sensors

MDPI

Article

Learning Active Depth Sensing for UAV Navigation: Reliability Supervision and Staged Adaptation

Weiqi Gai, Guodong Zhao, Jiachi Ji and Zhijun Meng *

School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China
* Correspondence: mengzhijun@buaa.edu.cn

Abstract

Adaptive depth acquisition changes the observations used for UAV navigation. We present a staged framework that connects differentiable depth-reliability supervision to flight performance. An active-stereo surrogate supplies power, exposure, and gain targets for the least reliable image patches. Student-rollout relabeling extends supervision to learned-acquisition states, and independent visual stems preserve the camera mapping during flight adaptation. In the primary simulated aperture-crossing benchmark (60-step cap), success reaches 75.1% versus 66.7% for nominal fixed acquisition, while valid-depth fill rises from 72.1% to 97.5%. Under a separate ablation protocol (5 s timeout; 120-step cap), removing relabeling and retraining lowers success from 77.89% to 70.78%; sharing the visual stem lowers it to 74.30%. The frozen policy retains 75.33%–79.67% success across four partial-observation layouts. A bounded-depth-bias test yields 70.00% success at 0.10 m and 37.00% at 0.30 m, compared with its 74.67% zero-bias control. Supplementary D455 captures show a coverage gain after camera adaptation, and a separate PX4 hardware-in-the-loop test completes 44/60 crossings with synthetic depth. The results demonstrate navigation gains from the complete differentiable-sensing and flight-adaptation framework while quantifying its sensitivity to incorrect ranges.

Keywords: active depth sensing; UAV navigation; differentiable sensing; depth reliability; camera control; flight adaptation

1. Introduction

Autonomous unmanned aerial vehicles (UAVs) need local geometry to avoid obstacles and traverse confined spaces. Compact depth cameras support these tasks, but measurement availability depends on acquisition conditions. For aperture crossing, the relevant sensing requirement is local: depth must reveal the boundary needed for the next motion decision, even when most background pixels remain valid.

Active stereo makes this requirement controllable: projector power, exposure, and gain affect correspondence through material, illumination, range, and motion [1,2]. Perception-aware flight changes trajectories or speed to preserve observations [3–5]; acquisition control changes the measurement at a given pose [6–8]. We study the latter interface for depth-based aperture crossing.

Learning acquisition for navigation involves two distribution shifts. Camera decisions change the observations received by the camera controller itself, and flight must learn to use the resulting depth stream. Updating a shared encoder during flight adaptation can additionally change the camera mapping. Figure 1 addresses these dependencies

Check for updates

Academic Editor: Shaoming He

Received: 14 September 2026
Revised: 15 October 2026
Accepted: 22 October 2026
Published: 24 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

by teaching local depth reliability, querying the teacher on student-induced states, and adapting flight while freezing an independent camera branch.

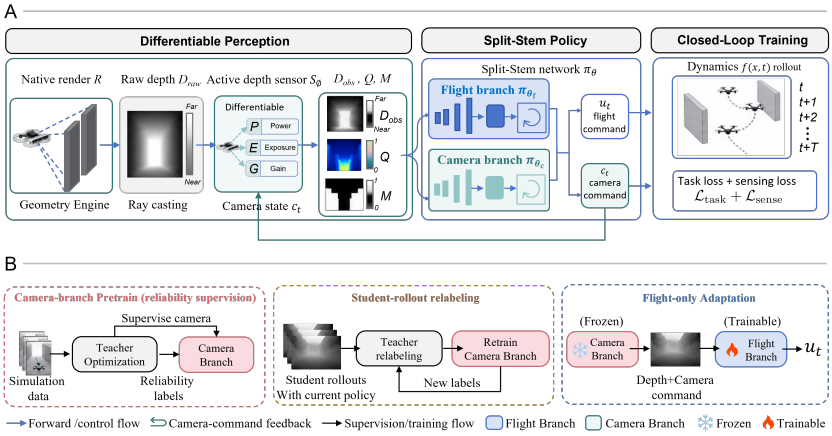


Figure 1. Staged sensing and flight learning. (**A**) Acquisition-dependent depth feeds independent camera and flight branches. (**B**) Reliability teaching and camera fitting are followed by student-state relabeling/retraining and flight adaptation with the complete camera mapping frozen. Quality maps are sensor auxiliaries; only observed depth enters the policy. Blue and teal identify the flight and camera branches, respectively. Blue solid arrows indicate forward data and control flow, and the teal feedback arrow indicates camera-command feedback for subsequent acquisition. Blue dashed rays illustrate geometric depth rendering. Arrows in panel (**B**) indicate supervision and training flow; colored dashed boxes distinguish training stages. Snowflake and flame symbols denote frozen and trainable parameters, respectively.

The contribution is a learning interface that makes differentiable acquisition useful for downstream control. Its three components provide reliability-based register supervision, camera learning on student-induced states, and flight adaptation with a preserved camera mapping. The primary navigation comparisons establish the complete-system gain; separate ablations test the latter two design choices. Added occlusion and depth-bias tests examine the frozen policy's operating range, while physical D455 acquisition and PX4 simulation-in-hardware (SIH) test the sensing and flight-controller interfaces separately.

2. Related Work

We organize prior work by sensor response, navigation, and learning under adaptive observations: the three interfaces connected by the proposed method.

2.1. Depth Measurement and Adaptive Camera Acquisition

RealSense metrology [1] and studies of pattern projection and gain in active stereo [2] motivate our response factors. Exposure optimization for visual odometry links a photometric proxy to downstream localization [6]. We similarly distinguish the sensing objective from task performance, but control projector power, exposure, and gain to regulate depth availability during recurrent flight.

2.2. Depth-Based and Perception-Aware UAV Navigation

Depth-based systems combine route following and depth thresholds [9], actuated sensing for landing and avoidance [10], or directional constraints under depth dropout [11]. Active perception [12–14] also motivates viewpoint, trajectory, and speed control [3–5,15,16]. Agile-flight systems further demonstrate tight perception–action coupling [17–19]. Register adaptation complements these approaches by changing the depth acquired at a given pose.

2.3. Differentiable Sensing and Learning with Adaptive Observations

Differentiable simulation supports gradient-based flight learning [20–23] and differentiable model-predictive control [24]. Embodied simulation [25–28], image-system optimization, and sensor transfer [29–33] connect observation formation to learning. Neural Camera Simulators and DiffPhysCam expose controllable image transformations and camera response [34,35]; our surrogate follows their modular separation of geometry and acquisition.

TaCOS and JOCA optimize camera design or adaptive camera control jointly with perception [7,8]. We instead keep hardware fixed and connect local depth-reliability teaching to recurrent flight adaptation. DAgger supplies the learner-state querying principle [36]: here the teacher labels acquisition registers while the student retains flight control. The ablation tests this relabeling together with camera retraining, rather than proposing a new aggregation algorithm.

3. Materials and Methods

The teacher is a finite-step optimizer that supplies camera-register targets through a differentiable sensing surrogate; the student is the recurrent camera controller. Training proceeds from an initial teacher collection to camera pretraining, student-rollout relabeling and camera retraining, then flight adaptation with the entire camera mapping frozen. Algorithm 1 summarizes the sequence.

Algorithm 1 Reliability Teaching, Camera Learning, and Flight Adaptation

Require: fixed flight policy, sensing regimes, teacher objective (21)
1: Collect $\mathcal{D}_0$ under the initial acquisition policy.
2: At each collected state, optimize a camera target with geometry and motion fixed.
3: Fit only Θ_{cam} on $\mathcal{D}_0$ using (24).
4: Roll out the camera-pretrained student with its own flight and acquisition actions.
5: Query the same teacher at visited states to form $\mathcal{D}_{\text{dag}}$.
6: Retrain Θ_{cam} on $\mathcal{D}_{\text{dag}}$.
7: Freeze the complete camera mapping and detach its recurrent path.
8: Adapt $\Theta_f = \Theta \setminus \Theta_{\text{cam}}$ under learned acquisition using (27).
9: **return** final split-stem policy

