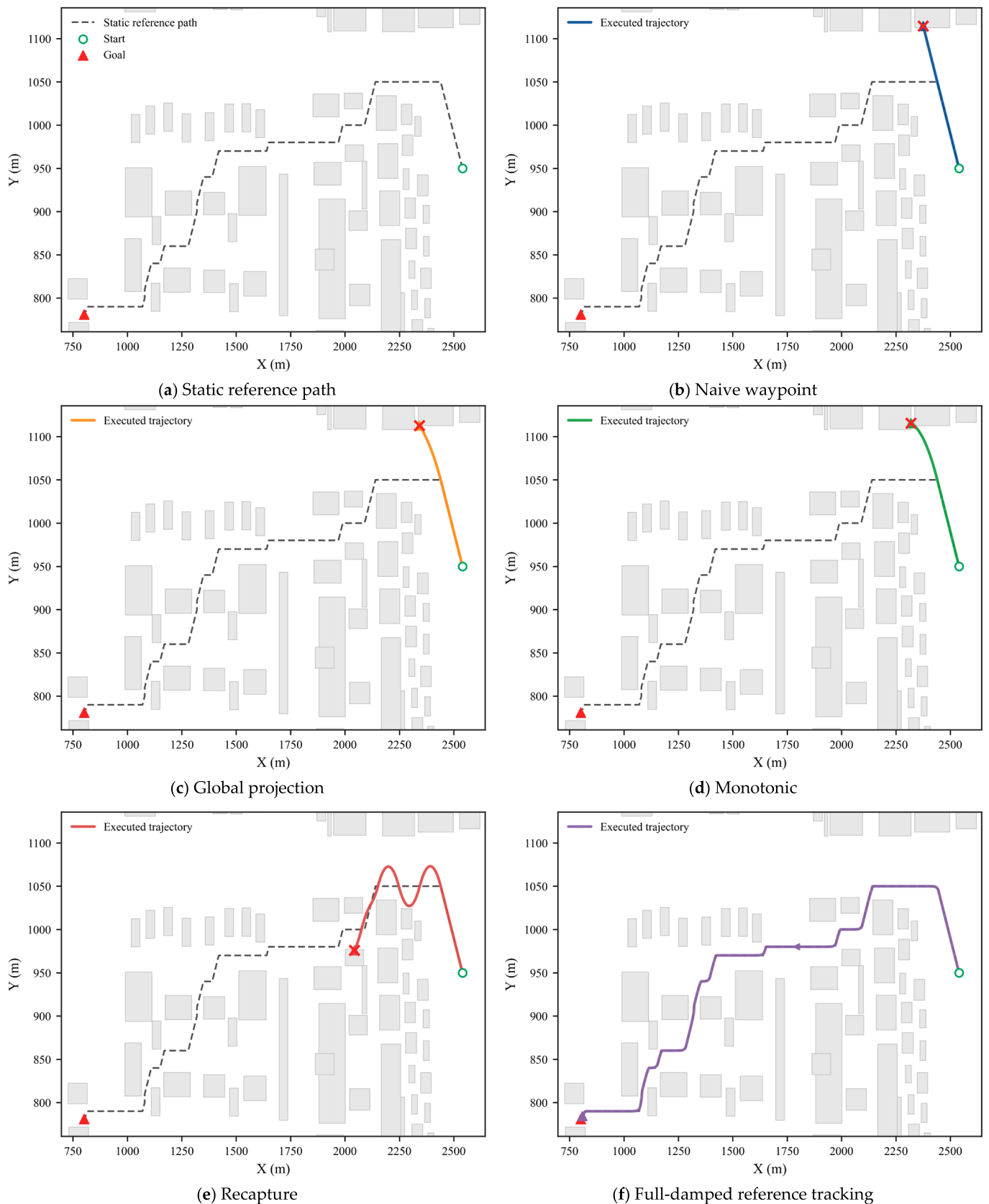


Figure 5. Representative execution trajectories of different reference-tracking variants in task 000. Gray boxes denote building footprints, red crosses indicate collision or failure positions, the dashed line denotes the static reference path, and the solid line denotes the executed trajectory of the selected UAV. Simplified tracking variants show unstable execution and early failure, whereas the full-damped reference tracker preserves path progress and reaches the goal under the adopted motion model.

The robustness of the reference-tracking result was further examined through speed-sensitivity tests. Table 3 compares naive waypoint tracking and pure reference tracking under maximum speeds of 10, 15, and 20 m/s.

Table 3. Speed-sensitivity results.

