UAST: Unified Active Search and Tracking for Arbitrary Targets with UAVs

Liang Qin[1,2], Min Wang[3*], Xingyu Lu[1,2], Aowen Qiu[1,2], Wengang Zhou[1,2*], and Houqiang Li[1,2]

[1]MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, China
[2]University of Science and Technology of China, China
[3]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, China
{qinliang6218,luxingyu010,qaw}@mail.ustc.edu.cn; wangmin@iai.ustc.edu.cn; {zhwg,lihq}@ustc.edu.cn);

Abstract

Active search and tracking of arbitrary targets by Unmanned Aerial Vehicles (UAVs) in cluttered environments remains a highly challenging problem. Existing methods either construct complex modular pipelines, leading to substantial computational costs, or adopt end-to-end controllers that often fail to generalize across different targets and scenes. Moreover, search and tracking are typically treated separately despite their strong interdependence. In this paper, we present UAST, a simple yet effective mapping-free framework that unifies active search and persistent tracking using only RGB-D observations. The proposed system couples a dual-branch perception module with a Rule-Based Point Search Policy that adaptively switches between tracking and search-based recovery. A lightweight control network generates dynamically feasible trajectories directly from fused perception and UAV states. Furthermore, we introduce a training strategy with an elaborated tracking-aware visibility loss and a tailored data construction. Extensive experiments in both simulated and real-world environments show that our approach achieves higher success rates, more stable long-term tracking, and faster target search compared with existing methods, while maintaining high efficiency. The code is available at https://github.com/qinliangql/UAST.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are indispensable tools for tasks such as inspection, surveillance, and search-and-rescue. A long-standing challenge in this domain is enabling a UAV to actively search for and persistently track arbitrary targets in complex, cluttered environments solely with onboard sensing. Such a capability requires tightly coupling perception, decision-making, and control under real-time constraints, where issues such as long-range motion, rapid dynamics, and frequent occlusions make continuous target awareness particularly difficult.

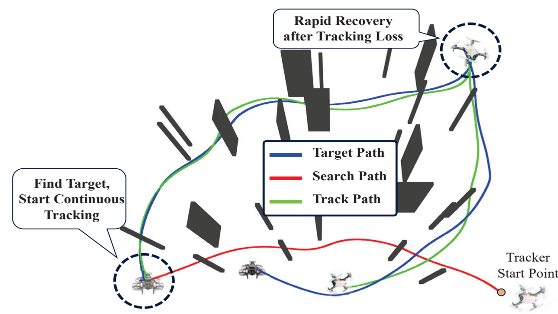


Figure 1. Demonstration of active search and tracking using the proposed **UAST**. The UAV autonomously searches the environment (Search Path) to locate the target and seamlessly transitions into continuous tracking (Tracking Path) once the target is found. Short interruptions in tracking occur due to rapid target motion, but UAST rapidly recovers, demonstrating robust, unified active search and tracking in a fully mapping-free manner.

Most prior works address the problem by splitting it into several subproblems: aerial object detection [20, 21, 26, 29, 31–35, 39], navigation [10, 23, 27, 42, 44, 47], scene exploration and search [7–9, 38, 40, 41, 43, 46], and UAV-based target tracking [5, 11, 13–15, 18, 25, 30]. While each of these areas achieves notable progress, the core of this task lies in tracking with inherent search capability, which remains particularly challenging under long-range motion, high-speed, and frequent occlusions.

Existing tracking solutions follow two predominant trends. Classical model-based pipelines integrate perception, mapping, front-end path search, and back-end trajectory optimization. While these methods achieve stable performance in structured settings, maintaining voxel maps introduces memory overhead and contributes to high computational costs, limiting their efficiency in high-speed or resource-constrained scenarios [11, 15, 30]. Learning-based, end-to-end controllers simplify the software stack and enable agile flight, yet they often show limited gen-

*Corresponding author.

eralization across unseen targets and scenes, and lack reliable recovery mechanisms after tracking loss [23–25, 42]. Moreover, most existing works treat search and tracking as independent tasks, overlooking their intrinsic interdependence—whereas sustained long-term performance demands both robust in-frame tracking and active re-acquisition when the target becomes occluded or lost.

To address these limitations, we propose UAST, a unified, mapping-free framework that jointly performs active search and persistent tracking of arbitrary targets using only RGB-D inputs. As illustrated in Fig. 1, UAST enables the UAV to autonomously explore, re-acquire, and continuously follow the target through a single, integrated perception–control pipeline. The framework extracts target and environmental representations through dual RGB–depth branches, while a rule-based point-search policy generates a guidance point based on target visibility. This guidance point is jointly encoded with the UAV's velocity and acceleration, fused with perception features, and then processed by a lightweight Control Network to predict feasible trajectories directly. Without explicit mapping or multi-stage planning, this design enables adaptive transitions between exploration and tracking refinement, ensuring smooth recovery from target occlusion or loss. To further enhance robustness and generalization, UAST employs a tracking-aware visibility loss and a diverse data generation strategy that jointly promote geometry-aware and stable control. Through these designs, UAST achieves real-time, geometry-consistent, and resilient UAV tracking across cluttered environments.

Extensive simulation and real-world experiments are conducted, and UAST consistently outperforms prior methods across both tracking and search tasks. In long-range, high-speed scenarios, our approach improves the tracking success rate by over 50% compared to state-of-the-art baselines, while achieving significantly faster recovery and higher search efficiency. Besides, we deploy this method on a quadrotor and validate it in real-world scenarios, confirming its practical feasibility.

2. Related Work

2.1. Aerial Detection

Aerial object detection [16, 17, 19] is a fundamental perception task for UAVs. Traditional detectors like Faster R-CNN [12] and SSD [22] achieve high accuracy but are computationally heavy for onboard deployment. Recent work focuses on lightweight, real-time solutions. Li et al. [20] propose AutoTrack, improving UAV detection and tracking via adaptive spatio-temporal optimization. Cao et al. [3, 4] introduce hierarchical and temporal transformers for robustness under motion and occlusion. Wu et al. [34] design an occlusion-robust vision transformer for real-time UAV inference. Anchor-free and one-stage methods, such as FCOS [28] and SiamFC++ [37], further enhance speed and simplicity while maintaining accurate detection and tracking.