3.1. Problem Formulation and Closed-Loop Acquisition

Let $\mathbf{x}_t$ contain pose, velocity, attitude, and collision margin, and let $\mathcal{E}$ specify the scene geometry and sensing regime ρ_t. Scene denotes the spatial layout; regime denotes its glare, dark-material, or specular acquisition response. Normalized projector power, exposure, and gain form the camera state

$$\mathbf{c}_t = [P_t, E_t, G_t]^\top \in [0,1]^3, \tag{1}$$

The closed loop is

$$D_t^{\text{raw}} = \mathcal{R}(\mathbf{x}_t, \mathcal{E}), \tag{2}$$
$$(D_t^{\text{obs}}, Q_t^{\text{obs}}) = \mathcal{S}_\phi(D_t^{\text{raw}}, \mathbf{c}_t, \rho_t, \mathbf{x}_t), \tag{3}$$
$$\mathbf{s}_t = \psi_{\text{flight}}(\mathbf{x}_t, \mathbf{x}_{\text{goal}}, \mathbf{c}_t), \tag{4}$$
$$\mathbf{m}_t = \psi_{\text{cam}}(\mathbf{x}_t, \mathbf{c}_t), \tag{5}$$
$$(\mathbf{u}_t, \mathbf{c}_{t+1}^\star, \mathbf{h}_{t+1}, \mathbf{q}_{t+1}) = \Pi_\theta(D_t^{\text{obs}}, \mathbf{s}_t, \mathbf{c}_t, \mathbf{m}_t, \mathbf{h}_t, \mathbf{q}_t), \tag{6}$$
$$\mathbf{c}_{t+1} = \alpha \mathbf{c}_t + (1-\alpha)\mathbf{c}_{t+1}^\star, \quad \alpha = 0.7, \tag{7}$$
$$\mathbf{x}_{t+1} = f_{\text{dyn}}(\mathbf{x}_t, \mathbf{a}_{t-2}, \Delta t_t). \tag{8}$$

The renderer $\mathcal{R}$ produces ideal depth and $\mathcal{S}_\phi$ produces observed depth and auxiliary quality. Only observed depth enters the policy. Flight inputs $\mathbf{s}_t$ contain velocity, goal-directed reference, attitude, safety margin, and current registers rescaled to $[-1, 1]$; camera-state feedback is included in all reported experiments. Camera motion input $\mathbf{m}_t$ excludes the goal vector. Hidden states $\mathbf{h}_t$ and $\mathbf{q}_t$ serve the two branches. The acceleration output $\mathbf{u}_t$ is clipped to $\pm a_{\max}$, rotated to the world frame to obtain $\mathbf{a}_t$, and applied after two control steps. Primary and ablation runs use simulator state; SIH uses PX4 estimated state (Section 4.5).

$$\Delta t_t = \Delta t_{\text{base},t} + \kappa_\tau \tau(\langle E_t \rangle_{\text{batch}}), \tag{9}$$

Here $\kappa_\tau = 0.01$ and the base interval has mean $1/15$ s and standard deviation $0.1/15$ s. The batch-mean exposure term is detached from gradients to share one integration interval across vectorized rollouts. With $\tau(E) = 0.25 + 2.75E$, exposure adds 2.5–30 ms: the nominal total is 69.17–96.67 ms before jitter, and the maximum individual-versus-mean discrepancy is 27.5 ms. Evaluation uses batch size one, making batch-mean and per-state timing identical. The training integrator is held fixed across comparisons. These parameters define the acquisition timing of the simulation model.

3.2. Differentiable Active-Stereo Response Model

The teacher compares acquisition settings at a fixed state. Following modular camera simulation [35], geometric rendering supplies ideal depth $Z_t = D_t^{\text{raw}}$ and the acquisition response predicts its availability. Figure 2 groups projected return, washout, blur, noise, and edge difficulty; the surrogate models dropout rather than photometric corruption or biased retained ranges.

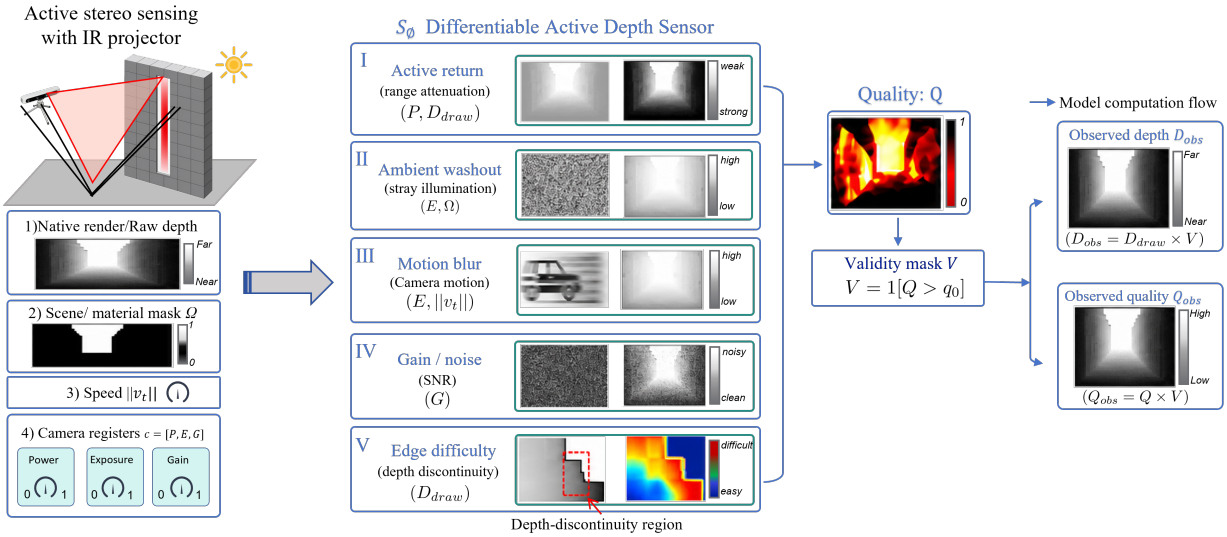


Figure 2. Conceptual active-stereo response model. Rendered depth, localized sensing hazards, motion, and power/exposure/gain settings determine acquisition quality Q through five illustrated response factors: (I) active return, (II) ambient washout, (III) motion blur, (IV) gain-dependent noise, and (V) depth-edge difficulty. Hard validity V gates depth and quality to produce D^{obs} and the auxiliary Q^{obs}; only observed depth enters the policy. The image examples illustrate response mechanisms, rather than intermediate images generated by the surrogate. Blue arrows indicate model computation flow, and the red dashed box highlights a depth-discontinuity region. Colored maps schematically illustrate spatial variation in sensing response and quality. Time and pixel indices are omitted.

For camera position $\mathbf{p}_t^{\mathcal{E}}$ and orientation columns $(\mathbf{r}_{x,t}, \mathbf{r}_{y,t}, \mathbf{r}_{z,t})$, the pixel ray is

$$\begin{aligned} \eta_i &= \left(2\frac{i+1/2}{H} - 1\right) f_y - \epsilon_r, \quad f_y = f_x \frac{H}{W}, \\ \xi_j &= \left(2\frac{j+1/2}{W} - 1\right) f_x - \epsilon_r, \\ \mathbf{d}_{tij} &= \mathbf{r}_{x,t} - \eta_i \mathbf{r}_{z,t} - \xi_j \mathbf{r}_{y,t}, \end{aligned} \tag{10}$$

Here $f_x = \tan(\mathrm{FOV}_x/2)$ and ϵ_r is a numerical offset. Ideal depth is the smallest positive ray-intersection parameter. Clipping to $\bar{Z} = \mathrm{clip}(Z, z_{\min}, z_{\max})$ gives the neighborhood discontinuity score

$$\begin{aligned} Z_{tij}^{+} &= \max_{(m,n)\in\mathcal{N}_{ij}} \bar{Z}_{tmn}, \quad Z_{tij}^{-} = \min_{(m,n)\in\mathcal{N}_{ij}} \bar{Z}_{tmn}, \\ \delta_{tij} &= \mathrm{clip}\left(\frac{Z_{tij}^{+} - Z_{tij}^{-}}{\bar{Z}_{tij} + \epsilon_z}, 0, 1\right). \end{aligned} \tag{11}$$

The 3×3 neighborhood $\mathcal{N}_{ij}$ approximates correspondence difficulty near depth boundaries. A projected hazard mask $\Omega_{tij} = m_{\rho_t}(\bar{Z}_{tij}, \mathbf{x}_t, \mathcal{E}) \in [0,1]$ localizes illumination or material degradation. Geometry, pose, and this mask are held fixed when differentiating camera registers. For clipped commands $p_t, e_t, g_t \in [0,1]$, the exposure and gain mappings are

$$\tau(E_t) = t_{\min} + t_{\mathrm{span}} e_t, \tag{12}$$

$$\Gamma(G_t) = \Gamma_0 + \Gamma_s \frac{(g_t + \epsilon_g)^{\gamma} - \epsilon_g^{\gamma}}{(1+\epsilon_g)^{\gamma} - \epsilon_g^{\gamma}}. \tag{13}$$