Maximum Speed	Method	All-Success Rate	Collision Episode Rate	Building Collision Rate	Mean Minimum Inter-UAV Separation (m)
10 m/s	Naive waypoint	0.00%	100.00%	100.00%	28.17
10 m/s	Pure reference	91.67%	8.33%	0.00%	32.10
15 m/s	Naive waypoint	0.00%	100.00%	100.00%	23.88
15 m/s	Pure reference	91.67%	8.33%	0.00%	32.10
20 m/s	Naive waypoint	0.00%	100.00%	100.00%	23.88
20 m/s	Pure reference	91.67%	8.33%	0.00%	32.10

Pure reference tracking maintained the same 91.67% all-success rate under all three speed settings and did not introduce building collision episodes. In contrast, naive waypoint tracking failed under every tested speed limit. The consistent performance within the tested speed range indicates that the main conclusion of the reference-tracking ablation is not caused by a single aggressive speed parameter. The unchanged failure case also suggests that the remaining bottleneck is associated with multi-UAV interaction rather than only the maximum speed bound.

5.3. Residual Correction Under Nominal and Conflict-Stress Scenarios

After establishing the executable reference baseline, the next experiments evaluated whether residual learning further improves performance. The nominal test set used the original start-goal settings without artificially constructed head-on or crossing perturbations. In this setting, pure reference tracking already provided a strong baseline, as shown in Table 4.

Table 4. Nominal test-set results.

Method	All-Success Rate	Collision Episode Rate	Inter-UAV Collision Rate	Hard-Safety Rate
Pure reference	66.67%	33.33%	33.33%	66.67%
Safety teacher	66.67%	33.33%	33.33%	66.67%
BC + residual TD3	66.67%	33.33%	33.33%	66.67%

Residual TD3 preserved the pure-reference performance but did not improve the original nominal test-set success rate. The difficult original conflict case remained unresolved. This result shows that residual learning should not be interpreted as a universal improvement over pure reference tracking in all nominal tasks. Its value is better examined under scenarios in which pure reference execution generates systematic inter-UAV conflicts.

For this reason, constructed conflict-stress tests were introduced. The stress set included head-on and crossing scenarios. In head-on scenarios, two UAVs approached each other along opposite or nearly opposite directions in a shared corridor. In crossing scenarios, two UAVs reached an intersection region at similar times. These cases were designed to activate inter-UAV conflicts that pure reference tracking cannot reliably resolve. The overall stress-test results are shown in Table 5.

Table 5. Overall conflict-stress test results.

Method	All-Success Rate	Total Collision Rate	Inter-UAV Collision Rate	Building Collision Rate	Hard-Safety Rate	Mean Minimum Separation (m)
Pure reference	33.33%	66.67%	66.67%	0.00%	33.33%	15.65
Safety teacher	80.56%	19.44%	11.11%	8.33%	87.50%	24.67
BC + residual TD3	80.09%	19.91%	11.57%	8.33%	87.04%	23.53

The paired stress-trial comparison further supported this improvement. Compared with pure reference tracking under the same stress scenarios, BC + residual TD3 improved the success outcome in 101 out of 216 paired trials, eliminated collision in 119 out of 216 paired trials, and improved hard-safety satisfaction in 116 out of 216 paired trials. These paired results indicate that the improvement was not caused by using different stress samples.

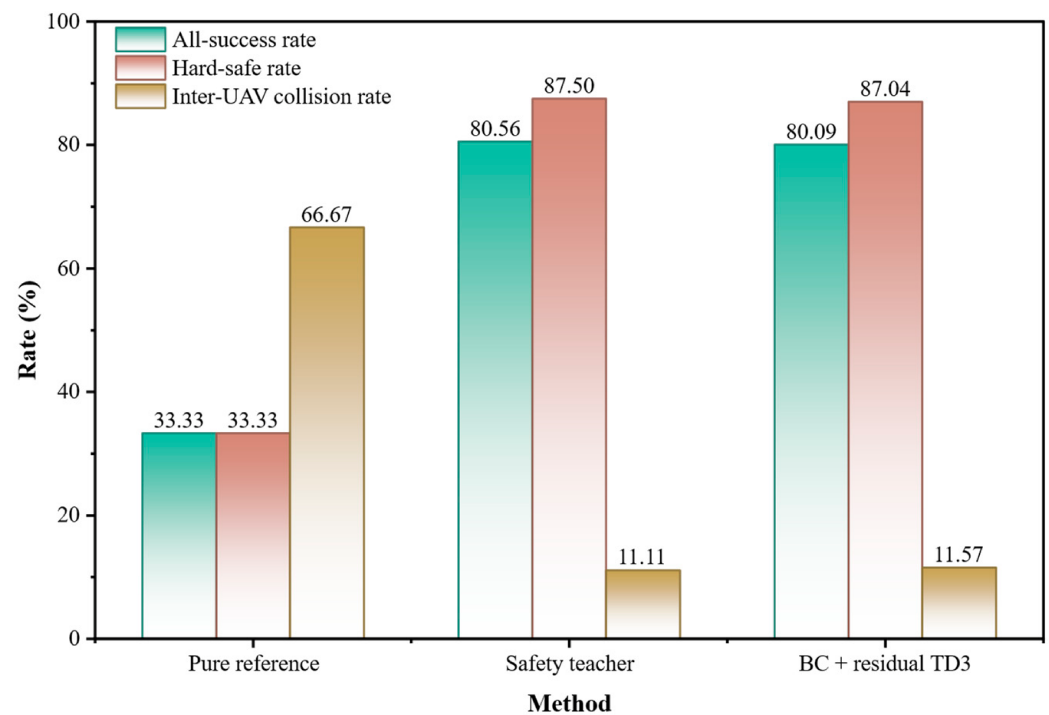
The residual TD3 result was close to the safety teacher, whose all-success rate was 80.56% and inter-UAV collision rate was 11.11%. This indicates that the learned residual policy approximated the teacher-induced conflict-resolution behavior in the constructed stress scenarios. At the same time, the building collision rate increased from 0.00% under pure reference to 8.33% under both the safety teacher and residual TD3. This shows that some avoidance corrections transferred the failure mode from inter-UAV collision to building collision, especially in building-constrained corridors. The overall conflict-stress comparison is further visualized in Figure 6.