2.2. Exploration and Search

Autonomous exploration [2, 6, 41] enables UAVs to map and search unknown environments efficiently under sensing and computation limits. Early methods like NBVP [1] and FUEL [45] use frontier-based strategies but suffer from revisits and limited coverage. RACER [46] and ECHO [40] improve consistency and motion smoothness via global guidance or heuristic frontier evaluation. FALCON [41] leverages hierarchical decomposition and connectivity-aware planning for efficient coverage. Despite these advances, most planners rely on explicit mapping and staged optimization, incurring computational cost and reducing adaptability. Our approach differs by coupling perception and control within a unified, mapping-free framework for active search and tracking.

2.3. Active Tracking

Active tracking requires a UAV to maintain continuous visibility of a moving target while actively adjusting its motion to reorient and pursue under dynamic conditions. Traditional methods decompose this process into perception, mapping, and planning modules, maintaining visibility through occlusion-aware trajectory planning and kinodynamic optimization [11, 15, 30]. Wang et al. [30] plan visibility-constrained trajectories to keep the target within the camera's field of view, while Ji et al. [15] introduce an elastic tracking corridor for smooth and safe motion in cluttered scenes. Gao et al. [11] enhance robustness via adaptive trajectory planning that combines motion prediction with control optimization. Although effective in structured settings, these modular and optimization-heavy frameworks rely on explicit mapping and iterative computation, limiting real-time agility and recovery from target loss.

Learning-based approaches couple perception and control within end-to-end frameworks to improve reactivity. Lu et al. propose Yopov2 [25], which directly predicts control commands from visual inputs, enabling agile, high-speed tracking without explicit mapping. However, purely end-to-end systems often struggle to balance efficiency, interpretability, and generalization across diverse targets and environments. Our method follows the end-to-end paradigm but incorporates structural priors through explicit target perception, geometric reasoning, and a rule-based search strategy, enabling unified search and tracking with stable recovery and robust long-term performance.

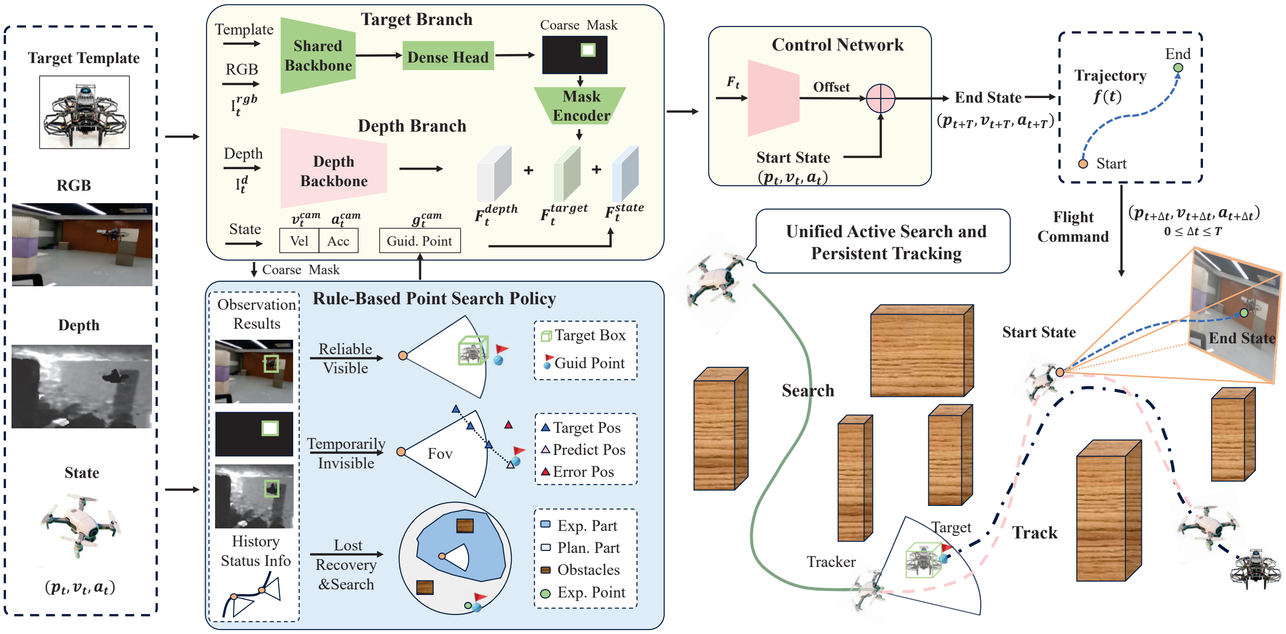


Figure 2. Overview of our **UAST** framework. The system takes RGB-D observations, UAV states, and a target template as inputs. The *Target Branch* extracts target appearance features, while the *Depth Branch* encodes geometric and navigability cues. The *Rule-Based Point Search Policy* adaptively selects guidance points for tracking and active search, correcting estimates when occluded or switching to exploration upon tracking failure. The *Control Network* predicts end-state offsets and generates feasible trajectories. The right illustration shows the UAV's transition from search to tracking, demonstrating unified, mapping-free operation.

3. Method

3.1. Problem Formulation

We address active search and persistent tracking of arbitrary targets using a UAV equipped with an RGB-D camera in cluttered 3D environments. The goal is to autonomously locate, pursue, and maintain visibility of a user-specified target without explicit mapping or prior environmental knowledge. At each time step t, the UAV receives an RGB-D observation $\mathcal{I}t = \{I_t^{rgb}, I_t^d\}$, its kinematic state $s_t = v_t, a_t$, and a target template M^{target}. The tracker's current state is defined as $S_t^{start} = \{p_t, v_t, a_t\}$. Our method predicts a short-horizon end-state $S_{t+T}^{end} = \{p_{t+T}, v_{t+T}, a_{t+T}\}$, which determines a dynamically feasible trajectory. The immediate control command at step t is derived from this trajectory, guiding the UAV toward the target while ensuring visibility and safety.

3.2. Overall Framework

UAST performs unified search and persistent tracking using only RGB-D inputs, without global mapping or multi-stage planning. As shown in Fig. 2, the framework consists of four components: Target Branch, Depth Branch, Rule-Based Point Search Policy, and Control Network.

The Target Branch extracts appearance features from the RGB observation and a target template. A dense correlation head localizes the target and generates a coarse mask, which is encoded into the final target representation. The Depth Branch encodes geometric and navigability information from the depth image, capturing obstacle layout and local free space in the UAV frame. These perception features, together with historical state information, are fused in the Rule-Based Point Search Policy for unified active perception. When the target is visible, the policy updates the guidance point based on position consistency and prediction error. If the target becomes occluded or deviates from its predicted trajectory, past observations are used for correction. During tracking failure, the policy switches to a coverage-based exploration strategy, generating feasible exploration points to quickly regain visual contact, enabling seamless transition between search and tracking.

