UNI-FLY360: LEARNING OMNIDIRECTIONAL OBSTACLE AVOIDANCE WITHIN END-TO-END FLIGHT

Anonymous authors
Paper under double-blind review

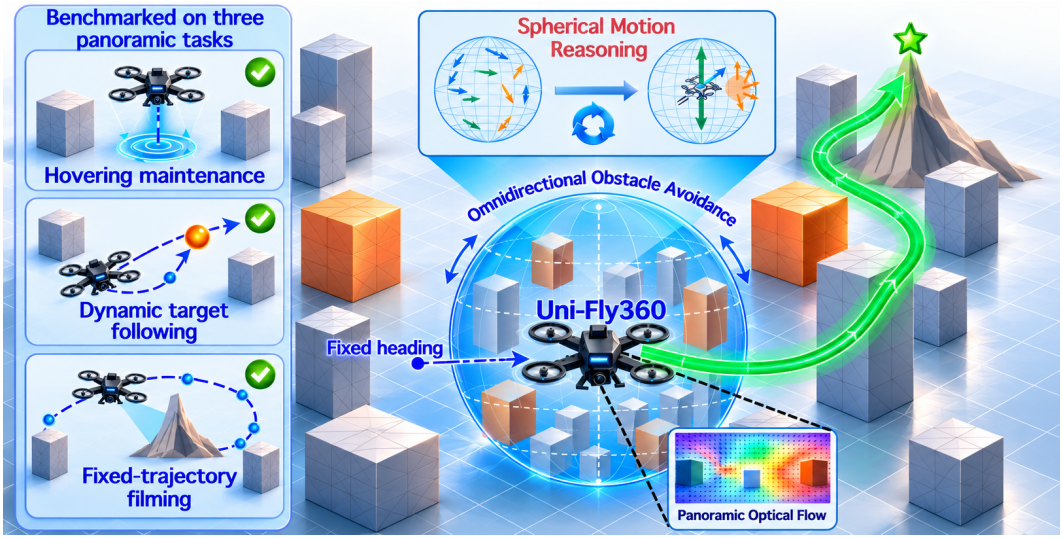

Figure 1: We propose **Uni-Fly360**, an end-to-end optimized framework for omnidirectional UAV obstacle avoidance from panoramic vision. It maps consecutive panoramic RGB observations and state to continuous flight commands. Through differentiable spherical motion reasoning, **Uni-Fly360** jointly optimizes the optical-flow estimator and the recurrent policy, exploiting geometry and motion cues for robust omnidirectional obstacle avoidance while mitigating sim-to-real discrepancies.

ABSTRACT

Reliable autonomous flight is essential for unmanned aerial vehicles (UAVs), yet many tasks demand omnidirectional obstacle avoidance: when the flight direction deviates from a task-specific heading, obstacles may approach from arbitrary directions. This requirement exposes the limited coverage of forward-facing sensors and motivates panoramic perception and control. Existing approaches, however, either suffer from restricted visual coverage or lack designs tailored to panoramic imagery. We propose Uni-Fly360, an end-to-end framework for omnidirectional obstacle avoidance from panoramic image sequences. It adopts optical flow as an intermediate representation that captures motion cues while reducing dependence on domain-specific appearance, thereby mitigating the sim-to-real gap. To disentangle ego-motion from object motion in optical flow, we introduce a differentiable spherical motion reasoning module that exploits antipodal consistency and full-sphere constraints to suppress rotation, estimate the ego-translation direction, and construct local motion residuals. Its differentiable formulation integrates spherical geometric reasoning into the joint optimization of perception and control. Experiments on three representative panoramic tasks show that Uni-Fly360 achieves state-of-the-art performance and transfers well to real-world videos. Code and models will be released.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Autonomous flight is a core capability of unmanned aerial vehicles (UAVs), enabling them to execute tasks in complex environments (Tang & Kumar, 2018). Existing methods, ranging from modular perception–planning–control pipelines (Zhou et al., 2021b) to end-to-end learned policies (Loquercio et al., 2021), have demonstrated effective autonomous navigation and obstacle avoidance.

However, UAVs moving with six degrees of freedom while holding a task-specific viewpoint or heading may encounter obstacles from arbitrary directions. This exposes the limited coverage of forward-facing perception and calls for omnidirectional obstacle avoidance for safe task execution. To this end, recent efforts have expanded UAV visual coverage through multi-view and panoramic sensing (Liu et al., 2024). In particular, Fly360 (Zhang et al., 2026) combines pretrained panoramic depth estimation with a learned control policy in a two-stage perception–control pipeline, enabling omnidirectional obstacle avoidance even when the flight direction differs from the UAV's heading.

Nevertheless, it still retain two-stage training pipeline, in which the perception model is pretrained to optimize geometric prediction accuracy and then kept frozen during policy optimization. While navigation prioritizes task progress and safe obstacle clearance, this decoupled training prevents task-specific losses from updating the perception model. More broadly, such learning-based methods rely on simulation for large-scale policy training, which remains a source of the sim-to-real gap. The limitations motivate our central question: *How can we unify this pipeline with sim-to-real transfer?*

To answer this question, we must address the key challenge: integrating image-based perception into large-scale policy training while maintaining high performance and real-world generalization. Encouragingly, optical flow naturally reconciles these goals. Firstly, it captures the image motion induced by the UAV's ego-motion and surrounding objects, providing cues directly relevant to collision avoidance. Secondly, effective flow estimators can be learned from non-photorealistic synthetic imagery(Sun et al., 2021). Building on this insight, we propose **Uni-Fly360**, which adopts optical flow as the interface between panoramic perception and control and trains on non-photorealistic panoramic sequences from a differentiable simulator (Zhang et al., 2025a). The flow estimator is optimized jointly with a recurrent policy so that navigation and collision-avoidance objectives shape both modules, while the appearance-invariant representation narrows the sim-to-real gap.

However, a new issue arises, it entangles UAV rotation, translation, and independent object motion, making it hard for the policy to separate ego-induced motion from moving obstacles. We therefore introduce a differentiable spherical motion geometry module that exploits geometric relationships across viewing directions: it cancels rotation via antipodal motion consistency (Lim & Barnes, 2008), aggregates flow constraints over the sphere to estimate a global translation model from the predominantly static background, and computes pixel-wise directional residuals against this model to highlight motion unexplained by ego-motion. Since every step is differentiable, navigation losses propagate through these geometric constraints and refine the flow estimator for obstacle avoidance.

Our contributions are summarized as follows:

- We identify optical flow as an effective intermediate representation for omnidirectional obstacle avoidance: it captures collision-relevant motion with reduced dependence on scene appearance and can be learned from non-photorealistic synthetic panoramas, helping bridge the sim-to-real gap and supporting joint perception–control optimization.
- We propose **Uni-Fly360**, an panorama-to-policy framework that jointly optimizes a panoramic optical-flow estimator and a recurrent policy in a differentiable simulator, enabling omnidirectional obstacle avoidance. Our approach better aligns perception with navigation objectives, improving task performance and supporting sim-to-real transfer.
- We evaluate Uni-Fly360 on three representative navigation tasks, demonstrating improved obstacle avoidance over the evaluated baselines in simulation, and assess the transfer of its learned motion representations to real-world panoramic video observations.