A more detailed decomposition is given in Table 6. In head-on conflicts, pure reference tracking failed in all trials, with a 100.00% inter-UAV collision rate and a mean minimum separation of only 3.32 m. Residual TD3 reduced the inter-UAV collision rate to 1.39% and improved the hard-safety rate to 94.44%, although the total collision rate remained 22.22% because some cases failed due to building collision.

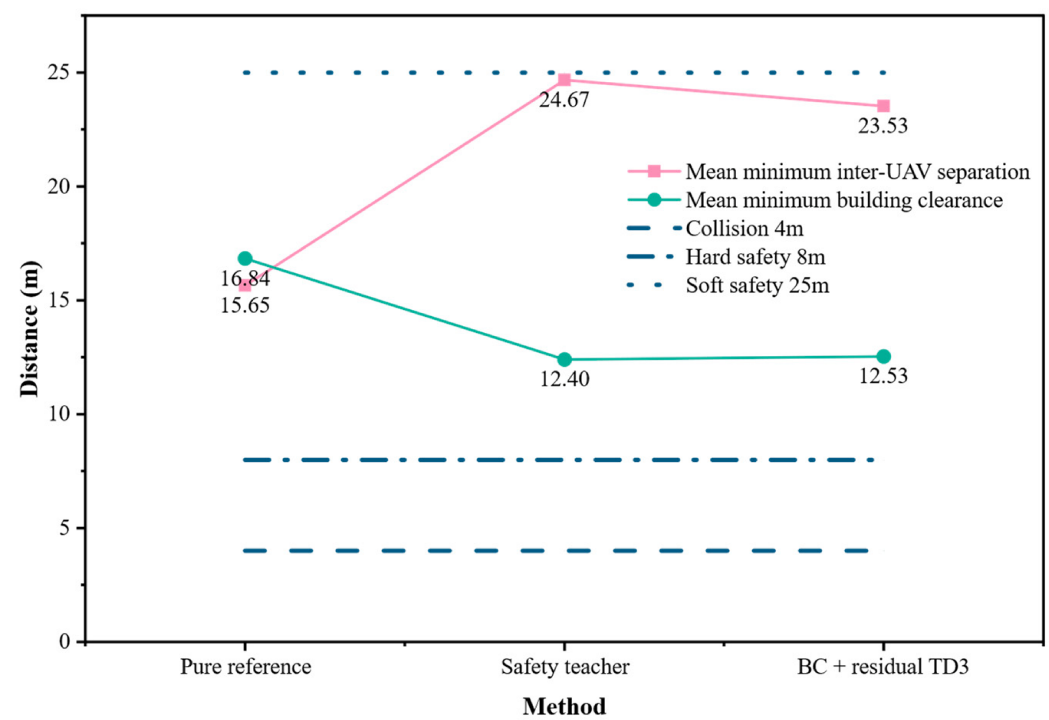
Table 6. Head-on and crossing conflict decomposition.