Finally, the Control Network produces dynamically feasible motion commands. The guidance point g_t^{cam}, along with the UAV velocity v_t^{cam} and acceleration a_t^{cam}, forms the state representation F_t^{state}. This state is fused with perception features to generate the input F_t, from which the network predicts an end-state offset S_{t+T}^{off} and constructs a smooth trajectory that ensures stable flight and target visibility. The overall framework achieves mapping-free unified active search and tracking with robust and efficient performance.

Theoretical Perspective. UAST formulates search and tracking as a unified trajectory optimization problem. The optimal trajectory $\mathbf{f}(t)$, representing the UAV's planned motion, is derived by minimizing a cost function conditioned on the guidance point $\mathbf{g}_t(m_t)$ from the Rule-Based Point Search Policy:

$$\mathbf{f}^*(t) = \arg\min_{\mathbf{f}(t)} \mathcal{L}_{traj}(\mathbf{f}(t); \mathbf{g}_t(m_t)). \tag{1}$$

where $\mathcal{L}_{traj}$ encodes trajectory quality. High confidence in the belief state m_t steers $\mathbf{g}_t$ toward tracking, while uncertainty directs it to exploration. This yields a continuous behavioral spectrum from a single optimization framework.

3.3. Target and Depth Branches

The perception module provides complementary target-conditioned and environment-aware cues for unified search and tracking. It consists of a Target Branch for template-based localization and a Depth Branch for geometric scene understanding.

Target Branch. The RGB frame I_t^{rgb} and target template I^{target} are processed by a shared-weight AlexNet backbone to produce embeddings F_t^{rgb} and F^{temp}. The template feature is extracted once at initialization and reused throughout tracking. Cross-correlation between F_t^{rgb} and F^{temp} generates a response map, from which a lightweight head predicts a bounding box and coarse mask. The mask is encoded into a compact representation F_t^{target}, providing target-conditioned localization cues without relying on full appearance features, enabling class-agnostic generalization.

Depth Branch. The depth observation I_t^d is processed by a ResNet-18 backbone to produce a compact feature tensor F_t^{depth}. The feature encodes structural information, including surface geometry, depth discontinuities, and coarse free-space layout. This representation provides environment-aware cues that complement the target-conditioned feature from the Target Branch.

The outputs $\{F_t^{target}, F_t^{depth}\}$ are fused with the UAV's kinematic state and the guidance point from the Rule-Based Search Policy, and then input to the Control Network. The resulting unified representation supports obstacle-aware and target-directed motion prediction.

3.4. Rule-Based Point Search Policy

The Rule-Based Point Search Policy generates coarse guidance points for the Control Network using aggregated perception and state cues, without explicit mapping or path planning. It handles three scenarios: (1) target reliably visible, (2) target temporarily invisible, and (3) target lost for an extended period, triggering exploration-based recovery.

Target reliably visible: Guidance point estimation. The target is considered reliably detected if the response score m_t of the predicted bounding box exceeds a threshold τ_{det}. In this case, a 3D guidance point is computed from

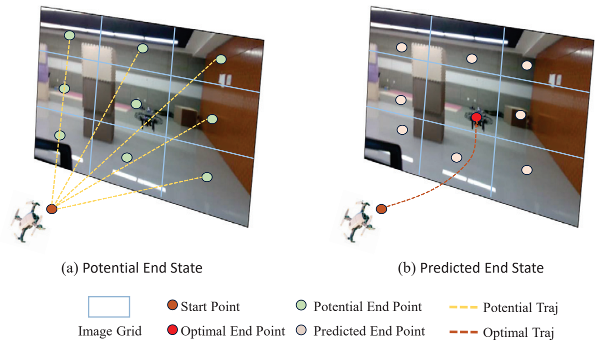


Figure 3. Illustration of the Control Network's prediction. The depth and target features are spatially encoded into a $V \times H$ grid, where each cell corresponds to an initial end state defined by pos p_t, vel v_t, and acc a_t. The network predicts an offset and a trajectory score for each cell, from which the optimal end state is selected based on the score. With the start and end states, a 5th-order polynomial trajectory is solved along the x, y, and z axes, yielding a smooth 3D motion curve that determines the UAV's control commands.

the bounding box and depth. Given the 2D bounding box $b_t = [u_{\min}, v_{\min}, u_{\max}, v_{\max}]$, the central region is

$$b_t^c = \Big[u_c - \tfrac{1}{4}w,\ v_c - \tfrac{1}{4}h,\ u_c + \tfrac{1}{4}w,\ v_c + \tfrac{1}{4}h\Big], \tag{2}$$

where (u_c, v_c) is the center of the box, and w, h are its width and height. Valid depth values within b_t^c are filtered to compute the median $\bar{z}$, and the corresponding 3D point in the camera frame is

$$g_t^{cam} = \big[(u_c - c_x)\bar{z}/f_x,\ (v_c - c_y)\bar{z}/f_y,\ \bar{z}\big]^\top, \tag{3}$$

where f_x and f_y are the focal lengths, and (c_x, c_y) is the principal point of the camera. This point g_t^{cam} is transformed into the world frame g_t^{world} and serves as the current guidance point.

Target temporarily invisible: Short-term prediction. To smooth guidance point estimates under noisy depth or partial occlusion, a Kalman filter (KF) is applied. The KF state vector is defined as $\mathbf{x}_t = \big[g_t^{world}, \dot{g}_t^{world}\big]^T$, containing the guidance point in world coordinates and its velocity. The state evolves according to a constant-velocity model:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{w}_t, \quad A = \begin{bmatrix} I_3 & \Delta t I_3 \\ 0 & I_3 \end{bmatrix}, \tag{4}$$

where I_3 denotes the 3×3 identity matrix, and $\mathbf{w}_t$ denotes process noise. The measurement vector is

$$\mathbf{y}_t = H\mathbf{x}_t + \alpha_t, \quad H = [I_3\ \ 0], \tag{5}$$

where $H \in \mathbb{R}^{3\times 6}$ maps the state to the measurement space, and $\alpha_t \sim \mathcal{N}(0, R)$ is the measurement noise with covariance R. The innovation (measurement residual) is computed as $\eta_t = \|\mathbf{y}_t - H\hat{\mathbf{x}}_{t|t-1}\|_{R^{-1}}$ where $\hat{\mathbf{x}}_{t|t-1}$ is the predicted state prior. If η_t exceeds a threshold τ_{innov}, the current detection is deemed unreliable, and the predicted guidance point $\hat{g}_t^{world} = H\hat{\mathbf{x}}_{t|t-1}$ is used instead.