2 RELATED WORK

Visual Perception for UAV Obstacle Avoidance. UAV obstacle avoidance relies on onboard perception systems, including LiDAR (Ren et al., 2025; Schleich & Behnke, 2022) and vision-based modalities such as RGB images (Loquercio et al., 2018; Gandhi et al., 2017), depth maps (Loquer-

cio et al., 2021; Yu et al., 2025b; Bhattacharya et al., 2025), and optical flow (Bouwmeester et al., 2023). While RGB and depth encode appearance and geometry, respectively, optical flow provides motion-centric cues for collision avoidance (de Croon et al., 2021; Xiao et al., 2021) and has enabled simulation-trained navigation policies to generalize directly to real-world environments (Hu et al., 2025; Deng et al., 2026). Meanwhile, limited sensing coverage restricts UAV navigation, as six-degree-of-freedom motion exposes the vehicle to obstacles from arbitrary directions (Zhang et al., 2026). Forward-facing cameras are computationally efficient but cannot perceive obstacles outside their limited field of view (Song et al., 2023; Zhou et al., 2021a), motivating multi-view (Liu et al., 2024; Deng et al., 2025; Won et al., 2019; Meuleman et al., 2021; Li et al., 2022a) and panoramic sensing approaches (Bhandari et al., 2022; Shi et al., 2023; Liu et al., 2025; Li et al., 2022b; Yuan & Richardt, 2021). Recent panoramic methods extend perception view and enable omnidirectional avoidance (Artizzu et al., 2022; Zhang et al., 2026), but only focus on increasing visual coverage rather than exploiting the underlying global motion structure in panoramic observations.

Policy Learning for UAV Obstacle Avoidance. Learning-based UAV navigation, such as Imitation learning and Reinforcement learning, aims to reduce the complexity of traditional perception–planning–control pipelines (Zhou et al., 2019; 2021b; Tordesillas et al., 2022) by directly mapping observations to control commands (Geles et al., 2024; Wu et al., 2025; Zhang et al., 2025a; Hu et al., 2023). Imitation learning methods learn navigation behaviors from expert demonstrations (Ross et al., 2013; Loquercio et al., 2021; Bhattacharya et al., 2025) and achieve efficient policy optimization (Xing et al., 2025; Messikommer et al., 2025), but their performance depends on demonstration coverage and can suffer from distribution shifts during deployment (Ross et al., 2011; Ciftci et al., 2025). Reinforcement learning avoids expert supervision by optimizing policies through environment interaction and has achieved strong results in simulated UAV navigation (Sadeghi & Levine, 2017; Kim et al., 2022; Yu et al., 2025b; Zhang et al., 2025b; Xu et al., 2025). However, learning effective policies often rely on large-scale simulation (Song et al., 2021; Li et al., 2025), which can introduce sim-to-real gaps (Loquercio et al., 2020; Yu et al., 2025a; Kim et al., 2025; Dutta et al., 2026). Moreover, existing end-to-end approaches typically use pretrained depth or optical-flow estimators as fixed inputs, leaving visual representations independently optimized from the navigation objective (Kim et al., 2022; Hu et al., 2025; Deng et al., 2026; Zhang et al., 2026).

3 METHOD

3.1 OVERVIEW

Learning omnidirectional obstacle avoidance from images requires large-scale policy training in simulation, which must yield policies that both achieve high avoidance performance and generalize to the real world. To this end, we propose **Uni-Fly360**, an end-to-end optimized framework for omnidirectional UAV obstacle avoidance based on panoramic vision (Fig. 2). After formulating the problem (Sec. 3.2), we introduce the key insight behind Uni-Fly360 (Sec. 3.3): panoramic optical flow provides motion cues for omnidirectional obstacle avoidance, reduces the policy's reliance on scene appearance, and can be learned from non-photorealistic imagery. These properties enable joint perception–control training in a differentiable simulator (Sec. 3.4) while mitigating appearance-induced sim-to-real gaps. However, raw optical flow entangles camera and object motion. We therefore introduce a differentiable Spherical Motion Reasoning module (Sec. 3.5) that exploits antipodal constraints to suppress rotation and estimate a global translation axis, and then computes local residuals that expose deviations from this motion model. These complementary geometric cues are fed to the navigation policy, and navigation losses, together with auxiliary flow and geometry supervision, jointly refine perception for obstacle avoidance.

3.2 PROBLEM FORMULATION

We formulate the task as end-to-end autonomous UAV flight with omnidirectional obstacle avoidance based on panoramic visual observations. Given a goal $\boldsymbol{g}_t$ and a desired heading $\chi_t^\star$, the UAV must make progress toward the goal while avoiding surrounding obstacles. The desired heading is specified independently of the motion direction, allowing the vehicle to maintain a task-specific viewpoint during avoidance maneuvers. At each control step t, the UAV receives an equirectangular

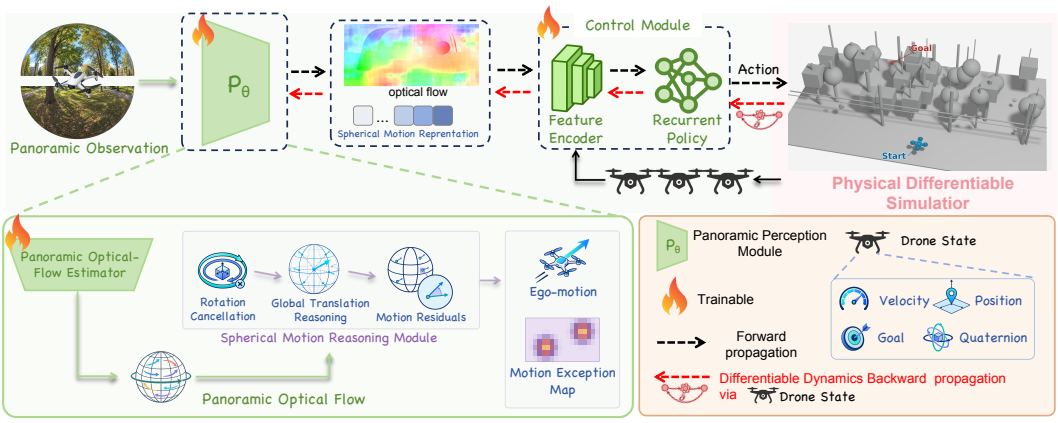

Figure 2: **Illustation of Uni-Fly360.** Our framework consists of three components. The perception module estimates optical flow from consecutive panoramic RGB observations and applies spherical motion reasoning to extract geometry and motion cues. The recurrent control module fuses these cues with the UAV ego state to produce flight commands. The differentiable physics simulator generates synthetic observations and closed-loop trajectories, allowing navigation gradients to flow through the dynamics and controller into perception.

RGB panorama $I_t^{H\times W\times 3}$ and obtains its navigation state

$$\boldsymbol{s}_t = [\boldsymbol{p}_t^{\mathcal{W}}, \boldsymbol{q}_t, \boldsymbol{v}_t^{\mathcal{B}}], \qquad \|\boldsymbol{q}_t\| = 1, \tag{1}$$

where $\mathcal{W}$ and $\mathcal{B}$ denote the world and body frames, respectively. Here, $\boldsymbol{p}_t^{\mathcal{W}}$ is the position, $\boldsymbol{q}_t \in \mathbb{R}^4$ is the orientation quaternion, and $\boldsymbol{v}_t^{\mathcal{B}}$ is the body-frame translational velocity. Then a perception-to-policy model π_Θ maps consecutive panoramic observations and navigation inputs to a continuous control command defined as

$$\boldsymbol{y}_t = \pi_\Theta(\boldsymbol{I}_t, \boldsymbol{s}_t, \boldsymbol{g}_t), \tag{2}$$

where $\boldsymbol{y}_t$ denotes the predicted action vector, which is subsequently converted into low-level control commands for execution on the real UAV.