Active and passive returns are

$$S_{tij}^{\mathrm{act}} = \frac{k_a p_t \tau(E_t)}{\bar{Z}_{tij}^2 + c_d}, \tag{14}$$

$$\begin{aligned} S_t^{\mathrm{pass}} &= k_p \tau(E_t) \sqrt{\Gamma(G_t)}, \\ S_{tij} &= S_{tij}^{\mathrm{act}} + S_t^{\mathrm{pass}}. \end{aligned} \tag{15}$$

The inverse-square term models projected-pattern attenuation; the passive term captures ambient or texture-assisted correspondence. Signal, washout, motion, edge, and range factors yield

$$\begin{aligned} I_{tij}^{\mathrm{amb}} &= a_0 + a_1 \Omega_{tij}, \qquad R_{tij} = [\bar{Z}_{tij}/z_{\max} - r_0]_+, \\ B_t^{\mathrm{mot}} &= \mathrm{clip}(k_m \|\mathbf{v}_t\|_2 \tau(E_t), 0, b_{\max}), \\ W_{tij}^{\mathrm{out}} &= \frac{I_{tij}^{\mathrm{amb}} \tau(E_t)}{S_{tij}^{\mathrm{act}} + c_w}, \qquad N_{tij}^{\mathrm{shot}} = \frac{n_0(n_1 + n_2 \Gamma(G_t))}{S_{tij} + c_n}, \\ \chi_{tij} &= \frac{S_{tij}}{b_0 + b_1 I_{tij}^{\mathrm{amb}} + b_2 N_{tij}^{\mathrm{shot}} + b_3 B_t^{\mathrm{mot}}(b_4 + \delta_{tij})}, \\ Q_{tij}^0 &= \sigma\Big(\beta_s \chi_{tij} - \beta_w W_{tij}^{\mathrm{out}} - \beta_e \delta_{tij} - \beta_r R_{tij}\Big). \end{aligned} \tag{16}$$

The terms I^{amb}, W^{out}, N^{shot}, and χ denote ambient infrared level, washout, noise proxy, and signal-to-noise proxy. Their coefficients encode qualitative active-stereo re-

sponse regimes. Regime-specific penalties A_ρ and bonuses B_ρ adjust quality inside the hazard mask:

$$\begin{aligned} C_{\rho,t} &= B_{\rho_t}(p_t, e_t, g_t) - A_{\rho_t}(p_t, e_t, g_t), \\ Q_{tij} &= \mathrm{clip}\Big(Q^0_{tij} + \Omega_{tij} C_{\rho,t}, 0, 1\Big). \end{aligned} \tag{17}$$

Glare favors high power with low exposure/gain; specularity penalizes excessive active return; dark material rewards recovery of weak returns while retaining blur/noise costs. The resulting quality score defines soft and hard validity:

$$\begin{aligned} \widetilde{V}_{tij} &= \sigma((Q_{tij} - q_0)/s_v), \qquad V_{tij} = \mathbb{I}[Q_{tij} > q_0], \\ D^{\mathrm{obs}}_{tij} &= \bar{Z}_{tij} V_{tij}, \quad Q^{\mathrm{obs}}_{tij} = Q_{tij} V_{tij}. \end{aligned} \tag{18}$$

We use $q_0 = 0.42$ and $s_v = 0.055$. Valid pixels retain clipped depth; specular blooming is approximated by dropout, not multipath-induced false surfaces. Here Q is a reliability score and $\widetilde{V}$ is its smooth validity gate. The teacher uses soft validity; forward observations use hard validity, with straight-through gate $\mathrm{sg}[V - \widetilde{V}] + \widetilde{V}$ when needed. Evaluation reports full-image fill

$$F_t = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} V_{tij}. \tag{19}$$

3.3. Reliability-Based Camera Teaching

Image-wide fill can hide a poorly observed aperture boundary. The teacher, therefore, targets the least reliable patches at each fixed state, optimizing a bounded target $\bar{\mathbf{c}} = \mathrm{clip}(\sigma(\boldsymbol{\eta}), \epsilon, 1-\epsilon)$ with $\epsilon = 0.001$. A 6×8 adaptive-pooling grid and lower-tail fraction β define

$$\begin{aligned} \mathbf{p}(\bar{\mathbf{c}}) &= \mathrm{vec}\Big(\mathrm{Pool}^{R_p \times C_p}_{\mathrm{avg}}[\widetilde{V}(\bar{\mathbf{c}})]\Big), \\ K &= \lceil \beta R_p C_p \rceil, \\ r_{\mathrm{fill}}(\bar{\mathbf{c}}) &= \frac{1}{K} \sum_{i \in \mathcal{B}_K(\mathbf{p})} p_i, \end{aligned} \tag{20}$$

The set $\mathcal{B}_K$ contains the K smallest patch means. This piecewise-smooth statistic emphasizes weak regions without explicitly identifying the aperture. The teacher balances reliability, register variation, and acquisition costs:

$$\begin{aligned} \bar{\mathbf{c}}_{t+1} &= \arg \min_{\bar{\mathbf{c}} \in [\epsilon, 1-\epsilon]^3} \mathcal{L}_{\mathrm{teach}}(\bar{\mathbf{c}}), \\ \mathcal{L}_{\mathrm{teach}} &= \lambda_f [r_{\min} - r_{\mathrm{fill}}(\bar{\mathbf{c}})]^2_+ + \frac{\lambda_s}{3} \|\bar{\mathbf{c}} - \mathbf{c}_t\|_2^2 \\ &\quad + \lambda_P [\bar{P} - P_0]^2_+ + \lambda_b \|\mathbf{v}_t\|_2^2 \tau(\bar{E})^2 \\ &\quad + \lambda_G \bar{G}^2 + \frac{\lambda_0}{3} \omega(\bar{\mathbf{c}}) \|\bar{\mathbf{c}} - \mathbf{c}_{\mathrm{nom}}\|_2^2. \end{aligned} \tag{21}$$

The terms penalize insufficient reliability, register changes, excess power, motion-dependent exposure blur, and gain. Nominal recovery is weighted by

$$\omega(\bar{\mathbf{c}}) = \mathrm{sg}\left[\mathrm{clip}\left(\frac{r_{\mathrm{fill}}(\bar{\mathbf{c}}) - r_{\min}}{m_{\mathrm{fill}}}, 0, 1\right)\right], \tag{22}$$

Here sg stops gradients through the recovery gate. Logits start at $\mathrm{logit}(\mathrm{clip}(\mathbf{c}_t, \epsilon, 1-\epsilon))$; Adam returns a finite-step label rather than an exact minimizer. Primary teaching

uses 120 steps at learning rate 0.08. Numerical weights and ablation budgets are given in Tables 1 and 2. Teaching is offline and adds no iterative optimization to deployed policy inference.

Table 1. Numerical targets and objective weights in the accompanying implementation. The parameter names follow the teacher and flight objectives in Section 3; a zero weight disables the corresponding penalty.

Parameter	Symbol	Value
Reliability target and patch selection		
Patch grid	$R_p \times C_p$	6×8
Lower-tail fraction	β	0.25 (12 patches)
Reliability target	$r_{\min}$	0.9
Nominal-recovery margin	m_{fill}	0.25
Projector-power reference	P_0	0.5
Teacher objective weights		
Reliability shortfall	λ_f	50
Register-change penalty	λ_s	5×10^{-3}
Excess-power penalty	λ_P	0
Motion-dependent blur	λ_b	1.5×10^{-4}
Gain penalty	λ_G	7×10^{-4}
Nominal-register recovery	λ_0	0.075
Camera fitting		
SmoothL1 transition	δ	1
Temporal fitting penalty	$\lambda_{\Delta c}$	0
Flight objective weights		
Velocity tracking	λ_v	10.5
Acceleration regularization	λ_a	0.05
Jerk regularization	λ_j	0.05
Collision barrier	λ_c	6

Table 2. Training and evaluation settings that differ between the primary and ablation comparisons. Outcomes are compared within each protocol.