Target lost: Exploration-based recovery and search. When the target remains invisible for T_{lost} consecutive frames or at the beginning of a search task when no target has been detected, the system activates an exploration mode. The UAV initializes an unexplored circular region centered at its current position p_t with radius r_{max}. The area within the current camera field of view, including all visible regions up to the maximum sensing range or until occluded by obstacles, is marked as explored.

The policy then randomly samples candidate points $\{c_k\}$ within the unexplored region and selects the farthest valid point as the next guidance point: $g_t^{world} = \arg\max_{c_k} \|c_k - p_t\|$. If no unexplored points exist within the condition, a spiral exploration strategy is adopted to generate a new point:

$$r(\theta) = r_0 + \epsilon\theta, \quad p(\theta) = p_t + [r(\theta)\cos\theta,\ r(\theta)\sin\theta], \tag{6}$$

where r_0 defines the initial search radius and ϵ controls the spiral expansion rate. When the UAV reaches a guidance point, the same rule is applied to update the next point.

If the exploration region becomes saturated—when most areas within the initial range are explored or the target is in an unreachable zone—the search radius is adaptively expanded by a factor (e.g., 1.5). This gradually extends the search frontier, allowing the UAV to quickly re-localize a lost target or efficiently explore for a new target.

3.5. Control Network

The Control Network translates fused perception features and UAV state information into executable motion commands. Its goal is to predict a dynamically feasible end state within a short horizon and generate a smooth 3D trajectory from start to end, as illustrated in Fig. 3.

Inputs include fused Depth and Target Branch features, the UAV state $\{v_t, a_t\}$ in the camera frame, and guidance point g_t^{cam}. The network predicts offsets in the camera frame, so the current position p_t is not required. The processed perception tensor is represented as a $V \times H$ grid, where each element corresponds to a local receptive field. This spatial structure is preserved throughout the network while the channel dimension is modified to extract higher-level features. Each grid element has a default *initial end state* $S_{v,h}^{init} = \{p_{v,h}, v_{v,h}, a_{v,h}\}$, defined by its geometric position, orientation, and a default prediction radius along the trajectory. Directional adjustments for velocity and acceleration provide a reasonable prior.

A lightweight CNN predicts an offset for each cell $\Delta S_{v,h}^{off} = \{\Delta p, \Delta v, \Delta a\}$ and a scalar trajectory cost score $s_{v,h}$, which estimates the predicted trajectory's loss. The predicted end state of each cell is computed as: $S_{v,h}^{pred} = S_{v,h}^{init} + \Delta S_{v,h}^{off}$. The cell with the lowest score is selected as the optimal end state: $S_{t+T}^{end} = \arg\min_{v,h}(s_{v,h})$, defining the UAV's predicted position, velocity, and acceleration after horizon T. Given $S_t^{start} = \{p_t, v_t, a_t\}$ and S_{t+T}^{end}, a 5th-order polynomial is fitted for each axis:

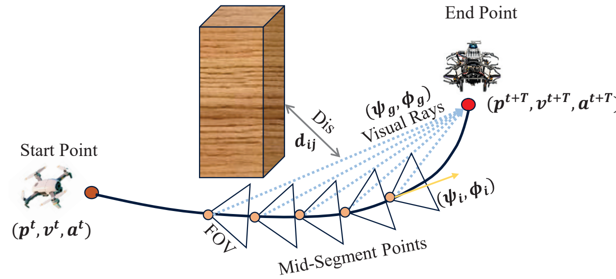


Figure 4. Illustration of the proposed tracking-aware visibility loss. The UAV trajectory (black curve) is evaluated at mid-segment points for visibility consistency. Line-of-sight occlusion (visual rays) is assessed through differentiable SDF-based sampling to estimate obstacle distance, while vertical and horizontal FOV deviation is measured by the angle difference between the UAV's velocity direction and the target direction.

$$f_x(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0, \tag{7}$$

with $f_y(t)$ and $f_z(t)$ similarly defined. The 5-order polynomial satisfies six constraints per axis (position, velocity, and acceleration at both endpoints) while ensuring a higher-order smooth, continuous 3D trajectory $f(t) = \{f_x(t), f_y(t), f_z(t)\}$. Control commands are derived from the trajectory as:

$$u_t = \{p(t), v(t), a(t)\} = \{f(t), \dot{f}(t), \ddot{f}(t)\}, \tag{8}$$

providing references for low-level flight control.

3.6. Training Objectives and Data Construction

Our framework follows a modular yet fully differentiable training design. The Target Branch, responsible for RGB-template feature extraction and bounding-box prediction, is trained independently following the SiamFC++ [37] protocol, and its outputs are fixed for subsequent modules. The Depth Branch, Mask Encoder in Target Branch, and Control Network are trained jointly end-to-end, as the predicted end-state offsets determine a continuous 3D trajectory for computing all physical and perceptual losses.

The total objective consists of two terms: a predicted trajectory loss and a score regression loss:

$$\mathcal{L} = \lambda_t\mathcal{L}_{traj} + \lambda_s\mathcal{L}_{score}. \tag{9}$$

The trajectory loss enforces perceptual consistency, goal orientation, path smoothness, and flight safety:

$$\mathcal{L}_{traj} = \lambda_1\mathcal{L}_{track} + \lambda_2\mathcal{L}_{goal} + \lambda_3\mathcal{L}_{smooth} + \lambda_4\mathcal{L}_{safe}. \tag{10}$$

Tracking-aware Visibility Loss. To maintain target visibility and stable camera alignment, we introduce a *tracking-aware visibility loss* $\mathcal{L}_{track}$ that evaluates the geometric quality of the predicted trajectory (Fig. 4). Only the mid-segment (one-third to two-thirds of the horizon) is used, reflecting short-term UAV behavior.