3.3 Learning Optical Flow from Synthetic RGB

Given panoramic RGB observations, the policy must extract cues that are informative for collision avoidance yet robust to the appearance gap between simulation and the real world. Raw RGB is dominated by scene textures and illumination, which vary widely across environments and are costly to render faithfully at scale. We therefore adopt optical flow as the intermediate representation between perception and control. Optical flow captures inter-frame image displacements induced by camera and object motion, providing collision-relevant cues without explicit scene reconstruction. Since it encodes motion rather than appearance, it reduces the policy's dependence on textures and illumination and thereby mitigates appearance-related sim-to-real discrepancies.

However, learning optical flow from RGB, raises the question of what training imagery is required. Prior work provides strong evidence that photorealism is not necessary (Fig. 3(a) and (b)). Dosovitskiy et al. (2015) demonstrated generalization from the non-photorealistic FlyingChairs dataset, and subsequent work showed that greater rendering realism does not necessarily improve estimation accuracy (Mayer et al., 2018; 2016). Sun et al. (2021) further showed layered synthetic images with optimized shape, appearance, and motion parameters yield effective training data. This motivates a synthetic data design that emphasizes appearance and motion diversity rather than photorealistic rendering (Fig. 3(c)). To integrate this data-generation strategy with joint perception–control learning, we implement it within a differentiable flight simulation environment, described next.

3.4 Differentiable Simulation Environment

We build on the differentiable flight simulator of Zhang et al. (2025a), which combines a point-mass quadrotor model augmented with aerodynamic drag and a delayed, smoothed thrust response,

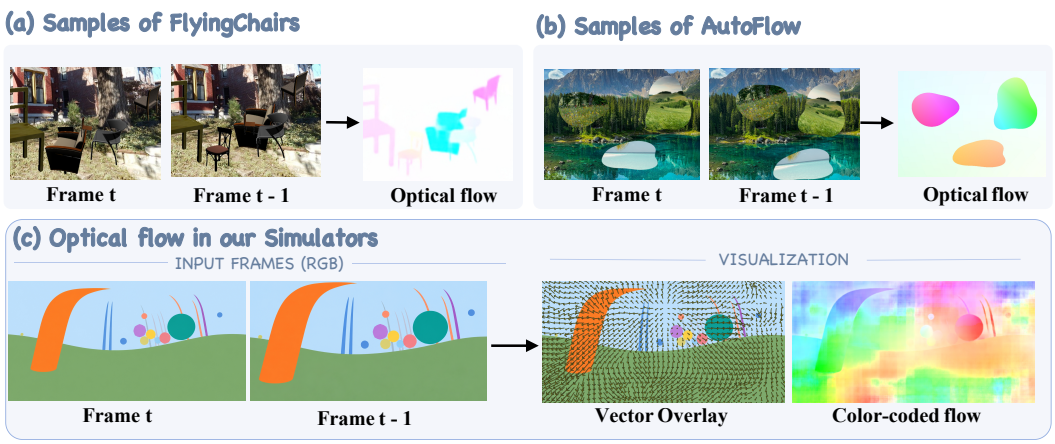

Figure 3: **Optical flow from non-photorealistic synthetic imagery.** (a,b) Training samples from FlyingChairs and AutoFlow. (c) An illustrative example from our simulator, with optical flow visualized as a vector overlay and a color-coded map. These examples illustrate how synthetic imagery with simplified appearance can provide structured motion cues for optical-flow learning.

a depth-rendering engine, and randomly arranged geometric primitives as obstacles. Its differentiable dynamics allow navigation losses, including velocity tracking, obstacle avoidance, and control smoothness, to backpropagate through simulated trajectories to the policy parameters. To supply visual inputs for joint perception–control optimization, we extend the simulator with temporally consistent panoramic RGB synthesis: six 90° depth faces are rendered and resampled into an ERP depth map D_t, which is used solely for image synthesis and never exposed to the policy. The RGB renderer assigns separate randomized color palettes to the obstacles, background and ground regions, then applies depth-dependent shading, multi-frequency spherical textures, edge attenuation, fog, exposure, gamma, per-channel gains, and sensor noise. The visual domain is sampled once per rollout and held fixed over time, so that optical flow follows physical motion within a rollout while appearance is substantially randomized across different rollouts.

3.5 Spherical Optical Flow Representation and Motion Reasoning

We represent panoramic observations in equirectangular projection (ERP) on an $H \times W$ grid and, following Sec. 3.3, estimate pixel displacements $(\Delta u_t, \Delta v_t)$ between consecutive observations (I_{t-1}, I_t) with PanoRAFT (Shi et al., 2023). To express motion in resolution-independent angular units, we convert these displacements as

$$\Delta\theta_t = \frac{2\pi}{W}\Delta u_t, \qquad \Delta\phi_t = \frac{\pi}{H}\Delta v_t. \tag{3}$$

After normalization, the angular displacements provide bounded optical-flow features for the policy and also serve as input to the motion reasoning described next.

Raw optical flow entangles camera translation, camera rotation, and independently moving objects, obscuring the motion cues most relevant to collision avoidance. To address this, we introduce the **Spherical Motion Reasoning (SMR)** module, which exploits panoramic motion constraints to extract a global translation-axis descriptor Q_{ego} and a dense local motion-residual map D_{dyn}.

Global translation-axis reasoning. We lift the angular displacements to three-dimensional tangent vectors via the spherical logarithmic map, $\boldsymbol{f}_i = \mathrm{Log}_{\boldsymbol{s}_i}(\boldsymbol{s}'_i)$, where $\boldsymbol{s}_i$ and $\boldsymbol{s}'_i$ denote the viewing directions before and after the predicted displacement. Under a first-order static-scene motion model, averaging the flow at antipodal directions suppresses rotation and yields a local translation constraint:

$$\boldsymbol{g}_i = \frac{\boldsymbol{f}(\boldsymbol{s}_i) + \boldsymbol{f}(-\boldsymbol{s}_i)}{2}, \qquad \boldsymbol{c}_i = \boldsymbol{s}_i \times \boldsymbol{g}_i, \tag{4}$$

whose normal satisfies $\boldsymbol{c}_i^\top \boldsymbol{v} \approx 0$ for the camera translation $\boldsymbol{v}$. Aggregating these complementary constraints over the viewing sphere gives

$$M = \frac{\sum_i w_i \boldsymbol{c}_i \boldsymbol{c}_i^\top}{\sum_i w_i + \epsilon}, \tag{5}$$

where w_i downweights weak flow and compensates for ERP sampling density. When static-scene constraints dominate, the minimum-eigenvalue direction of M estimates the camera-translation axis. We encode this axis with a differentiable, sign-invariant descriptor,

$$Q_{\text{ego}} = \frac{\exp(-\overline{M}/\tau)}{\text{tr}[\exp(-\overline{M}/\tau)] + \epsilon}, \qquad \overline{M} = \frac{M}{\text{tr}(M) + \epsilon}, \tag{6}$$

where exp denotes the matrix exponential and $\tau > 0$ controls the concentration around the minimum-eigenvalue direction.