Scenario	Method	All-Success	Total Collision	Inter-UAV Collision	Building Collision	Hard-Safe	Mean Min Sep
Head-on	Pure reference	0.00%	100.00%	100.00%	0.00%	0.00%	3.32
Head-on	Safety teacher	79.17%	20.83%	0.00%	20.83%	95.83%	19.38
Head-on	BC + residual TD3	77.78%	22.22%	1.39%	22.22%	94.44%	17.89
Crossing	Pure reference	33.33%	66.67%	66.67%	0.00%	33.33%	9.59
Crossing	Safety teacher	95.83%	4.17%	0.00%	4.17%	100.00%	21.26
Crossing	BC + residual TD3	95.83%	4.17%	0.00%	4.17%	100.00%	21.74

In the representative crossing case, pure reference tracking resulted in an inter-UAV collision between UAV 1 and UAV 2, with the minimum separation decreasing to 3.31 m. In contrast, BC + residual TD3 increased the minimum separation to 26.08 m, satisfying the soft-safety threshold and completing all UAV tasks without building collision. A representative crossing-conflict case is shown in Figure 7.



(a) Success and safety rates



(b) Mean minimum separation

Figure 6. Overall conflict-stress comparison.

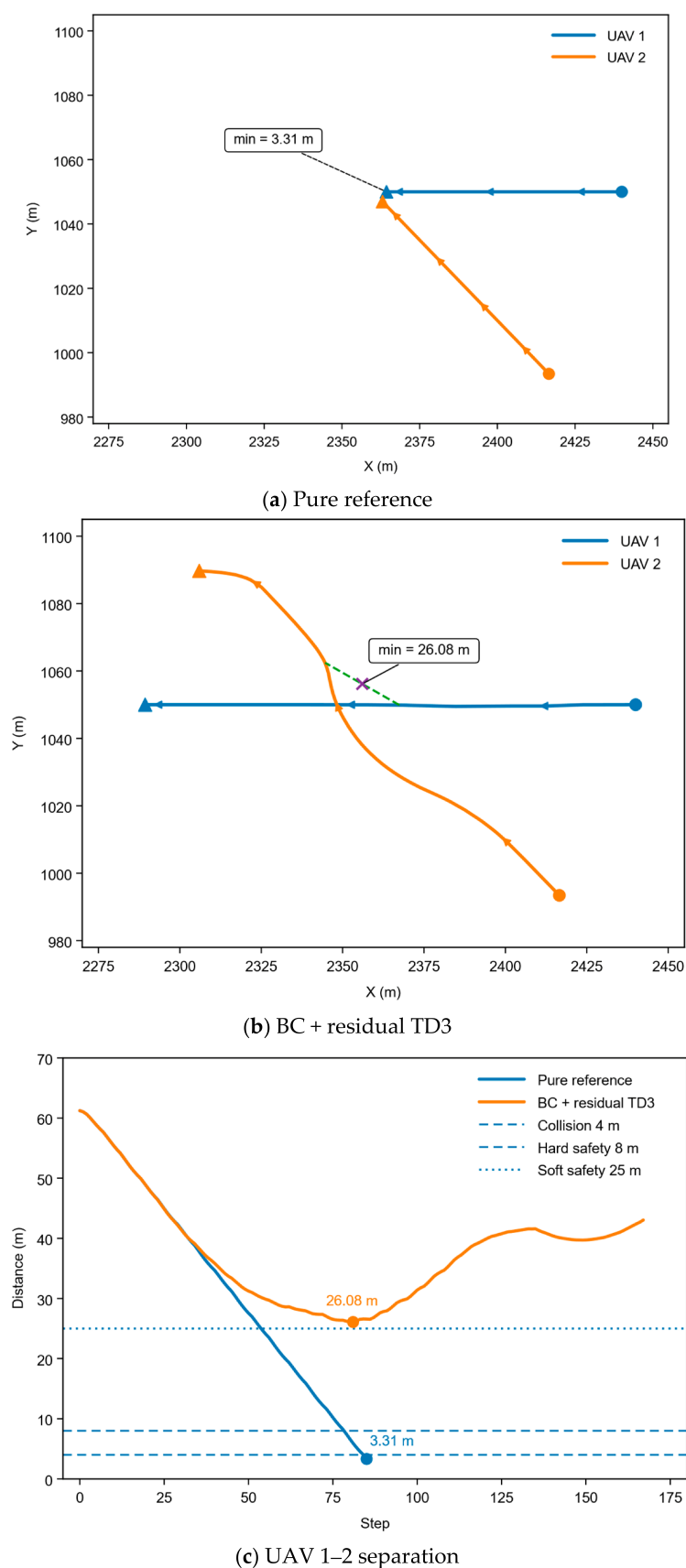


Figure 7. Representative crossing-conflict case. Pure reference tracking leads to an inter-UAV conflict between UAV 1 and UAV 2, whereas BC + residual TD3 enlarges the minimum separation and completes the task safely. Dashed horizontal lines in panel (c) denote the collision threshold, hard-safety threshold, and soft-safety threshold.