The loss integrates two differentiable terms:

$$\mathcal{L}_{track} = \lambda_a \mathcal{L}_{occ} + \lambda_f \mathcal{L}_{fov}. \tag{11}$$

The occlusion term $\mathcal{L}_{occ}$ enforces visibility along the UAV–target line of sight through differentiable Signed Distance Function (SDF)-based sampling. For each mid-segment point $\mathbf{p}_i$, N points are sampled along the ray toward the target $\mathbf{p}_g$, and the signed distance d_{ij} are accumulated as:

$$\mathcal{L}_{occ} = \sum_i \sum_j \mathrm{ReLU}(d_{\text{voxel}} - d_{ij}), \tag{12}$$

where d_{voxel} denotes the voxel size threshold. If $d_{ij} < d_{\text{voxel}}$, it implies that the line of sight is occluded by an obstacle. The field-of-view term $\mathcal{L}_{fov}$ penalizes deviations beyond the camera's vertical and horizontal FOV:

$$\mathcal{L}_{fov} = \sum_i \mathrm{ReLU}\Big(|\psi_i - \psi_g| - \psi_{\text{fov}}\Big)\mathrm{ReLU}\Big(|\phi_i - \phi_g| - \phi_{\text{fov}}\Big), \tag{13}$$

where (ψ_i, ϕ_i) and (ψ_g, ϕ_g) denote the yaw and pitch angles of the UAV's velocity vector and the target direction, respectively. These terms encourage trajectories with continuous visibility and stable in-frame orientation.

Goal Alignment, Smoothness, Safety. Goal alignment encourages the trajectory to follow the target or searching goal while minimizing lateral deviation:

$$\mathcal{L}_{goal} = (1-\omega_p)\left|\|\mathbf{o_g}\| - (\mathbf{o}_t \cdot \hat{\mathbf{o}}_g)\right| + \omega_p \left\|\mathbf{o}_t - (\mathbf{o}_t \cdot \hat{\mathbf{o}}_g)\hat{\mathbf{o}}_g\right\|, \tag{14}$$

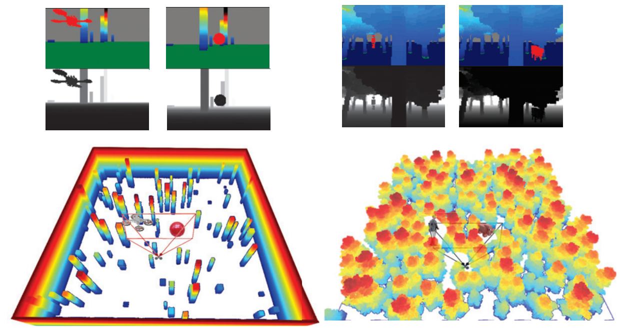
where $\hat{\mathbf{o}}_t$ and $\hat{\mathbf{o}}_g$ are normalized vectors representing the predicted trajectory direction and the goal direction, respectively. The first term is the trajectory length error, and the second is the cosine similarity with the goal. The weighting factor ω_p increases as the UAV approaches the goal.

The smoothness term regularizes transitions in position, velocity, and acceleration between start and predicted end states:

$$\mathcal{L}_{smooth} = \sum_{k \in \{x,y,z\}} \mathbf{s}_k^\top \mathbf{R}_s \mathbf{s}_k, \tag{15}$$

where $\mathbf{s}_k = [p_k^t, v_k^t, a_k^t, p_k^{t+T}, v_k^{t+T}, a_k^{t+T}]^\top$ represents the state vector for axis k, and $\mathbf{R}_s$ is a predefined diagonal matrix that weights the smoothness of position, velocity, and acceleration transitions for that axis. Safety is enforced via the signed distance along the predicted trajectory:

$$\mathcal{L}_{safe} = \int_0^T \exp\Big(-\frac{d(\mathbf{p}(t)) - d_0}{\beta}\Big)\, dt, \tag{16}$$



(a) Hall (b) Forest

Figure 5. Illustration of the proposed data generation pipeline. UAV RGB-D observations are collected without targets in simulated or reconstructed real scenes. Synthetic targets (UAV, human, ball, animal) are projected into the camera view using intrinsic calibration to produce realistic tracking data.

with $d(\mathbf{p}(t))$ the SDF distance to obstacles, d_0 a safety margin, and β a scaling factor, ensuring differentiable collision avoidance.

Score Regression. The Control Network predicts a scalar score $s_{v,h}$ for each grid cell, estimating its trajectory cost. A Smooth L1 loss aligns this prediction with the actual trajectory loss:

$$\mathcal{L}_{score} = \mathrm{SmoothL1}(l_{v,h}, \mathcal{L}_{traj}), \tag{17}$$

allowing the network to learn an interpretable cost.

Data Construction Strategy. Manual UAV tracking datasets often lack diversity in trajectories and target configurations. We therefore design an automated pipeline applicable to both simulated and real-world environments.

First, a static 3D point-cloud scene is generated without targets. Random UAV positions and attitudes (within feasible pitch limits) are sampled in navigable space, recording RGB-D observations. As shown in Fig. 5, a target point is randomly selected in the camera frame, and synthetic objects (quadrotor, sphere, human, etc.) are projected using camera intrinsics, updating color and depth channels. Target distances are distributed as 40% beyond 10 m, 50% within 5–10 m, and 10% within 5 m, ensuring that 90% are visible and the rest introduce mild occlusions. Meter-level guidance point noise is added for robustness.

4. Experiments

All training and simulation experiments are conducted on a workstation with an Intel i9-13980HX CPU and NVIDIA RTX 4090 GPU. Over 100,000 samples are generated from diverse targets and scenes. The batch size is 1024, the network is trained for 50 epochs, taking approximately 8 hours.

UAV dynamics are simulated in two scene types: *(1) Open environments* (*hall* and *forest*) with sparse obstacles (Fig. 5); *(2) Cluttered environments* with dense pillars and

Method	3m/s	4m/s	5m/s	6m/s
Vis. [30]	0.85	0.60	0.15	0.15
Elas. [15]	0.90	0.60	0.10	0.00
Yopov2. [25]	**1.00**	0.90	0.90	0.80
Yopov2.(multi)	0.90	0.75	0.63	0.60
Ours	**1.00**	**1.00**	**0.93**	**0.88**

Table 1. Short-range flight tracking rate on different target speeds. Yopov2.(multi) is trained on multiple target versions.