Local motion residual. Let $\widehat{\boldsymbol{p}}_i$ denote the unit tangential direction at $\boldsymbol{s}_i$ predicted by the estimated translation axis. We measure directional deviations from this translation model as

$$D_{\text{dyn},i} = w_i^{\text{loc}} \sqrt{\max\left(1 - \frac{(\boldsymbol{f}_i^\top \widehat{\boldsymbol{p}}_i)^2}{\|\boldsymbol{f}_i\|_2^2 + \epsilon}, 0\right) + \epsilon}, \tag{7}$$

where w_i^{loc} suppresses regions in which translational motion is weakly observable; larger residuals indicate stronger local departures from the global translation model. Together, Q_{ego} and D_{dyn} supply the policy with global motion structure and spatially resolved motion-inconsistency cues. Estimator configurations, normalization details, and further geometric analysis are provided in Appendices A.1 and A.2.

3.6 End-to-End Optimization for Omnidirectional Obstacle Avoidance

Control policy. We parameterize the control policy π_ψ as a recurrent network conditioned on panoramic motion features and the UAV ego state. The perception module $\mathcal{P}_\theta$ provides the angular-flow representation $\widehat{F}_t$, the global translation-axis descriptor $Q_{\text{ego},t}$, and the local motion-residual map $Q_{\text{Dyn},t}$ defined in Sec. 3.5. Together with the ego state $\boldsymbol{s}_t$ and the previous hidden state $\boldsymbol{h}_{t-1}$, these features determine the policy output:

$$(\boldsymbol{u}_t, \boldsymbol{h}_t) = \pi_\psi(\mathcal{P}_\theta(I_{t-1}, I_t), \boldsymbol{s}_t, \boldsymbol{h}_{t-1}). \tag{8}$$

The hidden state is reset at the start of each rollout. The output $\boldsymbol{u}_t \in \mathbb{R}^6$ is transferred to a acceleration $\hat{\boldsymbol{a}}_t$ and expressed in a yaw-aligned level frame.

Perception-to-Control Optimization via Differentiable Simulation. We jointly optimize the perception module and recurrent control policy on closed-loop trajectories generated from online-sampled scenes and synthesized panoramic RGB observations. The simulator differentiates trajectory-level objectives through the vehicle dynamics, while backpropagation through time propagates these gradients through the recurrent controller into perception. Navigation gradients reach the optical-flow estimator through both the flow features and the differentiable spherical geometry module. The joint training objective is

$$\mathcal{L} = \mathcal{L}_{\text{nav}} + \mathcal{L}_{\text{aux}}, \tag{9}$$

which is minimized over the perception parameters θ and policy parameters ψ. The navigation objective $\mathcal{L}_{\text{nav}}$ encourages task-oriented flight with obstacle avoidance and smooth control. However, navigation supervision alone may drive predicted flow toward task-specific features that no longer faithfully represent image motion. We therefore introduce $\mathcal{L}_{\text{aux}}$, using ground-truth optical flow and camera translation from the simulator to encourage motion accuracy and geometric consistency. Detailed loss definitions are provided in Appendix A.3.

This joint formulation allows navigation feedback to adapt perception to obstacle avoidance while auxiliary supervision preserves its intended geometric meaning. Together with online scene generation and RGB synthesis, it enables training on diverse motion cues without requiring photorealistic rendering, helping mitigate appearance-related sim-to-real discrepancies.

Table 1: Quantitative performance comparison across three tasks for methods with different fields of view and intermediate representations. "–" indicates not applicable. Best and second-best results are marked in **bold** and underline, respectively.

Method	Visual configuration		Hovering Maintenance		Dynamic Target Following		Fixed-Trajectory Filming	
	View	Representation	SR ↑	CT (s) ↓	SR ↑	CT (s) ↓	SR ↑	CT (s) ↓
DiffPhys	Forward	Depth	3/40	13.4	7/40	48.9	–	–
ViT-Fly	Forward	Depth	1/40	14.2	6/40	50.6	–	–
OmniNxt-4	Multi-view	Depth	4/40	11.4	12/40	27.4	3/40	73.7
OmniNxt-6	Multi-view	Depth	13/40	4.6	21/40	13.0	24/40	4.9
Fly360	Panoramic	Depth	28/40	**1.19**	33/40	2.2	**37/40**	**0.27**
Pixel Motion	Forward	Flow	0/40	14.5	1/40	64.7	–	–
Trans. Flow	Forward	Flow	0/40	16.0	0/40	68.9	–	–
Uni-Fly360 (ours)	Panoramic	Flow	**38/40**	1.86	**40/40**	**0**	**37/40**	2.8

4 Experiments

4.1 Experimental Setup

We evaluate Uni-Fly360 on obstacle avoidance in both static and dynamic environments. Following the Fly360 benchmark (Zhang et al., 2026), the evaluation comprises three representative flight tasks that capture the core challenges of omnidirectional spatial awareness and collision avoidance.

Task 1: Hovering Maintenance. The UAV holds a target position (p_x, p_y, p_z) and yaw ψ_c toward a target. When obstacles approach, it must evade them and recover a stable hover at the desired pose.

Task 2: Dynamic Target Following. The UAV tracks a moving target at a predefined relative offset (e.g., 5 m in front) while keeping its yaw ψ_c toward the target. To isolate obstacle avoidance performance, the ground-truth target position is provided in this task.

Task 3: Fixed-Trajectory Filming. The UAV follows a prescribed trajectory while keeping its camera oriented toward the target and avoiding obstacles that appear along the path.

Together, these tasks assess omnidirectional obstacle avoidance under diverse flight conditions. For quantitative evaluation, the policy runs closed-loop in simulation, taking panoramic RGB images as input and directly outputting control actions. For qualitative evaluation, we feed real-world videos to the model and examine its avoidance responses, which also serves to assess sim-to-real transfer.

4.2 Simulation Experiments

Evaluation Protocol. Following Fly360 Zhang et al. (2026), we use two complementary metrics: Success Rate (SR), the fraction of collision-free episodes, and Collision Time (CT), the mean cumulative collision duration per episode, measured in seconds. Each episode lasts 120 s and continues after collisions, so CT accounts for all contact events throughout the rollout. For N episodes,

$$\mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[T_i^{\mathrm{coll}} = 0], \qquad \mathrm{CT} = \frac{1}{N}\sum_{i=1}^{N} T_i^{\mathrm{coll}}, \tag{10}$$

where T_i^{coll} is the total time spent in collision during episode i. SR measures collision-free completion, whereas CT captures cumulative contact duration, including in unsuccessful episodes.

Comparison & Ablation Study. We compare against existing methods along two design dimensions: observation coverage and intermediate representation. The depth-based group includes the forward-view controller trained with differentiable physics (DiffPhys) (Zhang et al., 2025a), the transformer-based ViT-Fly (Bhattacharya et al., 2025), multi-view depth controllers inspired by OmniNxt (Liu et al., 2024), and the panoramic depth pipeline Fly360 (Zhang et al., 2026). Following the multi-view adaptations in Fly360, we consider two OmniNxt variants: a four-fisheye

Table 2: Component ablation results. A: optical-flow representation; B: panoramic observations; C: spherical motion reasoning; D: joint perception–control optimization. ✓ indicates an enabled component and ✗ indicates its absence. ❄ denotes frozen perception during policy training, whereas 🔥 denotes joint perception–control optimization. SMR denotes spherical motion reasoning. "–" indicates not applicable and "×" denotes the method does not work.