5.4. Learning Stability and Experimental Summary

The final group of experiments examined learning stability and diagnostic baselines. Residual TD3 was compared with residual DDPG under the same reference tracker, successful demonstration dataset, residual-action structure, and behavior-cloning initialization. Residual DDPG initially reproduced the reference behavior, but later became unstable as training continued. Repeated building collision termination appeared after the early training stage. Residual TD3 was more stable, which is consistent with its use of twin critics, target-policy smoothing, delayed actor updates, and fused-action value evaluation.

The training summary of the conflict-aware residual TD3 runs is shown in Table 7. Across three training seeds, the mean episode success rate reached 85.28%, the Individual-UAV success rate reached 96.87%, and the hard-safety satisfaction rate reached 94.67%. The mean minimum inter-UAV separation was 21.92 m, and no non-finite update was skipped.

Table 7. Training stability summary for conflict-aware residual TD3.

Metric	Value
Mean episode success rate	85.28%
Individual-UAV success rate	96.87%
UAV collision episode rate	4.61%
Hard-safety satisfaction rate	94.67%
Mean minimum inter-UAV separation	21.92 m
Mean critic loss	0.276
Skipped non-finite updates	0

BC-only was also retained as a diagnostic comparison. Although behavior cloning reduced supervised action error on successful demonstrations, it did not reliably preserve the reference behavior during closed-loop deployment. This is consistent with covariate shift: small prediction errors can accumulate and move the system into states that are not well represented in the demonstration dataset. For this reason, behavior cloning is used as initialization rather than as the final controller.

Overall, the experiments support a hierarchical interpretation of the proposed method. Static reference paths provide global geometric priors, but they are not directly executable. The full-damped reference tracker is necessary for transforming those paths into closed-loop reference trajectories and eliminating building collision episodes in the ablation tests. Residual TD3 adds value mainly under conflict-stress conditions, where pure reference tracking produces head-on or crossing inter-UAV conflicts. At the same time, the nominal test-set result and the remaining head-on building collision failures indicate that the current residual module is not a complete multi-UAV conflict-resolution solution. Future improvements should incorporate obstacle-aware residual projection, runtime safety shielding, and online local replanning.

6. Discussion

The experimental results show that the proposed framework should be interpreted as a hierarchical navigation pipeline rather than as a purely learning-based planner. The main difficulty in the studied urban multi-UAV task lies not only in finding collision-free geometric paths, but also in transforming these paths into stable closed-loop flight processes under motion constraints and inter-UAV interactions. The reference-tracking ablation, speed-sensitivity test, and conflict-stress evaluation together clarify the role of each component in the framework.

6.1. From Static Geometric Paths to Executable Reference Processes

The reference-tracking ablation confirms that static geometric feasibility is not sufficient for closed-loop execution. Although the planned paths avoid buildings at the map level, naive waypoint tracking and simplified projection-based tracking fail in all evaluated tasks. These failures are mainly caused by path-association errors, overshoot near turns, attraction to inappropriate path segments, and insufficient damping under the adopted velocity and acceleration constraints. This result indicates that the output of a path planner cannot be treated as an executable UAV trajectory without a carefully designed tracking mechanism.

The full-damped reference tracker substantially changes this situation. By combining local path association, monotonic progress updating, path recapture, look-ahead guidance, and damped bounded action construction, the tracker converts a discrete geometric path into a closed-loop reference process. The elimination of building collision episodes in the tracking-ablation experiment suggests that the main geometric-to-execution gap is handled by this module. The speed-sensitivity results further indicate that this conclusion is not tied to a single maximum-speed setting within the tested range of 10–20 m/s.

This finding has a broader implication for urban UAV path-planning studies. Evaluating only the geometric quality of a path, such as length, smoothness, or static obstacle clearance, may overestimate the practical value of a method. A path must also be tested under a closed-loop execution model. In dense urban scenes, where small deviations can lead to collision with buildings, executable tracking should be considered part of the planning problem rather than a downstream implementation detail.

6.2. Role of Residual TD3 in Conflict-Aware Local Correction

The residual TD3 module does not replace the reference tracker. Its effect depends on whether the reference execution leaves a local correction opportunity. In the nominal test set, pure reference tracking already provides a strong executable baseline, and residual TD3 does not improve the original difficult conflict case. This result shows that residual learning is not a general-purpose performance booster for all tasks. If the reference baseline already succeeds, the bounded residual policy mainly preserves the reference behavior; if the remaining failure involves a conflict pattern not resolved by the learned correction, the nominal success rate may remain unchanged.

The conflict-stress tests reveal a different picture. In constructed head-on and crossing scenarios, pure reference tracking frequently produces inter-UAV conflicts because independently generated reference paths can bring multiple UAVs into the same spatial region at similar times. Under these conditions, residual TD3 significantly reduces inter-UAV collision episodes and improves hard-safety satisfaction. The improvement is especially clear in crossing conflicts, where the residual policy eliminates inter-UAV collision episodes while remaining close to the reference behavior.