Method	Vis. [30]	Elas. [15]	Yopov2. [25]	Ours
3m/s	0.44	0.51	0.67	**0.99**
5m/s	0.05	0.03	0.36	**0.89**
Cycle Time (ms)	60.5	26.4	**7.0**	8.6

Table 2. Long-term tracking success ratio and cycle time. Cycle time denotes the average per-step processing latency.

irregular low walls imposing occlusion challenges. Only the target object executes different-speed maneuvers, while other objects act as distractors. We evaluate performance using short- and long-term tracking success rate (SR), search efficiency, flight safety, and a FoV alignment measure quantifying target stability within the camera's view.

4.1. Short-Range Tracking Performance

We first test in open environments over 100-150 m flights, using 40 randomized trajectories with varied start and end locations. The target UAV follows point-to-point motion via YOPO [24], with distractors added to increase perceptual complexity. The short-range success rate is the ratio of trajectories in which the target remains visible and the UAV maintains at least 0.2 m obstacle clearance.

Table 1 shows that mapping-based trackers (Vis. [30], Elas. [15]) struggle at high speeds due to global reconstruction. Yopov2 [25] performs better, but its multi-target variant shows degraded robustness. Our method achieves the highest success rates, benefiting from joint perception and control design. Fig. 6 visualizes target position distribution in the tracker horizontal FOV. At the target speed of 3 m/s, Vis and Elas already show dispersed localization. Yopov2 drifts at 6 m/s. Our method maintains a sharply concentrated heatmap, showing superior stability and view consistency, mainly due to the tracking-aware visibility loss guiding trajectory planning to preserve in-frame target visibility.

4.2. Long-Range Tracking Performance

We evaluate long-term tracking in cluttered environments with dense pillars and irregular walls. The target performs abrupt maneuvers (sharp turns, stops, reversals) over 1–1.5 km flights, with 10 randomized trajectories per method. Long-range success rate is the fraction of flight time with uninterrupted, collision-free target tracking.

Method	Vis. [30]		Elas. [15]		Yopov2. [25]		Ours	
Type	Ave	Min	Ave	Min	Ave	Min	Ave	Min
3m/s	2.31	0.03	**2.34**	0.05	1.87	0.38	2.00	**0.63**
5m/s	/	/	/	/	1.75	0.11	**2.02**	**0.13**

Table 3. Average (Ave) and minimum (Min) UAV–obstacle distances (m) during valid tracking.

Method	ST (s) ↓	Dis (m) ↓	Vel (m/s) ↑	CT (ms) ↓
RACER [46]	176.62	293.19	1.66	43.1
FALCON [41]	146.67	**265.48**	1.81	37.2
Ours	**54.49**	268.67	**4.93**	**8.6**

Table 4. Active search comparison. "ST" denotes search time, "CT" cycle time. Our method achieves $\sim 3\times$ faster search with similar path length and the lowest computation latency.

Table 2 shows that our method maintains near-perfect tracking across speeds, whereas Vis. [30] and Elas. [15] degrade due to reliance on explicit mapping and optimization. Yopov2 [25] fails to recover effectively after target loss. Our unified search–tracking mechanism enables fast re-localization and robust recovery, yielding superior performance under all conditions. Fig. 7 shows projections of experimental high-speed (5 m/s) trajectories on the map: traditional methods lose the target early; Yopov2 and ours experience loss during sudden turns, only ours quickly resumes tracking. Smooth switching between search and track ensures continuous target re-acquisition.

Safety analysis shows Vis. [30] and Elas. [15] occasionally achieve larger average distances at low speed but suffer collisions due to execution mismatch in Table 3. Our method maintains higher minimum clearance and smoother motion, demonstrating both robust tracking and safe operation in obstacle-dense environments.

4.3. Active Search Evaluation

We assess the active search ability of our UAST against two state-of-the-art UAV search methods, RACER [46] and FALCON [41]. The environment is identical to the long-term tracking setup, where the target is initially over 100 m away, requiring autonomous searching.

As shown in Table 4, our method completes the search in about one-third of the time of RACER and FALCON while maintaining similar path length. This efficiency results from the rule-based point search policy, which avoids global mapping and costly frontier exploration. By coupling perception with motion generation, the UAV is guided reactively toward promising regions at nearly 5 m/s, enabling fast and reliable target re-localization.

4.4. Ablation Study

We conduct ablation experiments at a 5 m/s speed for both short- and long-term tracking to evaluate each component.

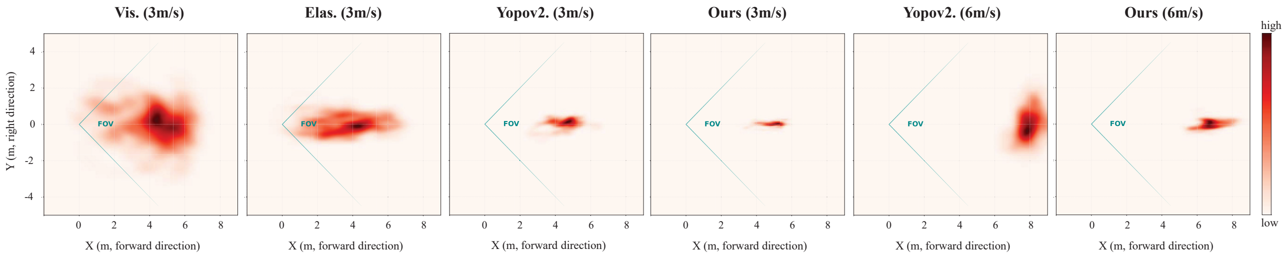


Figure 6. Target distribution in the Tracker-UAV's FOV at 3 m/s and 6 m/s. Higher intensity indicates more frequent target appearances.

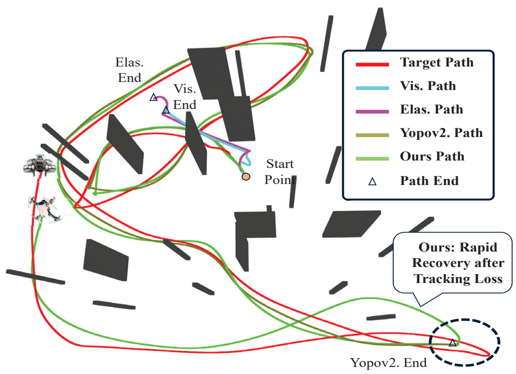


Figure 7. Trajectory comparison at 5 m/s. Vis. [30] and Elas. [15] lose the target early. Yopov2 [25] and our method experience temporary loss after sharp target turns, but only our UAST rapidly re-acquires the target via unified active search and tracking.