Quantity	Primary Comparison	Ablation Comparisons
Teacher updates per state	120	50
Camera pretraining/retraining	180/120 epochs	120/120 epochs
Evaluation stopping limits	60 steps	5 s timeout and 120-step cap
Episodes per regime	300 per method	300 per method per run
Reported aggregation	900 episodes per method	Three runs; 900 episodes per method per run

3.4. Learning Camera Control on Student-Induced States

The camera and flight branches have independent visual stems (Figure 3) but receive the same acquired depth. A camera collection contains observation/register/motion inputs $\mathbf{o}^c_{b,t}$ and teacher targets

$$\mathcal{D} = \{(\mathbf{o}^c_{b,t}, \bar{\mathbf{c}}_{b,t+1})\}_{b,t}. \tag{23}$$

Camera fitting unrolls its recurrent state over B sequences of T steps and minimizes

$$\mathcal{L}_{\text{sup}} = \frac{1}{3BT}\sum_{b=1}^{B}\sum_{t=0}^{T-1}\rho_\delta(\mathbf{c}^\star_{b,t+1} - \bar{\mathbf{c}}_{b,t+1}) + \lambda_{\Delta c}\frac{1}{3B(T-1)}\sum_{b=1}^{B}\sum_{t=1}^{T-1}\|\mathbf{c}^\star_{b,t+1} - \mathbf{c}^\star_{b,t}\|_2^2, \tag{24}$$

Here ρ_δ sums component-wise SmoothL1 losses; division by three averages register channels. Camera fitting uses $\delta = 1$ and $\lambda_{\Delta c} = 0$; temporal register regularization is, therefore, disabled during fitting. Only camera parameters Θ_{cam} are updated. Because learned acquisition changes subsequent observations, pretraining on $\mathcal{D}_0$ induces a distribution shift at deployment. We query the same teacher on student rollouts and retrain on

$$\mathcal{D}_{\text{dag}} = \left\{(\xi_t, \bar{\mathbf{c}}_{t+1}) \,\middle|\, \xi_t \sim \mu(\theta_{\text{pre}}),\ \bar{\mathbf{c}}_{t+1} = \arg\min_{\bar{\mathbf{c}}} \mathcal{L}_{\text{teach}}\right\}. \tag{25}$$

The distribution $\mu(\theta_{\text{pre}})$ includes the student's flight and acquisition history; ξ_t denotes the visited state. The student retains flight control and executes its own smoothed camera targets. One relabeling/retraining round is used; DAgger in the experiment labels denotes this combined stage. Collection sizes are specified separately from evaluation horizons in Table 3.

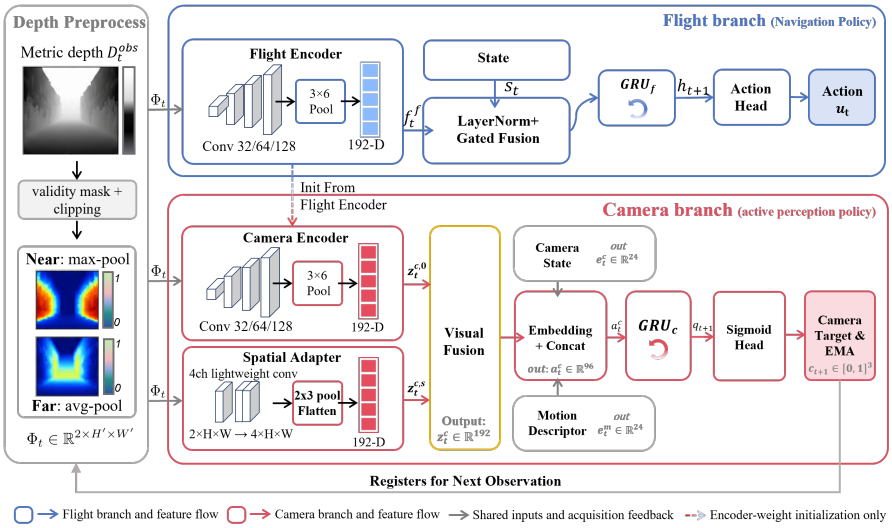


Figure 3. Split-stem recurrent policy. Depth preprocessing supplies near-obstacle and far-opening channels to independent flight and camera encoders. The flight branch fuses visual and vehicle-state features to output local acceleration $\mathbf{u}_t$. The camera branch combines global and spatial depth features with register and motion embeddings to predict an absolute camera target, followed by exponential moving average (EMA) smoothing. The dashed inter-stem arrow denotes initialization from the flight stem; subsequent parameters are independent. The bottom feedback arrow summarizes acquisition of the next depth frame through the sensor model. Blue and red identify the flight and camera branches and their feature flow, respectively. Gray arrows indicate shared inputs and acquisition feedback. Curved arrows within the gated recurrent units (GRUs) denote recurrent state updates.

Table 3. Teacher-supervision collections. Each stage uses four rollout batches per regime and 12 parallel trajectories per batch. Steps per sequence denote stored training horizons, independently of evaluation limits.

Collection Stage	Sequences	Steps/Sequence	State Labels
Initial teacher collection $\mathcal{D}_0$	144	60	8640
Student-rollout relabeling	144	80	11,520

3.5. Flight Adaptation Under Learned Acquisition

The final stage learns navigation under the new depth stream while preserving the camera mapping. Frozen parameters do not imply constant registers: camera actions remain responsive to observations. The camera output is detached from flight gradients,

$$\mathbf{c}^{\star}_{t+1} = \mathrm{sg}\Big[\Pi^{c}_{\theta_{\mathrm{cam}}}(D^{\mathrm{obs}}_{t}, \mathbf{c}_{t}, \mathbf{m}_{t}, \mathbf{q}_{t})\Big], \quad \theta_{\mathrm{cam}} \text{ fixed}, \tag{26}$$

and flight minimizes

$$\begin{aligned} \mathcal{L}_{\mathrm{flight}} &= \lambda_{v}\overline{\mathcal{L}}_{v} + \lambda_{a}\overline{\mathcal{L}}_{\mathrm{acc}} \\ &\quad + \lambda_{j}\overline{\mathcal{L}}_{\mathrm{jerk}} + \lambda_{c}\overline{\mathcal{L}}_{\mathrm{collide}}, \\ \theta^{\star} &= \arg\min_{\Theta_{f}} \mathbb{E}_{\rho,\mathbf{x}_{0}}\mathcal{L}_{\mathrm{flight}}. \end{aligned} \tag{27}$$

Overbars average valid time indices and the batch. Velocity tracking uses a $W_v = 12$ step window and the same SmoothL1 transition $\delta = 1$, with a goal-directed reference,

$$\mathcal{L}_{v}(t) = \rho_{\delta}\left(\left\| \frac{1}{W_{v}} \sum_{i=0}^{W_{v}-1} \mathbf{v}_{t-i} - \mathbf{v}^{\mathrm{ref}}_{t} \right\|_{2}\right), \tag{28}$$

$$\mathcal{L}_{\mathrm{acc}}(t) = \|\mathbf{a}_{t}\|_{2}^{2}, \qquad \mathcal{L}_{\mathrm{jerk}}(t) = \|15(\mathbf{a}_{t} - \mathbf{a}_{t-1})\|_{2}^{2}. \tag{29}$$

The jerk scale uses a nominal 15 Hz. Clearance is queried at ten extrapolated positions $\mathbf{p}_t + \mathbf{v}_t k/(9 \cdot 15)$, $k = 0, \ldots, 9$, using a spherical body radius of 0.12 m and zero additional training margin. These finite lookahead queries approximate continuous swept-volume checking. With clearance $d_{t,k}$, the barrier is

$$\mathcal{L}_{\mathrm{collide}}(t) = \frac{1}{K_{o}} \sum_{k=1}^{K_{o}} \zeta_{t,k}\, \mathrm{softplus}(-32 \max(d_{t,k}, -3)), \tag{30}$$

Here $K_o = 9$: the barrier averages the nine forward samples $k = 1, \ldots, 9$, while $k = 0$ supplies the initial clearance for differencing. The detached factor $\zeta_{t,k} = \mathrm{sg}[\max(1, 135(d_{t,k-1} - d_{t,k}))]$ upweights decreasing clearance along this lookahead, not across rollout time. Auxiliary avoidance and camera losses have zero weight during flight adaptation.

3.6. Policy Implementation

Observed depth is encoded into near and far channels. Define $V^{d} = \mathbb{I}[D^{\mathrm{obs}} \geq z_{\min}]$ and let $\bar{D}$ equal clipped valid depth, with $z_{\max}$ substituted at invalid pixels. Then

$$\begin{aligned} \phi^{\mathrm{near}}_{t} &= \mathrm{Pool}^{H_d \times W_d}_{\max}\left[V^{d}_{t} \odot \mathrm{clip}\left(\frac{1/\bar{D}_{t} - 1/z_{\max}}{1/z_{\min} - 1/z_{\max}}, 0, 1\right)\right], \\ \phi^{\mathrm{far}}_{t} &= \mathrm{Pool}^{H_d \times W_d}_{\mathrm{avg}}\left[V^{d}_{t} \odot \mathrm{clip}\left(\frac{\bar{D}_{t} - z_{\min}}{z_{\max} - z_{\min}}, 0, 1\right)\right], \\ \Phi_{t} &= 2[\phi^{\mathrm{near}}_{t}, \phi^{\mathrm{far}}_{t}] - 1. \end{aligned} \tag{31}$$

The 36×48 near channel uses inverse depth and max pooling; the far channel uses metric depth and average pooling. Both suppress invalid pixels. Each independent visual stem has 32/64/128 convolutional channels, stride-2 downsampling in the second block, 3×6 pooling, and a 192-dimensional projection. The flight branch gates layer-normalized visual and state features, updates a GRU, applies a 0.1-scaled residual MLP and layer normalization, and predicts acceleration. LeakyReLU uses slope 0.05.