Variant	Components				Hovering Maintenance		Dynamic Target Following		Fixed-Trajectory Filming	
	A	B	C	D	SR ↑	CT (s) ↓	SR ↑	CT (s) ↓	SR ↑	CT (s) ↓
RGB 🔥	✗	✓	✗	✓	×	×	×	×	×	×
Depth ❄	✗	✓	✗	✗	28/40	1.19	33/40	2.2	37/40	0.27
Depth 🔥	✗	✓	✗	✓	3/40	2.5	7/40	3.8	0/40	3.4
Forward-flow ❄	✓	✗	✗	✗	0/40	16.0	0/40	68.9	–	–
Forward-flow 🔥	✓	✗	✗	✓	0/40	15.6	0/40	73.8	–	–
Panoramic flow ❄	✓	✓	✗	✗	22/40	3.2	27/40	4.5	29/40	5.6
Panoramic flow 🔥	✓	✓	✗	✓	31/40	2.5	34/40	3.4	35/40	4.1
Panoramic flow + SMR ❄	✓	✓	✓	✗	21/40	3.3	28/40	4.2	29/40	5.7
Uni-Fly360 (ours)	✓	✓	✓	✓	**38/40**	**1.86**	**40/40**	**0**	**37/40**	**2.8**

configuration (OmniNxt-4) and a six-view configuration (OmniNxt-6) with 90° cameras facing forward, backward, left, right, up, and down. Both are our re-implementations rather than the original OmniNxt hardware system. The optical-flow group comprises Seeing Through Pixel Motion (Pixel Motion) (Hu et al., 2025) and Translational Optical Flow (Trans. Flow) (Deng et al., 2026), both using forward-view observations. Note that forward-view methods cannot decouple flight direction from camera orientation as Fixed-Trajectory Filming requires; we therefore exclude them from this task, here and in the ablation study below. Table 1 reports the main comparison results.

To assess the contribution of each design choice, we construct ablation variants along four axes (Table 2): optical flow representation (A, Sec. 3.3), panoramic observation (B, Sec. 3.4), spherical motion reasoning (C, Sec. 3.5), and joint perception–control optimization (D, Sec. 3.6). With panoramic observations, we compare RGB, depth, and optical flow as intermediate representations. Forward-view and panoramic flow variants quantify the benefit of omnidirectional coverage, while panoramic flow variants with and without spherical motion reasoning isolate the effect of explicit geometric reasoning. Finally, comparing frozen and jointly optimized perception for each representation isolates the effect of refining perception with navigation feedback.

Results and Discussion. Table 1 shows that Uni-Fly360 attains the highest or joint-highest SR on all three tasks: 95.0% on Hovering Maintenance, 100.0% on Dynamic Target Following, and 92.5% on Fixed-Trajectory Filming. Relative to Fly360, it improves SR by 25.0 and 17.5 percentage points on the first two tasks and matches its SR on filming. On target following, all 40 episodes are collision-free, reducing CT from 2.2 s to zero. However, on hovering and filming, CT remains higher than that of Fly360 (1.86 vs. 1.19 s and 2.80 vs. 0.27 s). Through visualization, this attributes to unstable optical-flow estimates after collisions, which should be improved in the future.

The ablations in Table 2 corroborate this observation. With frozen perception, extending the observation from a forward view to a panoramic view raises SR from zero to 55.0% and 67.5% on the two tasks, respectively. The benefit of joint optimization further depends on the intermediate representation. Learning directly from RGB fails entirely: the jointly trained RGB variant does not yield a working policy on any task, confirming that an explicit intermediate representation is indispensable. Among such representations, panoramic depth outperforms panoramic flow on both metrics across all three tasks with frozen perception, so the gains of the full framework cannot be attributed to optical flow alone. Joint optimization, however, raises the SR of panoramic flow by 22.5, 17.5, and 15.0 percentage points and consistently lowers CT, whereas under the same training setup it reduces the SR of the depth-based model from 70.0%, 82.5%, and 92.5% to 7.5%, 17.5%, and zero. These

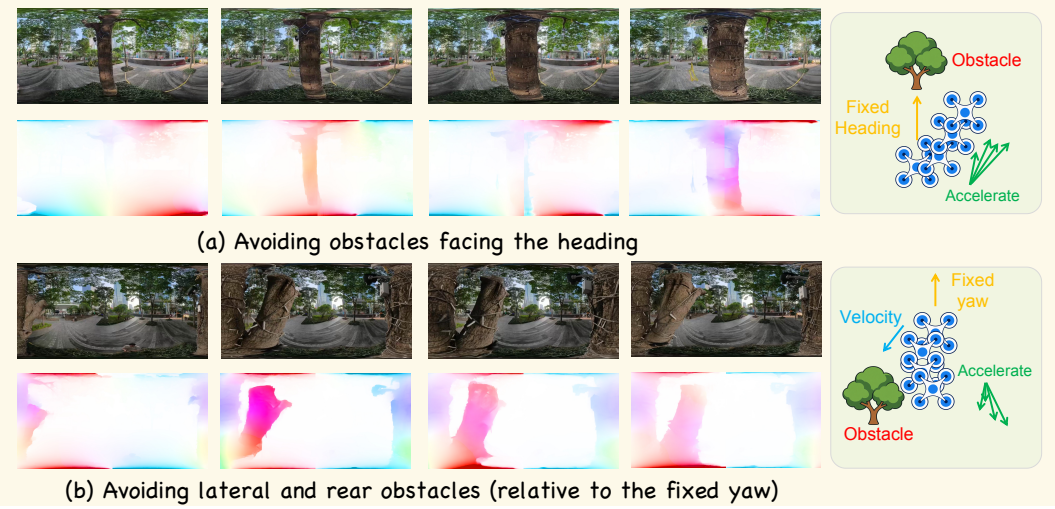

Figure 4: **Qualitative results on real-world panoramic videos.** We feed the recorded frames to the model and convert its output tensor into acceleration commands. Green arrows indicate the direction and relative magnitude of the acceleration. In each example, the model produces plausible optical flow and correct avoidance actions in real-world scenes, demonstrating effective sim-to-real transfer.

results support task-driven adaptation of panoramic flow and indicate that the effectiveness of joint perception–control optimization is representation-dependent.

Finally, SMR yields additional gains only in combination with joint optimization, raising SR from 77.5%, 85.0%, and 87.5% to 95.0%, 100.0%, and 92.5% and reducing CT from 2.50 to 1.86 s, 3.40 to zero, and 4.10 to 2.80 s. With frozen perception, adding SMR changes the number of successful episodes by at most one per task and yields no consistent reduction in CT. This contrast suggests that the geometric cues from SMR are most useful when perception can adapt through joint training. Taken together, the ablations confirm the complementary roles of panoramic coverage, task-driven optical-flow adaptation, and differentiable spherical motion reasoning in the full framework.

4.3 Real-World Qualitative Analysis

To assess sim-to-real transfer, we test Uni-Fly360 on real-world videos captured by a panoramic camera (Fig. 4). During recording, the camera orientation is kept fixed while the camera is moved in various directions, so that obstacles may approach from any direction. We feed the recorded frames to the model and convert its output tensor into acceleration commands. As obstacles approach, the model produces correct and timely avoidance actions, with the acceleration consistently directed away from the approaching obstacle. Despite being trained entirely in simulation on non-photorealistic imagery, the model transfers to real-world panoramic footage without any fine-tuning, demonstrating strong sim-to-real transfer capability.