Method	Short SR ↑	Long SR ↑	Center Dis (m) ↓
W/O search	0.88	0.52	**0.57**
W/O track loss	0.85	0.82	0.82
W/O data build	0.80	0.76	0.63
W/O guid point	0.70	0.33	0.69
Ours	**0.93**	**0.89**	**0.57**

Table 5. Ablation results on short-/long-term success rate (SR) and average point-to-center distance within the FoV. Lower center distance indicates more stable target alignment.

The following modules are removed individually: Search mechanism (W/O search), Tracking-aware loss (W/O track loss), Data construction strategy (W/O data build), and Point search policy (W/O guid point).

As shown in Table 5, removing the search mechanism causes the largest degradation in Long SR, underscoring its role in target re-acquisition. The tracking-aware loss notably reduces FoV center deviation, demonstrating its effectiveness in maintaining stable, centered target alignment. The data generation scheme contributes to robustness by improving generalization. Excluding the guidance policy leads to severe drops across all metrics, confirming the necessity of coordinated search–tracking behaviors.

4.5. Real-world Validation

We validate our framework on a quadrotor equipped with an NVIDIA Jetson NX and an Intel RealSense D435 RGB-D camera, using Fast-LIO [36] for localization. Tests are conducted in an underground parking lot and an outdoor forest. The UAV successfully performs autonomous search and persistent tracking, demonstrating robust real-time performance under onboard constraints and smooth transfer from simulation to reality. Fig. 8 provides an illustrative reconstruction of a flight trajectory in the parking lot.

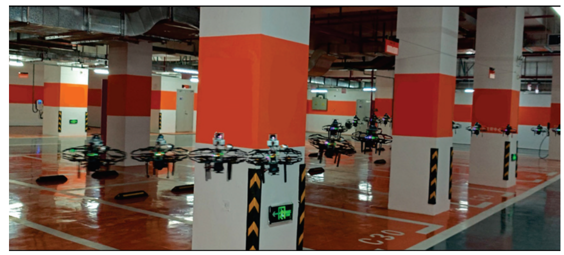


Figure 8. Real-world flight demonstration of the quadrotor in an underground parking lot.

5. Conclusion

We present a unified search–tracking framework for active UAV target pursuit that operates directly on RGB-D perception without explicit mapping or multi-stage planning. By combining a tracking-aware visibility loss with a spatially consistent data generation strategy, the system achieves robust, geometry-aware control. Extensive simulation and real-world experiments show superior tracking stability, search efficiency, and flight robustness, while maintaining real-time onboard performance. However, the framework relies on forward-facing RGB-D sensing, which restricts safe retreat during close interactions. Future work will explore multi-view perception to enhance robustness in such scenarios.

Acknowledgement

This work is supported by Natural Science Foundation of Anhui Province under Contract 2508085Y040, National Natural Science Foundation of China under Contract 62472141, the Youth Innovation Promotion Association CAS, and the GPU cluster built by MCC Lab of USTC and the Supercomputing Center of USTC.

References

[1] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon" next-best-view" planner for 3d exploration. In *ICRA*, pages 1462–1468. IEEE, 2016. 2

[2] Chao Cao, Hongbiao Zhu, Zhongqiang Ren, Howie Choset, and Ji Zhang. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. *Science Robotics*, 8(80):eadf0970, 2023. 2

[3] Ziang Cao, Changhong Fu, Junjie Ye, Bowen Li, and Yiming Li. Hift: Hierarchical feature transformer for aerial tracking. In *ICCV*, pages 15457–15466, 2021. 2

[4] Ziang Cao, Ziyuan Huang, Liang Pan, Shiwei Zhang, Ziwei Liu, and Changhong Fu. Tctrack: Temporal contexts for aerial tracking. In *CVPR*, pages 14798–14808, 2022. 2

[5] Jing Chen, Tianbo Liu, and Shaojie Shen. Tracking a moving target in cluttered environments using a quadrotor. In *IROS*, pages 446–453. IEEE, 2016. 1

[6] Xinyi Chen, Yichen Zhang, Hetai Zou, Junzhe Wang, and Shaojie Shen. Active contact engagement for aerial navigation in unknown environments with glass. *IEEE Robotics and Automation Letters*, 2025. 2

[7] Tung Dang, Frank Mascarich, Shehryar Khattak, Christos Papachristos, and Kostas Alexis. Graph-based path planning for autonomous robotic exploration in subterranean environments. In *IROS*, pages 3105–3112. IEEE, 2019. 1

[8] Mihir Dharmadhikari, Tung Dang, Lukas Solanka, Johannes Loje, Huan Nguyen, Nikhil Khedekar, and Kostas Alexis. Motion primitives-based path planning for fast and agile exploration using aerial robots. In *ICRA*, pages 179–185. IEEE, 2020.

[9] Chen Feng, Haojia Li, Mingjie Zhang, Xinyi Chen, Boyu Zhou, and Shaojie Shen. Fc-planner: A skeleton-guided planning framework for fast aerial coverage of complex 3d scenes. In *ICRA*, pages 8686–8692. IEEE, 2024. 1

[10] Fei Gao, William Wu, Yi Lin, and Shaojie Shen. Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial. In *ICRA*, pages 344–351. IEEE, 2018. 1

[11] Yuman Gao, Jialin Ji, Qianhao Wang, Rui Jin, Yi Lin, Zhimeng Shang, Yanjun Cao, Shaojie Shen, Chao Xu, and Fei Gao. Adaptive tracking and perching for quadrotor in dynamic scenarios. *IEEE Transactions on Robotics*, 40:499–519, 2023. 1, 2

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015. 2

[13] Zhichao Han, Ruibin Zhang, Neng Pan, Chao Xu, and Fei Gao. Fast-tracker: A robust aerial system for tracking agile target in cluttered environments. In *ICRA*, pages 328–334. IEEE, 2021. 1

[14] Dries Hulens and Toon Goedemé. Autonomous flying cameraman with embedded person detection and tracking while applying cinematographic rules. In *CRV*, pages 56–63. IEEE, 2017.