The head-on scenario also exposes the current boundary of the method. Residual TD3 nearly eliminates direct inter-UAV collision in head-on encounters, but some failures are transferred to building collision. This suggests that local inter-UAV avoidance and obstacle avoidance are coupled in dense urban scenes. A residual correction that increases separation from another UAV may simultaneously reduce clearance from buildings. Therefore, future residual policies should include obstacle-aware projection, explicit safety shielding, or local replanning constraints so that inter-UAV conflict resolution does not create new environmental collision risks.

6.3. Safety Teacher, Runtime Scope, and Practical Interpretation

The priority-based safety teacher provides structured right-of-way information in symmetric conflict situations. In head-on and crossing cases, two UAVs may face similar local observations and have no natural convention for yielding. The teacher introduces a consistent conflict-resolution pattern and supplies informative residual actions during training. The experimental comparison shows that residual TD3 reaches performance close to the safety teacher in the stress-test set, indicating that the learned policy approximates the teacher-induced correction behavior under the constructed conflict patterns.

This result should be interpreted as teacher-guided residual policy learning rather than as evidence that the learned policy has discovered a universally superior conflict-resolution rule. The teacher remains an important baseline because it represents the rule-based prior from which the residual policy learns. The value of the learned policy lies in embedding this correction behavior into a continuous observation-action mapping that can be fused with the reference tracker and evaluated under the same execution dynamics.

The runtime structure of the framework is also important. Static path generation, pure reference replay, behavior-cloning initialization, residual TD3 training, and checkpoint selection are performed offline. During execution, the system only computes the local reference action, extracts nearby-UAV interaction features, infers the residual correction, and applies the fused action. Therefore, the online computation is much lighter than global replanning or online reinforcement learning. At the same time, the current framework should not be viewed as a complete onboard replanning system. It does not update the global reference path in response to newly appearing obstacles, large disturbances, or unexpected no-fly regions.

From a practical perspective, the proposed method is most suitable for missions where a known urban map is available, static reference routes can be generated before execution, and online adaptation is mainly required for bounded local interaction. For more dynamic low-altitude operations, the framework would need to be integrated with runtime local replanning, model predictive control, or certified safety filters.

6.4. Limitations and Future Work

Several limitations remain. First, the current static path-generation stage is performed independently for each UAV. This design provides interpretable building-avoiding reference paths, but it does not guarantee inter-UAV temporal separation before execution. The residual module partially addresses this issue under constructed conflict stress, but a future extension could combine reference tracking with time-parameterized multi-UAV path scheduling or reservation-based planning.

Second, the residual TD3 module is trained and evaluated in simulation. Although the environment is derived from real urban map data, the UAV model is still a navigation-level point-mass abstraction. Full aircraft dynamics, attitude control, wind disturbance, communication delay, sensing uncertainty, and actuator limitations are not included. These factors may affect the feasibility of residual corrections in real flight. Hardware-in-the-loop simulation and small-scale flight experiments are needed to validate the method under more realistic dynamics.

Third, the current residual policy uses successful demonstrations for behavior-cloning initialization, while failed episodes are not fully exploited. Failed trajectories may contain useful information about unsafe states, recovery failures, and collision precursors. Incorporating these data into a safety critic, risk classifier, or constrained reinforcement learning formulation may improve the policy's ability to avoid rare but severe failure cases.

Fourth, the stress-test results show that reducing inter-UAV collisions can sometimes increase building collision risk. This indicates that bounded residual correction alone is

not sufficient in highly constrained corridors. Future work should introduce obstacle-aware residual projection or a runtime safety shield that verifies whether the fused action preserves both inter-UAV separation and building clearance before execution.

Finally, the present evaluation focuses on a fixed urban scene and a five-UAV setting. Larger fleets, more diverse urban maps, dynamic obstacles, uncertain localization, and heterogeneous UAV performance envelopes should be tested to evaluate scalability and generalization. These extensions would also help determine whether residual learning remains effective when the number of pairwise interactions grows and when conflict patterns become less regular.

Overall, the results support a clear interpretation of the framework. Static paths provide geometric reachability, reference tracking provides executable baseline behavior, and residual TD3 provides conflict-aware local correction under selected stress conditions. The current method improves the connection between planning and execution, but it does not yet constitute a complete solution to all urban multi-UAV conflict-resolution problems.