The camera stem is initialized from the flight stem but shares no parameters thereafter. A 4-channel spatial adapter with 2×3 pooling and a residual image adapter contribute with scales 0.35 and 0.25, respectively. Current registers $2\mathbf{c}_t - 1$ and the motion descriptor

$$\mathbf{r}_t^{\mathrm{up}} = \mathbf{R}_t \mathbf{e}_3, \qquad \mathbf{m}_t = [\mathrm{clip}(\mathbf{v}_t^{\mathrm{local}}/v_{\max}, -2, 2), \mathbf{r}_t^{\mathrm{up}}] \in \mathbb{R}^6, \tag{32}$$

are each embedded into 24 dimensions and fused with image features. Here $\mathbf{e}_3 = [0, 0, 1]^\top$; the up vector is in the world frame. The fused camera feature $\mathbf{a}_t^c$ feeds a 96-dimensional GRU; with φ denoting LeakyReLU and W_o the affine output map, it predicts

$$\mathbf{q}_{t+1} = \mathrm{LN}(\mathrm{GRU}_c(\mathbf{a}_t^c, \mathbf{q}_t)), \qquad \mathbf{c}_{t+1}^\star = \sigma(W_o \varphi(\mathbf{q}_{t+1})). \tag{33}$$

All camera visual modules, state/motion embeddings, fusion layers, recurrent units, output heads, and normalization layers belong to Θ_{cam}. They are trained during camera fitting and frozen during flight adaptation. The sigmoid yields absolute normalized registers; Equation (7) applies smoothing outside the network.

4. Experimental Setup

Experiments use a custom PyTorch/CUDA simulator, available at https://github.com/Gaiwqqq/ActiveDepthSensingForUAV (accessed on 21 October 2026). The archived simulation environment specifies Python 3.9.23, PyTorch 2.1.0, CUDA 12.1, NumPy 1.26.4, and pandas 2.1.3. It models vehicle dynamics, ray-based depth, and obstacle geometry; the acquisition surrogate converts ideal ranges into register-dependent observations. This controlled aperture-crossing study evaluates complete-policy navigation, training design, and robustness, with separate physical-camera and SIH experiments.

4.1. Benchmark and Training Configuration

Table 4 summarizes the core simulation and evaluation settings. The primary evaluator uses 60 steps at nominal 15 Hz, with exposure-dependent timing and earlier terminal events. Start, front wall, and goal are at local $x = -1.5, 0, 1.5$ m; the back wall is at $x \in [2.00, 2.85]$ m. The aperture has half-width 0.15 m, reset-sampled center $y \sim \mathcal{U}[-0.75, 0.75]$ m, and vertical parameter $z = 1.50$ m. Random yaw rotates the complete scene; geometry is static within each episode. Glare affects the aperture halo, whereas dark material and specularity affect adjacent-wall returns (Figure 4).

Table 4. Core simulation and evaluation settings. Evaluation uses 300 episodes per sensing regime for each primary method.

Setting	Value
Control frequency	15 Hz
Evaluation horizon	60 steps
Rendered depth resolution	96×72
Policy depth resolution	48×36
Camera action	power/exposure/gain in $[0, 1]^3$
Sensing regimes	glare, dark material, specular
Aperture center	$y \sim \mathcal{U}[-0.75, 0.75]$ m at each reset; $z = 1.50$ m
Scene yaw	randomized
Evaluation episodes	900 per primary method

Table 5 summarizes the stage-wise training settings.

Table 5. Stage-wise training settings. Camera-fitting epochs follow the primary configuration reported in this study; optimizer settings and the flight-adaptation budget are specified from the accompanying training implementation. Each stage updates only its designated parameters.

Stage	Optimization	Budget and Batch Size
Camera teaching	Adam; learning rate 0.08	120 updates per state; 12 states per batch
Camera pretraining	AdamW; learning rate 10^{-3}	180 epochs; 8 sequences per batch
Camera retraining	AdamW; learning rate 5×10^{-4}	120 epochs; 8 sequences per batch
Flight adaptation	AdamW; learning rate 8×10^{-5}, cosine decay to 8×10^{-7}	3000 scheduled updates; 96 trajectories per batch

Table 3 summarizes the two supervision collections. The initial teacher collection $\mathcal{D}_0$ is configured with three regimes, four rollout batches per regime, 12 parallel trajectories per batch, and 60 steps per trajectory. The collector queries the teacher at every step and stores the full horizon without early termination, giving $3 \times 4 \times 12 = 144$ sequences and $144 \times 60 = 8640$ state-wise labels (48 sequences and 2880 labels per regime). Each label contains the three camera registers. Student-rollout relabeling uses the same batch structure with 80 steps per trajectory, giving 144 sequences and 11,520 labels. These training-sequence horizons are independent of evaluation stopping limits. One relabeling round requires $11{,}520 \times 120 = 1{,}382{,}400$ state–optimizer-step evaluations, implemented as 115,200 batched updates of 12 states, excluding collection time.

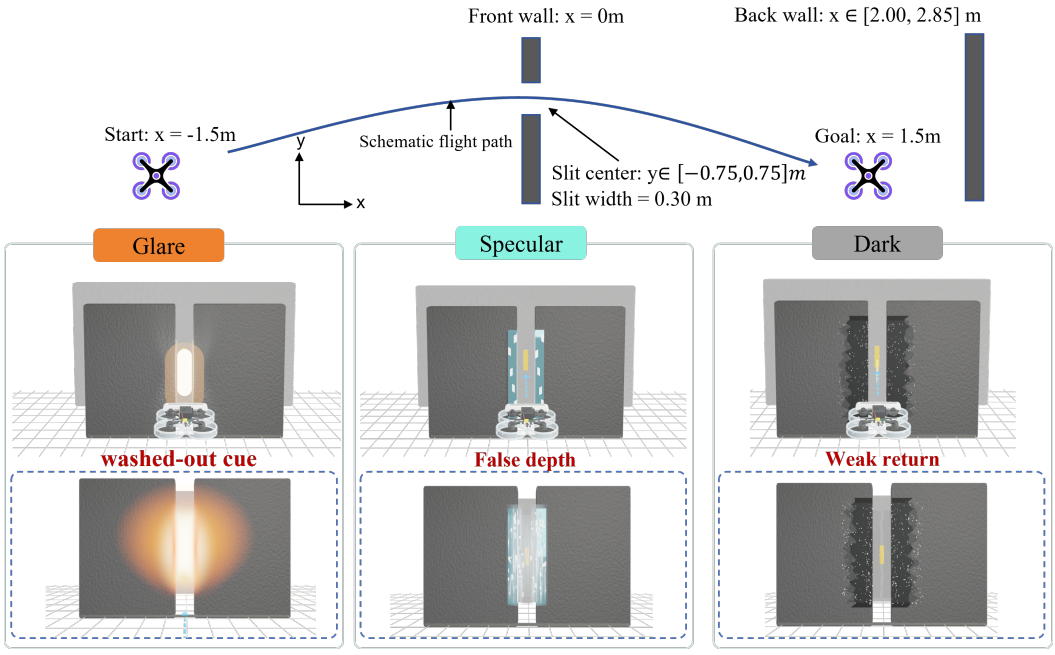


Figure 4. Conceptual evaluation scenes. Top: aperture-crossing geometry in the local scene frame, with coordinates in meters; the blue curve is a schematic path, not an evaluated trajectory, and the drawing is not to scale. Bottom, left to right: glare, specular, and dark-material sensing regimes, each with an approach view and a closer view of the affected region. The illustrations depict physical mechanisms motivating the surrogate; the implemented response models their effect on depth availability, with valid pixels retaining rendered depth. These are conceptual illustrations, not simulator screenshots or physical flight experiments. Geometry and randomization are shared across regimes. Colored headers identify the sensing regimes, and blue dashed boxes identify close-up views. The "False depth" label illustrates possible specular range errors; the base surrogate represents specular degradation through dropout rather than biased ranges. The separate test in Section 5.4 injects range bias explicitly.

4.2. Compared Policies

Table 6 separates complete-policy comparisons from intermediate training stages. Nominal fixed acquisition uses $(P, E, G) = (0.5, 0.5, 0.5)$; random-fixed draws $P \in [0.10, 0.95]$, $E \in [0.10, 0.92]$, and $G \in [0.02, 0.90]$ once per episode. Non-diff tests a camera-learning procedure without differentiable sensor training; Blind tests dependence on depth. These references address the acquisition interface within the task. Photometric auto-exposure [6] requires image information absent from the rendered depth interface, and external navigation systems differ in observations and training. The physical D455 comparison supplies a separate device-auto-exposure reference. We claim improvement over the evaluated complete policies, rather than a universal navigation ranking.