5 Conclusion

In this work, we presented Uni-Fly360, an end-to-end framework for omnidirectional UAV obstacle avoidance from panoramic image sequences. It adopts optical flow as an intermediate representation, jointly optimizes the flow estimator with a recurrent policy in a differentiable simulator, and extracts global ego-motion structure and local motion residuals through differentiable spherical motion reasoning. Uni-Fly360 achieves the highest or joint-highest success rates on three panoramic tasks, with ablations confirming the complementary roles of panoramic coverage, task-driven flow adaptation, and spherical motion reasoning, while transfers to real panoramic videos without fine-tuning. Improving flow robustness after collisions remains future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript drafting and language refinement, and the preparation of illustrative icons. The authors reviewed and revised the AI-assisted text to preserve the intended scientific meaning and ensure consistency with the methods and results presented in the paper. AI-assisted visual materials were checked for accurate representation and references identified with AI assistance were verified against their original sources. We take full responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the assistance of generative AI.

REFERENCES

Charles-Olivier Artizzu, Guillaume Allibert, and Cédric Demonceaux. OMNI-DRL: Learning to fly in forests with omnidirectional images. *IFAC-PapersOnLine*, 55(38):120–125, 2022. doi: 10.1016/j.ifacol.2023.01.143. 3

Keshav Bhandari, Bin Duan, Gaowen Liu, Hugo Latapie, Ziliang Zong, and Yan Yan. Learning omnidirectional flow in 360° video via siamese representation. In *Computer Vision – ECCV 2022*, pp. 557–574. Springer, 2022. doi: 10.1007/978-3-031-20074-8_32. 3

Anish Bhattacharya, Nishanth Rao, Dhruv Parikh, Pratik Kunapuli, Yuwei Wu, Yuezhan Tao, Nikolai Matni, and Vijay Kumar. Vision transformers for end-to-end vision-based quadrotor obstacle avoidance. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2025. doi: 10.1109/ICRA55743.2025.11128042. 3, 7

Rik J. Bouwmeester, Federico Paredes-Vallés, and Guido C. H. E. de Croon. NanoFlowNet: Real-time dense optical flow on a nano quadcopter. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1996–2003, 2023. doi: 10.1109/ICRA48891.2023.10161258. 3

Yusuf Umut Ciftci, Darren Chiu, Zeyuan Feng, Gaurav S. Sukhatme, and Somil Bansal. SAFE-GIL: SAFEty Guided Imitation Learning for Robotic Systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3559–3566, 2025. doi: 10.1109/ICRA55743.2025.11128298. 3

G. C. H. E. de Croon, C. De Wagter, and T. Seidl. Enhancing optical-flow-based control by learning visual appearance cues for flying robots. *Nature Machine Intelligence*, 3(1):33–41, 2021. doi: 10.1038/s42256-020-00279-7. 3

Jiaxi Deng, Yushen Wang, Haitao Meng, Zuoxun Hou, Yi Chang, and Gang Chen. OmniStereo: Real-time omnidireactional depth estimation with multiview fisheye cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1003–1012, 2025. doi: 10.1109/CVPR52734.2025.00102. 3

Yang Deng, Yu Hu, Feng Yu, Linzuo Zhang, and Danping Zou. Autonomous FPV flight with translational optical flow and uncertainty mask. *IEEE Robotics and Automation Letters*, 11(9): 10433–10440, 2026. doi: 10.1109/LRA.2026.3706940. 3, 8

Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015. doi: 10.1109/ICCV.2015.316. 4

Shiladitya Dutta, Aayush Gupta, Varun Saran, and Avideh Zakhor. Vision-Guided Outdoor Flight and Obstacle Evasion via Reinforcement Learning. *IEEE Robotics and Automation Letters*, 11 (2):1202–1209, 2026. doi: 10.1109/LRA.2025.3641120. 3

Dhiraj Gandhi, Lerrel Pinto, and Abhinav Gupta. Learning to fly by crashing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3948–3955, 2017. doi: 10.1109/iros.2017.8206247. 2

Ismail Geles, Leonard Bauersfeld, Angel Romero, Jiaxu Xing, and Davide Scaramuzza. Demonstrating Agile Flight from Pixels without State Estimation. In *Robotics: Science and Systems XX*, 2024. doi: 10.15607/RSS.2024.XX.082. 3

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01712. 3

Yu Hu, Yuang Zhang, Yunlong Song, Yang Deng, Feng Yu, Linzuo Zhang, Weiyao Lin, Danping Zou, and Wenxian Yu. Seeing through pixel motion: Learning obstacle avoidance from optical flow with one camera. *IEEE Robotics and Automation Letters*, 10(6):5871–5878, 2025. doi: 10.1109/LRA.2025.3560842. 3, 8

Minwoo Kim, Jongyun Kim, Minjae Jung, and Hyondong Oh. Towards monocular vision-based autonomous flight through deep reinforcement learning. *Expert Systems with Applications*, 198: 116742, 2022. doi: 10.1016/j.eswa.2022.116742. 3

Minwoo Kim, Geunsik Bae, Jinwoo Lee, Woojae Shin, Changseung Kim, Myongyol Choi, Heejung Shin, and Hyongdong Oh. RAPID: Robust and Agile Planner Using Inverse Reinforcement Learning for Vision-Based Drone Navigation. In *Robotics: Science and Systems XXI*, 2025. doi: 10.15607/RSS.2025.XXI.142. 3

Fanxing Li, Fangyu Sun, Tianbao Zhang, and Danping Zou. VisFly: An Efficient and Versatile Simulator for Training Vision-Based Flight. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11325–11332, 2025. doi: 10.1109/icra55743.2025.11128458. 3

Ming Li, Xueqian Jin, Xuejiao Hu, Jingzhao Dai, Sidan Du, and Yang Li. MODE: Multi-view Omnidirectional Depth Estimation with 360° Cameras. In *Computer Vision – ECCV 2022*, pp. 197–213, 2022a. doi: 10.1007/978-3-031-19827-4_12. 3

Yiheng Li, Connelly Barnes, Kun Huang, and Fang-Lue Zhang. Deep 360° Optical Flow Estimation Based on Multi-projection Fusion. In *Computer Vision – ECCV 2022*, pp. 336–352, 2022b. doi: 10.1007/978-3-031-19833-5_20. 3

John Lim and Nick Barnes. Directions of egomotion from antipodal points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. doi: 10.1109/CVPR.2008.4587497. 2

Longliang Liu, Miaojie Feng, Junda Cheng, Jijun Xiang, Xuan Zhu, and Xin Yang. PriOr-Flow: Enhancing primitive panoramic optical flow with orthogonal view. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5326–5336, 2025. doi: 10.1109/ICCV51701.2025.00506. URL https://openaccess.thecvf.com/content/ICCV2025/papers/Liu_PriOr-Flow_Enhancing_Primitive_Panoramic_Optical_Flow_with_Orthogonal_View_ICCV_2025_paper.pdf. 3

Peize Liu, Chen Feng, Yang Xu, Yan Ning, Hao Xu, and Shaojie Shen. OmniNxt: A fully open-source and compact aerial robot with omnidirectional visual perception. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10605–10612, 2024. doi: 10.1109/IROS58592.2024.10802134. 2, 3, 7

Antonio Loquercio, Ana I. Maqueda, Carlos R. del Blanco, and Davide Scaramuzza. DroNet: Learning to fly by driving. *IEEE Robotics and Automation Letters*, 3(2):1088–1095, 2018. doi: 10.1109/LRA.2018.2795643. 2