7. Conclusions

This paper presented a hierarchical navigation framework for urban multi-UAV navigation that connects static reference path generation, executable reference tracking, demonstration-guided initialization, and conflict-aware residual reinforcement learning. The motivation was to address the limitation that a collision-free geometric path is not necessarily an executable UAV trajectory under velocity limits, acceleration constraints, local path-association errors, and simultaneous multi-UAV interactions. Instead of using end-to-end reinforcement learning to learn the entire navigation pipeline from scratch, the proposed method uses geometric planning and damped reference tracking to construct an interpretable executable baseline and then applies bounded residual TD3 only as a local conflict-correction mechanism.

The experimental results show that executable reference tracking is a necessary component of the proposed pipeline. In the reference-tracking ablation test, simplified tracking strategies failed to execute the static paths reliably, with a 0.00% all-success rate and 100.00% building collision rate. In contrast, the full-damped reference-tracking controller achieved a 91.67% all-success rate and a 96.67% individual-UAV success rate, while eliminating building collision episodes and reducing the mean final three-dimensional error to 63.50 m. Speed-sensitivity tests at 10, 15, and 20 m/s further showed the same 91.67% all-success rate for pure reference tracking, indicating that the tracking conclusion was not caused by a single speed setting.

After the executable reference baseline was established, the conflict-aware residual TD3 module was evaluated under nominal and constructed conflict-stress scenarios. In the nominal test set, residual TD3 preserved the pure-reference performance but did not improve the all-success rate, which confirms that the residual policy should not be interpreted as a universal replacement for reference tracking. Its value became evident under conflict-stress conditions. In the overall stress test, BC + residual TD3 increased the all-success rate from 33.33% to 80.09%, reduced the inter-UAV collision rate from 66.67% to 11.57%, and improved the hard-safety satisfaction rate from 33.33% to 87.04%. In the representative crossing case, the minimum separation between the conflicting UAV pair increased from 3.31 m under pure reference tracking to 26.08 m under BC + residual TD3, satisfying the soft-safety threshold.

These results support the central interpretation of the proposed framework: static reference paths provide global geometric reachability, the damped reference tracker converts these paths into executable closed-loop reference trajectories, and residual TD3 provides bounded local correction under selected inter-UAV conflict conditions. The proposed

framework therefore improves both interpretability and execution stability compared with a purely end-to-end learning formulation.

Several limitations remain. First, static paths are generated independently for each UAV and do not guarantee temporal separation before execution. Second, the current evaluation is based on a navigation-level point-mass model in simulation, without full attitude dynamics, wind disturbance, communication delay, sensing uncertainty, or actuator limitations. Third, residual TD3 reduces inter-UAV collisions in conflict-stress tests but may transfer some failures to building collisions in constrained corridors. Future work will integrate obstacle-aware residual projection, runtime safety shielding, local replanning, and more realistic UAV dynamics. Larger fleets, more diverse urban maps, dynamic obstacles, and hardware-in-the-loop or flight experiments will also be considered to further evaluate scalability and practical applicability.

Author Contributions: Conceptualization, methodology, formal analysis, writing, X.Z., Q.X. and S.L. (Shanmei Li); software, data, resources, X.Z. and S.L. (Siqin Li). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ordos Institute of Technology 2024 scientific research project (Grant no. ZRYB2024014), the National Science Foundation of Tianjin (Grant no. 24JCY-BJC01170), the Graduate Research Innovation Project of Civil Aviation University of China (Grant 2025YJSKC03001), the Ordos Higher Education Institutions Scientific Research Innovation Project (Grant nos. KYQN25Z001, KYLJ25Z001, KYLJ25Z002), the Scientific Research Projects of Colleges and Universities in Inner Mongolia (Grant No. NJZY21158), and the Project of Ordos City Key Innovation Team (R&D Team for Drum-type Small and Micro Gas Turbines).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, X.; Li, L.; Mo, Y.; Huang, J.; Qin, S.J. A distributed route network planning method with congestion pricing for drone delivery services in cities. *Transp. Res. Part C Emerg. Technol.* **2024**, *160*, 104536. [\[CrossRef\]](#)
2. Ait Saadi, A.; Soukane, A.; Meraihi, Y.; Benmessaoud Gabis, A.; Mirjalili, S.; Ramdane-Cherif, A. UAV Path Planning Using Optimization Approaches: A Survey. *Arch. Comput. Methods Eng.* **2022**, *29*, 4233–4284. [\[CrossRef\]](#)
3. Zhang, J.; Chen, L.; Gao, Z.; Duo, Y. Low-Altitude Unmanned Aerial Vehicle Technology: Current Status and Prospects. *Strateg. Study CAE* **2025**, *27*, 73–85. [\[CrossRef\]](#)
4. Hart, P.; Nilsson, N.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [\[CrossRef\]](#)
5. Koenig, S.; Likhachev, M.; Furcy, D. Lifelong Planning A*. *Artif. Intell.* **2004**, *155*, 93–146. [\[CrossRef\]](#)
6. Jeong, I.-B.; Lee, S.-J.; Kim, J.-H. Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate. *Expert Syst. Appl.* **2019**, *123*, 82–90. [\[CrossRef\]](#)
7. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [\[CrossRef\]](#)
8. Sanchez-Lopez, J.L.; Olivares-Mendez, M.A.; Castillo-Lopez, M.; Voos, H. Towards trajectory planning from a given path for multirotor aerial robots trajectory tracking. In *Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1342–1351.
9. Ding, W.; Gao, W.; Wang, K.; Shen, S. An Efficient B-Spline-Based Kinodynamic Replanning Framework for Quadrotors. *IEEE Trans. Robot.* **2019**, *35*, 1287–1306. [\[CrossRef\]](#)
10. Liu, S.; Mohta, K.; Atanasov, N.; Kumar, V. Search-Based Motion Planning for Aggressive Flight in SE(3). *IEEE Robot. Autom. Lett.* **2018**, *3*, 2439–2446. [\[CrossRef\]](#)
11. Tal, E.; Ryou, G.; Karaman, S. Aerobatic Trajectory Generation for a VTOL Fixed-Wing Aircraft Using Differential Flatness. *IEEE Trans. Robot.* **2023**, *39*, 4805–4819. [\[CrossRef\]](#)
12. Xu, S.; Zhang, X.; Li, C.; Wang, D.; Yang, L. Deep Reinforcement Learning Approach for Joint Trajectory Design in Multi-UAV IoT Networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 3389–3394. [\[CrossRef\]](#)
13. Han, J.; Zhu, Y.; Yang, J. A Deep Reinforcement Learning Method for Collision Avoidance with Dense Speed-Constrained Multi-UAV. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2152–2159. [\[CrossRef\]](#)