Table 6. Policies used in the quantitative comparison and diagnostic analysis. Only the first five methods are primary navigation baselines; pretrain and DAgger are included to separate camera semantics from flight adaptation. The no-DAgger and shared-stem variants are defined below.

Method	Role	Description
Ours	Main model	split-stem learned camera with flight adaptation
Nominal fixed	Baseline	nominal constant camera setting (fixed-camera in plots)
Random-fixed	Baseline	episode-constant random camera setting
Non-diff.	Baseline	learned camera without differentiable sensor training
Blind	Baseline	zero-depth controller
Pretrain	Diagnostic	camera-pretrained policy before student-rollout relabeling
DAgger	Diagnostic	camera-retrained policy before flight adaptation

4.3. Ablation Settings

Ablations use a 5 s timeout and 120-step cap, with three training runs evaluated on the same 900 episodes (300 per regime): 2700 executions per variant. Their 77.89% full result is not a replacement for the primary success of 75.1%. The stem comparison uses architecture-specific base models, 3000 base updates, 50 teacher steps, 120 epochs for each camera-fitting stage, and 3000 flight-adaptation updates. Shared stems remain trainable during flight adaptation; split stems preserve the camera encoder. Paired intervals use 10,000 hierarchical bootstrap resamples.

The no-DAgger comparison retains split stems and the common downstream adaptation budget, omitting only student-rollout relabeling and camera retraining. We report run means and sample standard deviations. Its 20,000-resample bootstrap is stratified by regime and resamples episode identifiers jointly across runs; intervals are conditional on the evaluated model pairs. Both ablations compare final navigation, rather than camera fit alone.

4.4. Metrics and Supplementary Robustness Protocols

Success requires reaching within 0.35 m of the goal at any evaluated step without collision over the rollout; collision overrides reaching. Collision uses radius-adjusted minimum clearance at the ten lookahead queries, checked before and after integration, with threshold 0.0011 m. Timeout denotes neither success nor collision. We average fill first over steps and then episodes, and report terminal goal distance over all outcomes. Primary binary intervals are Wilson 95% intervals; continuous metrics use episode bootstrap intervals, conditional on the policy.

Four added foreground layouts alter aperture visibility while retaining the frozen policy and primary evaluator. Layout selection uses separate validation seeds, 20 episodes per regime; testing uses 100 per regime for each layout and its paired original control (1500 episodes). Occlusion is the fraction of the initially visible aperture-observation ROI replaced by foreground returns, with wall-plane bounds $|y - y_a| < 0.15$ m and

$|z - 1.5| < 0.75$ m. Valid foreground depth can hide the opening without lowering global fill. Geometry and all regime outcomes appear in Appendix A.

The depth-bias test uses the same frozen split-stem policy, three bounds $b \in \{0, 0.10, 0.30\}$ m, and 100 episodes per regime per bound (900 total). Episodes are paired across bias levels using identical scene randomization. An independently sampled 6×8 random field is bilinearly interpolated to 72×96 and held fixed in image coordinates within each episode. Its sign is selected once per episode, and its magnitude is $0.5b$–b before clipping. Bias is added to every valid pixel before both policy branches; resulting depths are clipped to $[0.05, 6]$ m. Invalid pixels are unchanged for each individual rendering. Feedback may change subsequent trajectories and validity masks across runs. The smooth field defines a controlled systematic range perturbation. Paired success-change intervals use 10,000 regime-stratified bootstrap resamples.

4.5. Physical-Camera and Flight-Controller Experiments

The physical camera is an Intel RealSense D455 (Intel Corporation, Santa Clara, CA, USA), running firmware version 5.15.1.55. The D455 pilot uses one stationary setup and five blocks, each containing an 8 s capture from an adapted camera policy, device auto-exposure, and an illustrative manual 1 ms setting, with order permuted. Raw 640×480 depth is requested at 30 frames/s without postprocessing; full-frame coverage in $[0.05, 6]$ m is averaged by clip. Intervals resample the five clips or paired blocks, not correlated frames. A shared-stem camera branch was adapted by imitation of physically selected register targets; it differs from the frozen split-stem navigation policy. Appendix B gives the acquisition and adaptation settings.

In SIH, a physical open-source NxtPX4 flight-controller board (HKUST UAV Group, Hong Kong, China; https://github.com/Xiaodao-chen/NxtPX4 (accessed on 21 October 2026)) runs PX4 v1.15.4 dynamics/sensor simulation, estimation, and flight control. The hardware experiments use PyTorch 2.1.0 with CUDA 12.1. A host generates surrogate depth and runs the frozen policy, exchanging estimated state and acceleration/yaw commands through USB MAVLink. Rendering and outcome checks use SIH truth. The final comparison pairs 20 episodes per regime with pure simulation, sharing geometry, randomization, policy weights, and the 60-action limit. The final integration configuration is evaluated on fresh seeds following preliminary runs and is separate from the primary benchmark. No motors or physical vehicle motion are involved.

5. Results and Discussion

We first establish the navigation gain and the need for flight adaptation, then examine the learning design. The remaining tests assess acquisition behavior under occlusion, sensitivity to incorrect ranges, and execution through physical sensing and flight-controller interfaces.

5.1. Navigation Performance and Flight Adaptation

The complete differentiable-acquisition and flight-adaptation framework improves navigation (Figure 5). Success is 75.1% versus 66.7% for nominal fixed, 62.7% for random-fixed, 58.0% for Non-diff, and 3.2% for Blind. Relative to nominal fixed, collision falls from 33.3% to 24.9%, fill rises from 72.1% to 97.5%, and mean terminal distance decreases from 0.672 to 0.550 m.

Success gains concentrate in dark material (68.3% to 82.0%) and specularity (66.3% to 76.3%). Glare fill increases from 68.7% to 95.8%, but success rises only from 65.3% to 67.0%. Global availability is, therefore, an imperfect proxy for crossing: the teacher improves weak patches without explicitly prioritizing aperture boundaries, and flight must exploit useful

geometry in time. These results motivate learning control under the changed observation stream rather than equating high fill with navigation success.

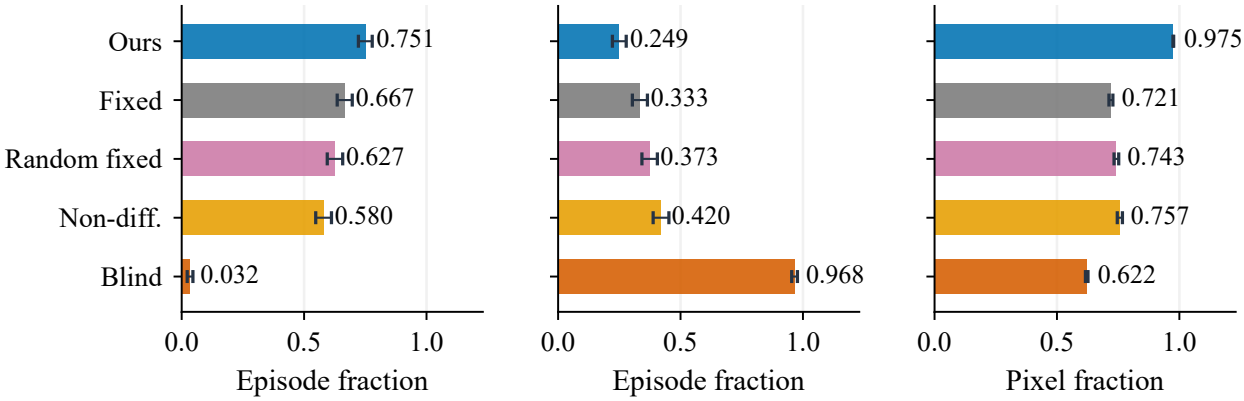


Figure 5. Held-out success, collision, and valid-depth fill rate. Fixed-camera denotes the nominal fixed-camera reference. Error bars show 95% confidence intervals: Wilson for binary outcomes and bootstrap for fill rate. For Blind, the plotted fill is a sensor-output diagnostic before input replacement; the depth input actually consumed by the policy is all zeros and has 0% valid-depth fill. The plotted value must not be interpreted as depth available to the blind controller.

The training-stage comparison makes that requirement explicit (Figure 6). The Pretrain and DAgger stages have approximately 95.8% fill but only 3.1% and 3.0% success. Flight adaptation raises success to 75.1% and fill to 97.5% while camera parameters remain frozen. Thus, reliable acquisition becomes useful through downstream adaptation; freezing its mapping still permits state-dependent register actions.