[15] Jialin Ji, Neng Pan, Chao Xu, and Fei Gao. Elastic tracker: A spatio-temporal trajectory planner for flexible aerial tracking. In *ICRA*, pages 47–53. IEEE, 2022. 1, 2, 7, 8

[16] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018. 2

[17] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4282–4291, 2019. 2

[18] Jun-Ming Li, Ching-Wen Chen, and Teng-Hu Cheng. Motion prediction and robust tracking of a dynamic and temporarily-occluded target by an unmanned aerial vehicle. *IEEE Transactions on Control Systems Technology*, 29(4): 1623–1635, 2020. 1

[19] Shuiwang Li, Yangxiang Yang, Dan Zeng, and Xucheng Wang. Adaptive and background-aware vision transformer for real-time uav tracking. In *ICCV*, pages 13989–14000, 2023. 2

[20] Yiming Li, Changhong Fu, Fangqiang Ding, Ziyuan Huang, and Geng Lu. Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization. In *CVPR*, pages 11923–11932, 2020. 1, 2

[21] Mengyuan Liu, Yuelong Wang, Qiangyu Sun, and Shuiwang Li. Global filter pruning with self-attention for real-time uav tracking. In *BMVC*, page 861, 2022. 1

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 2

[23] Antonio Loquercio, Elia Kaufmann, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Learning high-speed flight in the wild. *Science Robotics*, 6(59): eabg5810, 2021. 1, 2

[24] Junjie Lu, Xuewei Zhang, Hongming Shen, Liwen Xu, and Bailing Tian. You only plan once: A learning-based one-stage planner with guidance learning. *IEEE Robotics and Automation Letters*, 9(7):6083–6090, 2024. 7

[25] Junjie Lu, Yulin Hui, Xuewei Zhang, Wencan Feng, Hongming Shen, Zhiyu Li, and Bailing Tian. Yopov2-tracker: An end-to-end agile tracking and navigation framework from perception to action. *arXiv*, 2025. 1, 2, 7, 8

[26] Siyu Ma, Yuting Liu, Dan Zeng, Yaxin Liao, Xiaoyu Xu, and Shuiwang Li. Learning disentangled representation in pruning for real-time uav tracking. In *ACML*, pages 690–705. PMLR, 2023. 1

[27] Charles Richter, Adam Bry, and Nicholas Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *ISRR*, pages 649–666. Springer, 2016. 1

[28] Zhi Tian, Xiangxiang Chu, Xiaoming Wang, Xiaolin Wei, and Chunhua Shen. Fully convolutional one-stage 3d object detection on lidar range images. *NeurIPS*, 35:34899–34911, 2022. 2

[29] Kunyu Wang, Xueyang Fu, Yukun Huang, Chengzhi Cao, Gege Shi, and Zheng-Jun Zha. Generalized uav object detection via frequency domain disentanglement. In *CVPR*, pages 1064–1073, 2023. 1

[30] Qianhao Wang, Yuman Gao, Jialin Ji, Chao Xu, and Fei Gao. Visibility-aware trajectory optimization with application to aerial tracking. In *IROS*, pages 5249–5256. IEEE, 2021. 1, 2, 7, 8

[31] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *CVPR*, pages 9697–9706, 2023. 1

[32] Wanying Wu, Pengzhi Zhong, and Shuiwang Li. Fisher pruning for real-time uav tracking. In *IJCNN*, pages 1–7. IEEE, 2022.

[33] You Wu, Xucheng Wang, Dan Zeng, Hengzhou Ye, Xiaolan Xie, Qijun Zhao, and Shuiwang Li. Learning motion blur robust vision transformers with dynamic early exit for real-time uav tracking. *arXiv*, 2024.

[34] You Wu, Xucheng Wang, Xiangyang Yang, Mengyuan Liu, Dan Zeng, Hengzhou Ye, and Shuiwang Li. Learning occlusion-robust vision transformers for real-time uav tracking. In *CVPR*, pages 17103–17113, 2025. 2

[35] Fei Xie, Chunyu Wang, Guangting Wang, Wankou Yang, and Wenjun Zeng. Learning tracking representations via dual-branch fully transformer networks. In *ICCV*, pages 2688–2697, 2021. 1

[36] Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021. 8

[37] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, pages 12549–12556, 2020. 2, 5

[38] Fan Yang, Dung-Han Lee, John Keller, and Sebastian Scherer. Graph-based topological exploration planning in large-scale 3d environments. In *ICRA*, pages 12730–12736. IEEE, 2021. 1

[39] Junjie Ye, Changhong Fu, Guangze Zheng, Danda Pani Paudel, and Guang Chen. Unsupervised domain adaptation for nighttime aerial tracking. In *CVPR*, pages 8896–8905, 2022. 1

[40] Jiajie Yu, Hao Shen, Jianyu Xu, and Tong Zhang. Echo: An efficient heuristic viewpoint determination method on frontier-based autonomous exploration for quadrotors. *IEEE Robotics and Automation Letters*, 8(8):5047–5054, 2023. 1, 2

[41] Yichen Zhang, Xinyi Chen, Chen Feng, Boyu Zhou, and Shaojie Shen. Falcon: Fast autonomous aerial exploration using coverage path guidance. *IEEE Transactions on Robotics*, 2024. 1, 2, 7

[42] Yuang Zhang, Yu Hu, Yunlong Song, Danping Zou, and Weiyao Lin. Back to newton's laws: Learning vision-based agile flight via differentiable physics. *arXiv*, 2024. 1, 2

[43] Yinghao Zhao, Li Yan, Hong Xie, Jicheng Dai, and Pengcheng Wei. Autonomous exploration method for fast unknown environment mapping by using uav equipped with limited fov sensor. *IEEE Transactions on Industrial Electronics*, 71(5):4933–4943, 2023. 1

[44] Boyu Zhou, Fei Gao, Luqi Wang, Chuhao Liu, and Shaojie Shen. Robust and efficient quadrotor trajectory generation for fast autonomous flight. *IEEE Robotics and Automation Letters*, 4(4):3529–3536, 2019. 1

[45] Boyu Zhou, Yichen Zhang, Xinyi Chen, and Shaojie Shen. Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters*, 6(2):779–786, 2021. 2

[46] Boyu Zhou, Hao Xu, and Shaojie Shen. Racer: Rapid collaborative exploration with a decentralized multi-uav system. *IEEE Transactions on Robotics*, 39(3):1816–1835, 2023. 1, 2, 7

[47] Xin Zhou, Zhepei Wang, Hongkai Ye, Chao Xu, and Fei Gao. Ego-planner: An esdf-free gradient-based local planner for quadrotors. *IEEE Robotics and Automation Letters*, 6(2): 478–485, 2020. 1