Antonio Loquercio, Elia Kaufmann, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Deep drone racing: From simulation to reality with domain randomization. *IEEE Transactions on Robotics*, 36(1):1–14, 2020. doi: 10.1109/TRO.2019.2942989. 3

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. doi: 10.1126/scirobotics.abg5810. 2, 3

Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4040–4048, 2016. doi: 10.1109/CVPR.2016.438. 4

Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, 126(9):942–960, 2018. doi: 10.1007/S11263-018-1082-6. 4

Nico Messikommer, Jiaxu Xing, Elie Aljalbout, and Davide Scaramuzza. Student-Informed Teacher Training. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a8223b0ad64007423ffb308b0dd92298-Abstract-Conference.html. 3

Andreas Meuleman, Hyeonjoong Jang, Daniel S. Jeon, and Min H. Kim. Real-Time Sphere Sweeping Stereo from Multiview Fisheye Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11423–11432, 2021. doi: 10.1109/cvpr46437.2021.01126. URL https://openaccess.thecvf.com/content/CVPR2021/html/Meuleman_Real-Time_Sphere_Sweeping_Stereo_From_Multiview_Fisheye_Images_CVPR_2021_paper.html. 3

Yunfan Ren, Fangcheng Zhu, Guozheng Lu, Yixi Cai, Longji Yin, Fanze Kong, Jiarong Lin, Nan Chen, and Fu Zhang. Safety-assured high-speed navigation for MAVs. *Science Robotics*, 10(98): eado6187, 2025. doi: 10.1126/scirobotics.ado6187. 2

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html. 3

Stephane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J. Andrew Bagnell, and Martial Hebert. Learning monocular reactive UAV control in cluttered natural environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1765–1772, 2013. doi: 10.1109/ICRA.2013.6630809. 3

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real single-image flight without a single real image. In *Robotics: Science and Systems XIII*, 2017. doi: 10.15607/RSS.2017.XIII.034. 3

Daniel Schleich and Sven Behnke. Predictive Angular Potential Field-based Obstacle Avoidance for Dynamic UAV Flights. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13618–13625, 2022. doi: 10.1109/iros47612.2022.9981677. 2

Hao Shi, Yifan Zhou, Kailun Yang, Xiaoting Yin, Ze Wang, Yaozu Ye, Zhe Yin, Shi Meng, Peng Li, and Kaiwei Wang. PanoFlow: Learning 360° optical flow for surrounding temporal understanding. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5570–5585, 2023. doi: 10.1109/TITS.2023.3241212. 3, 5

Yunlong Song, Selim Naji, Elia Kaufmann, Antonio Loquercio, and Davide Scaramuzza. Flightmare: A Flexible Quadrotor Simulator. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1147–1157. PMLR, 2021. URL https://proceedings.mlr.press/v155/song21a.html. 3

Yunlong Song, Kexin Shi, Robert Penicka, and Davide Scaramuzza. Learning Perception-Aware Agile Flight in Cluttered Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1989–1995, 2023. doi: 10.1109/ICRA48891.2023.10160563. 3

Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T. Freeman, and Ce Liu. AutoFlow: Learning a better training set for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10093–10102, 2021. doi: 10.1109/CVPR46437.2021.00996.

URL https://openaccess.thecvf.com/content/CVPR2021/html/Sun_AutoFlow_Learning_a_Better_Training_Set_for_Optical_Flow_CVPR_2021_paper.html. 2, 4

Sarah Y. Tang and Vijay Kumar. Autonomous flight. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:29–52, 2018. doi: 10.1146/annurev-control-060117-105149. 2

Jesus Tordesillas, Brett T. Lopez, Michael Everett, and Jonathan P. How. FASTER: Fast and Safe Trajectory Planner for Navigation in Unknown Environments. *IEEE Transactions on Robotics*, 38(2):922–938, 2022. doi: 10.1109/tro.2021.3100142. 3

Changhee Won, Jongbin Ryu, and Jongwoo Lim. OmniMVS: End-to-End Learning for Omnidirectional Stereo Matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8987–8996, 2019. doi: 10.1109/iccv.2019.00908. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Won_OmniMVS_End-to-End_Learning_for_Omnidirectional_Stereo_Matching_ICCV_2019_paper.html. 3

Tianyue Wu, Yeke Chen, Tianyang Chen, Guangyu Zhao, and Fei Gao. Whole-Body Control Through Narrow Gaps from Pixels to Action. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11317–11324, 2025. doi: 10.1109/ICRA55743.2025.11128088. 3

Feng Xiao, Peter Zheng, Julien di Tria, Basaran Bahadir Kocer, and Mirko Kovac. Optic flow-based reactive collision prevention for MAVs using the fictitious obstacle hypothesis. *IEEE Robotics and Automation Letters*, 6(2):3144–3151, 2021. doi: 10.1109/LRA.2021.3062317. 3

Jiaxu Xing, Angel Romero, Leonard Bauersfeld, and Davide Scaramuzza. Bootstrapping reinforcement learning with imitation for vision-based agile flight. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2542–2556. PMLR, 2025. URL https://proceedings.mlr.press/v270/xing25a.html. 3

Zhefan Xu, Xinming Han, Haoyu Shen, Hanyu Jin, and Kenji Shimada. NavRL: Learning Safe Flight in Dynamic Environments. *IEEE Robotics and Automation Letters*, 10(4):3668–3675, 2025. doi: 10.1109/LRA.2025.3546069. 3

Hang Yu, Christophe De Wagter, and Guido C. H. E. de Croon. Depth Transfer: Learning to See Like a Simulator for Real-World Drone Navigation. *IEEE Robotics and Automation Letters*, 10 (11):11848–11855, 2025a. doi: 10.1109/LRA.2025.3617729. 3

Hang Yu, Christophe De Wagter, and Guido C. H. E. de Croon. MAVRL: Learn to fly in cluttered environments with varying speed. *IEEE Robotics and Automation Letters*, 10(2):1441–1448, 2025b. doi: 10.1109/LRA.2024.3522778. 3

Mingze Yuan and Christian Richardt. 360° Optical Flow using Tangent Images. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2021. URL https://richardt.name/publications/360-flow/. 3

Xiangkai Zhang, Dizhe Zhang, WenZhuo Cao, Zhaoliang Wan, Yingjie Niu, Lu Qi, Xu Yang, and Zhiyong Liu. Fly360: Omnidirectional obstacle avoidance within drone view, 2026. URL https://arxiv.org/abs/2603.06573. arXiv preprint arXiv:2603.06573. 2, 3, 7

Yuang Zhang, Yu Hu, Yunlong Song, Danping Zou, and Weiyao Lin. Learning vision-based agile flight via differentiable physics. *Nature Machine Intelligence*, 7(6):954–966, 2025a. doi: 10.1038/s42256-025-01048-0. 2, 3, 4, 7

Yuhang Zhang, Jiaping Xiao, and Mir Feroskhan. Learning Cross-Modal Visuomotor Policies for Autonomous Drone Navigation. *IEEE Robotics and Automation Letters*, 10(6):5425–5432, 2025b. doi: 10.1109/LRA.2025.3559824. 3

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. doi: 10.1109/LRA.2019.2927938. 3

Boyu Zhou, Jie Pan, Fei Gao, and Shaojie Shen. RAPTOR: Robust and Perception-Aware Trajectory Replanning for Quadrotor Fast Flight. *IEEE Transactions on Robotics*, 37(6):1992–2009, 2021a. doi: 10.1109/tro.2021.3071527. 3

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, 2021b. doi: 10.1109/LRA.2020.3047728. 2, 3

APPENDIX

A ADDITIONAL DETAILS OF PANORAMIC MOTION REPRESENTATION

A.1 PANORAMIC OPTICAL FLOW ESTIMATION AND NORMALIZATION

Estimator configuration. The feature and context encoders of PanoRAFT incorporate deformable convolutions to accommodate ERP distortions. A recurrent update operator iteratively refines flow estimates using a four-level all-pairs correlation pyramid with a lookup radius of four. Given (I_{t-1}, I_t), the estimator produces

$$\Delta \boldsymbol{z}_t^{(j)} = \left(\Delta u_t^{(j)}, \Delta v_t^{(j)}\right), \qquad j = 1, \ldots, J, \tag{11}$$

where J is the number of refinement iterations. We use fewer iterations during training to reduce memory consumption and more iterations at inference for further refinement. The iteration index is omitted in the main text for clarity.