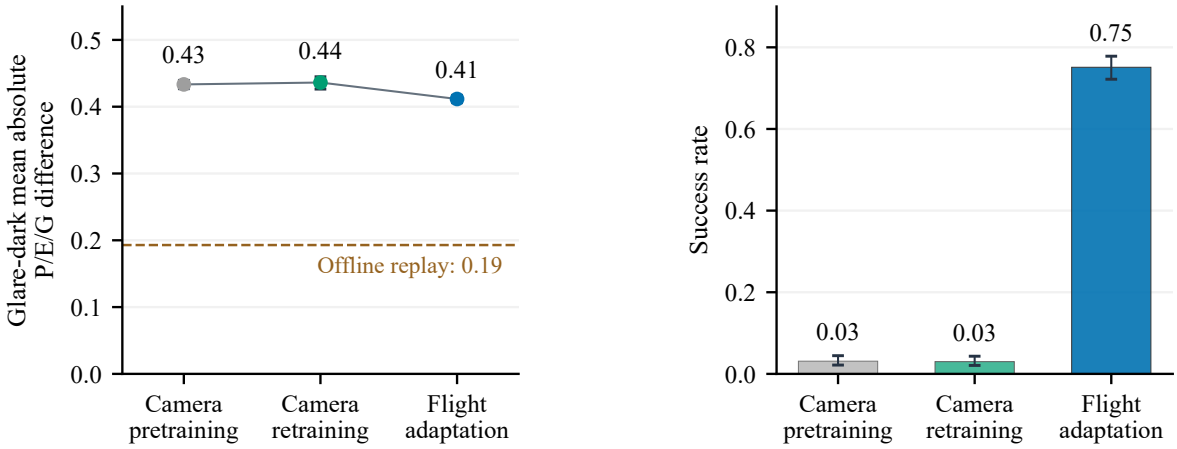


Figure 6. Camera-command separation (**a**) and success (**b**) after the three parameter-update stages. The dashed reference is the camera-retrained model's target predictions on offline replay, not teacher labels. Points in (**a**) use rollout commands. Error bars are bootstrap 95% intervals for separation and Wilson 95% intervals for success.

5.2. Ablation of the Learning Design

Under the separate ablation protocol, student-rollout relabeling plus camera retraining raises success by 7.11 points, from 70.78% to 77.89% (Table 7; paired 95% interval [5.78, 8.52]). All nine run–regime comparisons favor Full; across 2700 paired executions, 192 failures become successes and no success becomes failure. Fill changes only from 96.35%

to 96.79%, while terminal distance decreases from 0.754 to 0.654 m. The navigation gain, therefore, exceeds what the small aggregate fill change alone would suggest. Full records 701/900 successes in each run, explaining its zero overall sample standard deviation despite different regime-wise rates.

Table 7. Student-rollout relabeling and camera-retraining ablation after flight adaptation. Success rates (↑) are means ± sample standard deviations across the training runs described in Section 4.3. Each variant has 2700 test executions, including 900 per regime. Differences are Full minus no-DAgger in percentage points (pp). Both use split stems and the same flight-adaptation budget.

Regime	No-DAgger (%)	Full (%)	Δ (pp)
Overall	70.78 ± 1.02	77.89 ± 0.00	+7.11
Dark material	74.22 ± 0.51	81.67 ± 0.88	+7.44
Specular	71.33 ± 2.52	78.11 ± 0.51	+6.78
Glare	66.78 ± 1.26	73.89 ± 0.51	+7.11

Split stems improve overall success by 3.59 points over shared stems (77.89% versus 74.30%; interval [2.26, 4.89]), with fill increasing from 78.72% to 96.79% (Table 8). Glare is an exception: shared reaches 76.33% versus split 73.89%, with a split-minus-shared interval spanning zero. Preserving acquisition is beneficial overall, with regime-dependent gains. The comparison evaluates complete architectures, including their capacity and architecture-specific base models.

Table 8. Visual-stem comparison under the settings in Section 4.3. Success rates (↑) summarize 2700 paired test executions per variant, including 900 per regime. Differences are split minus shared in percentage points (pp). Intervals are paired hierarchical-bootstrap 95% intervals.

Regime	Shared (%)	Split (%)	Δ (pp)	95% Interval (pp)
Overall	74.30	77.89	+3.59	[2.26, 4.89]
Dark material	73.33	81.67	+8.33	[6.11, 10.67]
Specular	73.22	78.11	+4.89	[2.11, 7.44]
Glare	76.33	73.89	−2.44	[−5.78, 1.11]

5.3. Acquisition Behavior and Partial-Observation Robustness

Figure 7 compares nominal fixed and learned acquisition at identical poses in all three regimes. The first recorded pose and the pose nearest $x = -0.30$ m are shown to compare acquisition early in the approach and near the opening. Near the aperture, mean $P/E/G$ is 0.731/0.088/0.120 for glare, 0.665/0.694/0.683 for dark material, and 0.408/0.366/0.381 for specularity. These actions follow the surrogate's response factors. Matched-pose replay isolates their observation effects; each policy's closed-loop trajectories additionally reflect its visited states.

The four layouts vary how foreground geometry interrupts the approach. Layout A places two staggered boxes at different forward positions; B uses a one-sided box; C uses a wider offset box closer to the wall and D places a stepped pair near the start. The original wall, opening, and goal are retained. These boxes enter both depth rendering and collision queries, so the added geometry affects what the policy observes as well as where it can move. Figure 8 separates these effects: the trajectory panel shows the spatial response, the depth panel shows the acquired input, and the ROI panel identifies geometric occlusion. Unlike sensor dropout, geometric occlusion cannot be removed by changing acquisition registers. Navigation must, therefore, use the remaining observations as the viewpoint changes. The illustrated episodes are selected by identifier, while Table 9 summarizes the full evaluation set.

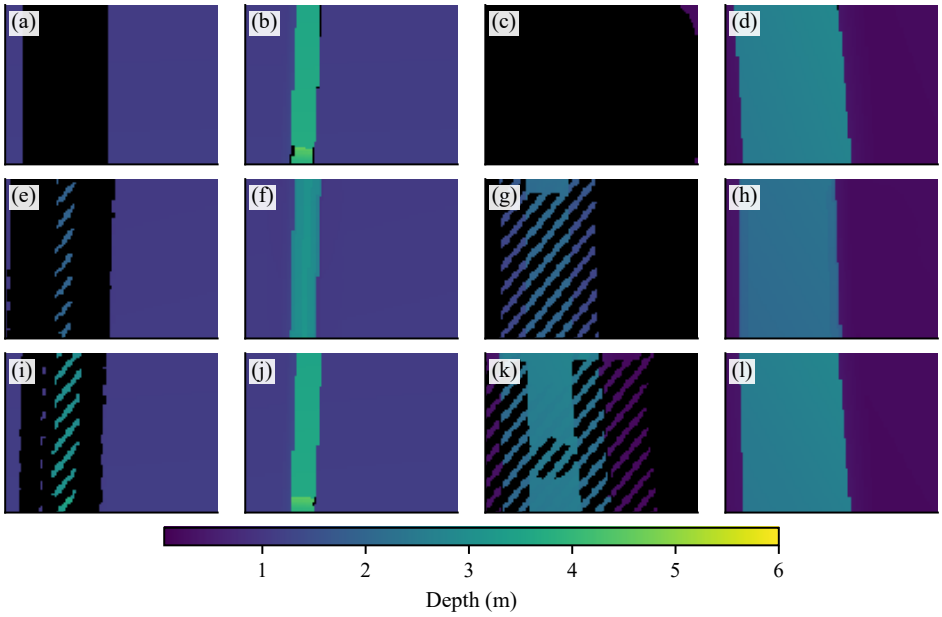


Figure 7. Matched-pose depth: (**a–d**) glare; (**e–h**) dark material; (**i–l**) specularity. Panels (**a,e,i**) and (**b,f,j**) show nominal fixed acquisition and Ours, respectively, at the first recorded pose; (**c,g,k**) and (**d,h,l**) show the same respective methods at the recorded pose nearest $x = -0.30$ m. Frames are selected by position, independently of image quality. Each pair shares geometry. The common depth scale is 0.05–6 m; black denotes invalid depth.