14. Xing, X.; Zhou, Z.; Li, Y.; Xiao, B.; Xun, Y. Multi-UAV Adaptive Cooperative Formation Trajectory Planning Based on an Improved MATD3 Algorithm of Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12484–12499. [\[CrossRef\]](#)
15. Sharon, G.; Stern, R.; Felner, A.; Sturtevant, N.R. Conflict-based search for optimal multi-agent pathfinding. *Artif. Intell.* **2015**, *219*, 40–66. [\[CrossRef\]](#)
16. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
17. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*; PMLR: London, UK, 2018; pp. 1587–1596.
18. Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*; PMLR: London, UK, 2016; pp. 1329–1338.
19. Liu, C.; Zhong, Y.; Wu, R.; Ren, S.; Du, S.; Guo, B. Deep Reinforcement Learning Based 3D-Trajectory Design and Task Offloading in UAV-Enabled MEC System. *IEEE Trans. Veh. Technol.* **2025**, *74*, 3185–3195. [\[CrossRef\]](#)
20. Zhao, G.; Wang, J.; Meng, Z.; Wang, Z.; Fu, H.; Jiang, C. Energy-Efficient Path Planning and Task Allocation for Multi-Drone-Aided IoT Cluster-Based Data Collection. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 14177–14191. [\[CrossRef\]](#)
21. Chen, Y.; Dong, Q.; Shang, X.; Wu, Z.; Wang, J. Multi-UAV Autonomous Path Planning in Reconnaissance Missions Considering Incomplete Information: A Reinforcement Learning Method. *Drones* **2022**, *7*, 10. [\[CrossRef\]](#)
22. Shao, M.; Liu, X.; Xiao, C.; Zhang, T.; Yuan, H. Research on UAV Trajectory Planning Algorithm Based on Adaptive Potential Field. *Drones* **2025**, *9*, 79. [\[CrossRef\]](#)
23. Hu, Z.-H.; Huang, Y.-L.; Li, Y.-N.; Bao, X.-Q. Drone-Based Instant Delivery Hub-and-Spoke Network Optimization. *Drones* **2024**, *8*, 247. [\[CrossRef\]](#)
24. Wen, S.; Shu, Y.; Rad, A.; Wen, Z.; Guo, Z.; Gong, S. A deep residual reinforcement learning algorithm based on Soft Actor-Critic for autonomous navigation. *Expert Syst. Appl.* **2025**, *259*, 125238. [\[CrossRef\]](#)
25. Zhang, Z.; Wang, Y.; Zhang, Z.; Wang, L.; Huang, H.; Cao, Q. A residual reinforcement learning method for robotic assembly using visual and force information. *J. Manuf. Syst.* **2024**, *72*, 245–262. [\[CrossRef\]](#)
26. Sheng, Z.; Huang, Z.; Chen, S. Traffic expertise meets residual RL: Knowledge-informed model-based residual reinforcement learning for CAV trajectory control. *Commun. Transp. Res.* **2024**, *4*, 100142. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.