Angular conversion and bounded features. Pixel-flow magnitudes depend on image resolution: the same angular displacement corresponds to different pixel displacements at different ERP resolutions. We therefore apply Eq. (3) to each refinement output, with angular coordinates oriented consistently with the corresponding image axes. To obtain bounded policy features, we apply componentwise normalization:

$$\widehat{F}_t^{(j)} = \tanh\left(\frac{[\Delta\theta_t^{(j)}, \Delta\phi_t^{(j)}]}{f_c}\right), \tag{12}$$

where $f_c > 0$ is an angular scale controlling saturation. Each feature component is smoothly bounded within $(-1, 1)$. Spherical motion reasoning uses the angular displacements before this normalization. At the beginning of each rollout, when no preceding frame is available, we initialize the flow field to zero.

A.2 GEOMETRIC INTERPRETATION AND RESIDUAL PROPERTIES

Antipodal constraints. Under the first-order static-scene motion model, rotational flow components at antipodal viewing directions have opposite signs. Their average therefore suppresses rotation. The remaining flow $\boldsymbol{g}_i$ is collinear with the projection of camera translation onto the tangent plane at $\boldsymbol{s}_i$. For nonzero $\boldsymbol{g}_i$, the translation vector consequently lies in the plane spanned by $\boldsymbol{s}_i$ and $\boldsymbol{g}_i$, whose normal is $\boldsymbol{c}_i = \boldsymbol{s}_i \times \boldsymbol{g}_i$. Different viewing directions provide complementary plane constraints on the same translation axis. This interpretation is a first-order approximation rather than an exact cancellation guarantee for arbitrary finite inter-frame motion.

Global constraint aggregation. For a unit candidate direction $\boldsymbol{a}$, the quadratic form

$$\boldsymbol{a}^\top M \boldsymbol{a} = \frac{\sum_i w_i (\boldsymbol{c}_i^\top \boldsymbol{a})^2}{\sum_i w_i + \epsilon} \tag{13}$$

measures the weighted squared violation of the local constraints. Its minimizer is a minimum-eigenvalue eigenvector of M. When this direction is uniquely determined and τ is sufficiently small, the soft descriptor in Eq. (6) satisfies

$$Q_{\text{ego}} \approx \widehat{\boldsymbol{v}}\widehat{\boldsymbol{v}}^\top, \tag{14}$$

where $\widehat{\boldsymbol{v}}$ is the estimated unit translation axis. This representation does not distinguish between $\widehat{\boldsymbol{v}}$ and $-\widehat{\boldsymbol{v}}$, avoiding the sign ambiguity of an eigenvector representation.

Interpretation of the local residual. For a well-defined translation axis $\widehat{\boldsymbol{v}}$, its tangential projection at viewing direction $\boldsymbol{s}_i$ is

$$\boldsymbol{p}_i = (I - \boldsymbol{s}_i \boldsymbol{s}_i^\top)\widehat{\boldsymbol{v}}. \tag{15}$$

The corresponding unit direction is $\widehat{\boldsymbol{p}}_i$ wherever this projection is nonzero. The squared inner product in Eq. (7) measures directional agreement independently of the sign of the estimated axis. The weight w_i^{loc} suppresses weakly observable regions, particularly near the focus of expansion and contraction, where the predicted translational flow approaches zero. The constant ϵ provides numerical stability.

A small residual indicates agreement with the estimated translation model, whereas a large residual indicates a local directional mismatch. Because the residual compares the original spherical flow $\boldsymbol{f}_i$ with a translation-only prediction, independently moving objects, camera rotation, and flow-estimation errors can all contribute to its response. Accordingly, D_{dyn} provides motion-inconsistency cues for the policy rather than a definitive dynamic-object segmentation.

A.3 Training Objectives

For each rollout, we sample an obstacle configuration and an initial UAV state, then generate a closed-loop trajectory by repeatedly executing the current policy. The training objective combines trajectory-level navigation losses with auxiliary supervision from simulator-provided optical flow and camera translation.

Navigation objective. The navigation loss is

$$\mathcal{L}_{\mathrm{nav}} = \lambda_v \mathcal{L}_v + \lambda_c \mathcal{L}_c + \lambda_a \mathcal{L}_a + \lambda_j \mathcal{L}_j, \tag{16}$$

where $\mathcal{L}_v$ penalizes deviations from the reference velocity, $\mathcal{L}_c$ penalizes insufficient obstacle clearance, and $\mathcal{L}_a$ and $\mathcal{L}_j$ regularize acceleration magnitude and jerk, respectively. These terms are evaluated along the simulated trajectory. The weights λ_v, λ_c, λ_a, and λ_j control their relative contributions, balancing reference tracking, collision avoidance, and control smoothness.

Auxiliary motion supervision. To retain the physical meaning of the learned motion representation, we introduce

$$\mathcal{L}_{\mathrm{aux}} = \lambda_f \mathcal{L}_{\mathrm{flow}} + \lambda_{\mathrm{ego}} \mathcal{L}_{\mathrm{ego}}, \tag{17}$$

where λ_f and λ_{ego} control the strengths of optical-flow and translation-axis supervision.

The optical-flow loss $\mathcal{L}_{\mathrm{flow}}$ measures the discrepancy between predicted and ground-truth flow before the bounded normalization in Eq. (12). Applying supervision before this normalization constrains the motion estimates directly, rather than their saturated policy features.

To supervise the sign-invariant translation-axis descriptor, we use

$$\mathcal{L}_{\mathrm{ego}} = \frac{1}{|\mathcal{T}_{\mathrm{valid}}|} \sum_{t \in \mathcal{T}_{\mathrm{valid}}} \left\| Q_{\mathrm{ego},t} - \boldsymbol{d}_t^{\star} (\boldsymbol{d}_t^{\star})^{\top} \right\|_F^2, \tag{18}$$

where $\boldsymbol{d}_t^{\star}$ is the ground-truth unit camera-translation direction expressed in the same camera frame as the spherical geometry module, and $\|\cdot\|_F$ denotes the Frobenius norm. The outer-product target is invariant to the sign of $\boldsymbol{d}_t^{\star}$, matching the axis representation used by $Q_{\mathrm{ego},t}$. The set $\mathcal{T}_{\mathrm{valid}}$ includes only steps with sufficient translational motion to define a reliable unit direction.

The navigation and auxiliary losses are combined as in Eq. (9) and optimized jointly over θ and ψ. The auxiliary terms encourage accurate image-motion estimation and consistent global translation geometry while navigation supervision adapts these representations to downstream avoidance requirements.