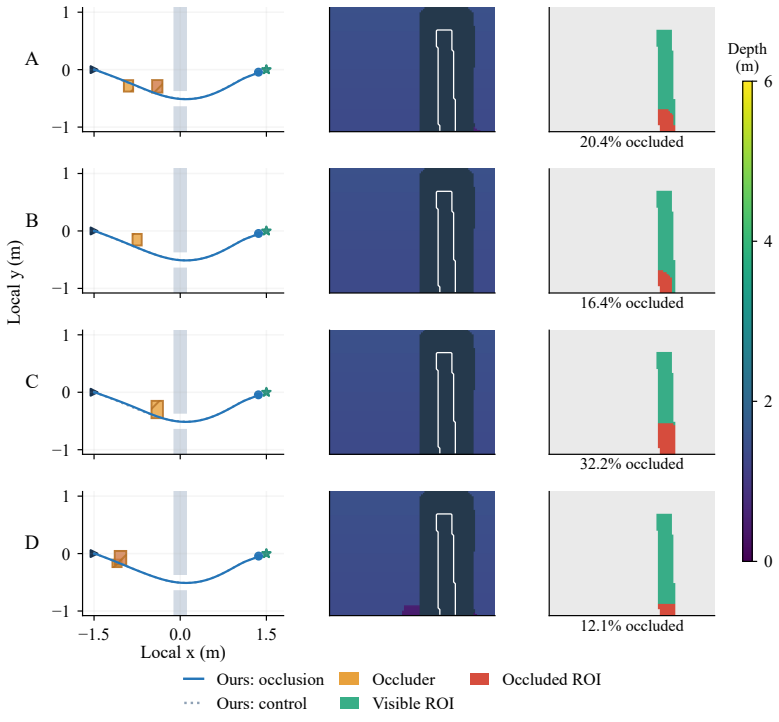


Figure 8. Four partial-observation layouts shown for the same preselected glare episode: (**A**) two staggered boxes; (**B**) one-sided box; (**C**) wider offset box; (**D**) stepped pair near the start. Left: Ours trajectories in the added layout (blue) and paired original scene (dotted gray); orange boxes are footprints, whose projected overlap need not imply collision. Middle: actual policy-input depth, with the original aperture-observation ROI outlined in white. Right: visible (green) and geometrically occluded (red) ROI pixels. Dark depth pixels are invalid measurements.

Added foreground occluders test whether the frozen policy retains this behavior when part of the opening is hidden. Across four validation-selected layouts, initial aperture-ROI

occlusion is 11.97%–30.59%, yet success remains 75.33%–79.67%, versus 74.67% in the paired original control (Table 9). Figure 8 connects actual depth, visibility, and trajectories. This is retention of Ours, not a new baseline ranking; intervals do not establish improvement in every layout. Appendix A retains all conditions, including layout C's dark-material decline from 83% to 78%.

Table 9. Frozen Ours: 300 episodes per setting (100 per regime). Occl.: mean initial aperture-ROI occlusion; Distance: mean final goal distance over all outcomes; ΔSR: paired change from the within-suite original control, in percentage points (pp). All timeout rates are 0%.

Layout	Occl. (%)	Success (%) ↑	Collision (%) ↓	Distance (m) ↓	ΔSR (pp)
Original	0.00	74.67	25.33	0.550	+0.00
A: staggered pair	19.19	79.67	20.33	0.472	+5.00
B: one-sided	14.50	76.33	23.67	0.523	+1.67
C: offset wide	30.59	79.00	21.00	0.480	+4.33
D: stepped	11.97	75.33	24.67	0.540	+0.67

5.4. Sensitivity to Biased Valid Depth

The bias experiment tests a failure mode absent from the training surrogate: valid measurements with incorrect ranges. The frozen policy retains 70.00% success at a 0.10 m bound, a 4.67-point decrease from its paired 74.67% zero-bias control. At 0.30 m, success falls to 37.00%, and collision rises to 63.00% (Table 10). Mean terminal distance increases from 0.562 to 0.633 and 1.138 m. All 900 episodes are included; no policy is retrained or selected using bias outcomes.

Table 10. Frozen-policy bounded-depth-bias test: 300 paired episodes per setting, 100 per regime. Intervals describe paired success changes conditional on the policy and perturbation. All timeout rates are 0%.

Bias Bound	Success (%)	Collision (%)	Fill (%)	Δ Success [95% CI] (pp)
0 m	74.67	25.33	97.60	Reference
0.10 m	70.00	30.00	97.42	−4.67 [−8.00, −1.33]
0.30 m	37.00	63.00	96.08	−37.67 [−43.33, −32.33]

Fill remains 97.42% at 0.10 m and 96.08% at 0.30 m, despite the success loss. This directly demonstrates that availability cannot substitute for range correctness. The method retains substantial crossing capability under the smaller perturbation but is not robust to the larger systematic bias. The resulting failure boundary establishes range correctness as a separate requirement for reliable navigation: differentiable acquisition improves the availability of useful observations, while systematic range errors can still invalidate the geometry used by the controller.

5.5. Physical Acquisition and Hardware-in-the-Loop Navigation

Physical acquisition improves after camera adaptation: D455 coverage is 50.07% versus 48.79% for device auto-exposure, a paired gain of 1.28 points (95% interval [1.21, 1.35]), positive in all five blocks. Throughput is 26.86 versus 29.99 frames/s (Table 11); the gain has a measurable delivery-rate cost. Figure 9 shows raw physical captures. The manual 1 ms setting was deliberately chosen as an underexposed illustration, not an optimized fixed baseline. This one-setup pilot establishes an executable camera interface after adaptation, rather than zero-shot surrogate transfer or depth accuracy.

Table 11. Stationary D455 acquisition: five 8 s clips per method. Coverage is averaged per clip; brackets give 95% clip-bootstrap intervals. Frames/s is measured delivered throughput. The manual setting is an illustrative underexposed reference.

Acquisition	Frames	Valid Depth (%)	Frames/s
Adapted camera policy	1077	50.07 [50.00, 50.15]	26.86
Device auto-exposure	1200	48.79 [48.77, 48.82]	29.99
Manual 1 ms (illustrative)	1200	46.46 [46.39, 46.53]	29.99

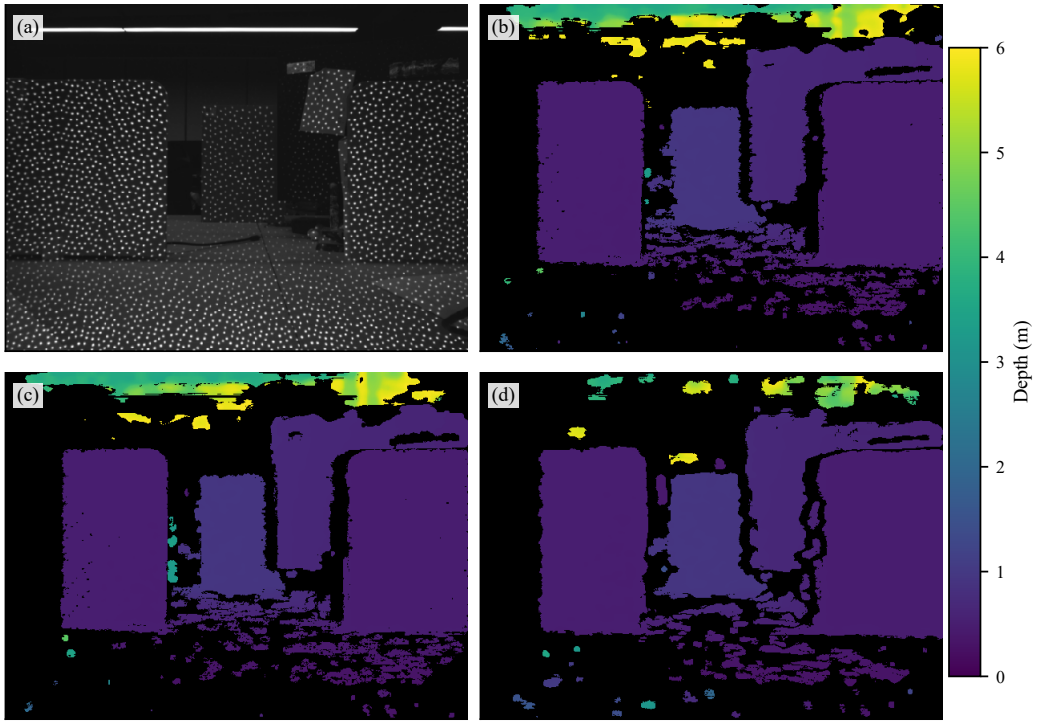


Figure 9. Physical D455 observations in the stationary setup: (**a**) left-infrared image at the adapted-policy capture; (**b**) adapted-policy depth; (**c**) device auto-exposure depth; (**d**) illustrative manual 1 ms depth. Frames are nearest the midpoint of each clip in the last complete acquisition block, rather than selected for maximum coverage. Depth panels use the same 0–6 m scale, with invalid/out-of-range pixels black and no hole filling. The views are sequential captures of the same stationary setup.

Figure 10 shows the coverage–throughput tradeoff in all five blocks. Each point summarizes a complete clip; consistency across these repeated captures concerns one stationary setup, not independent scenes.

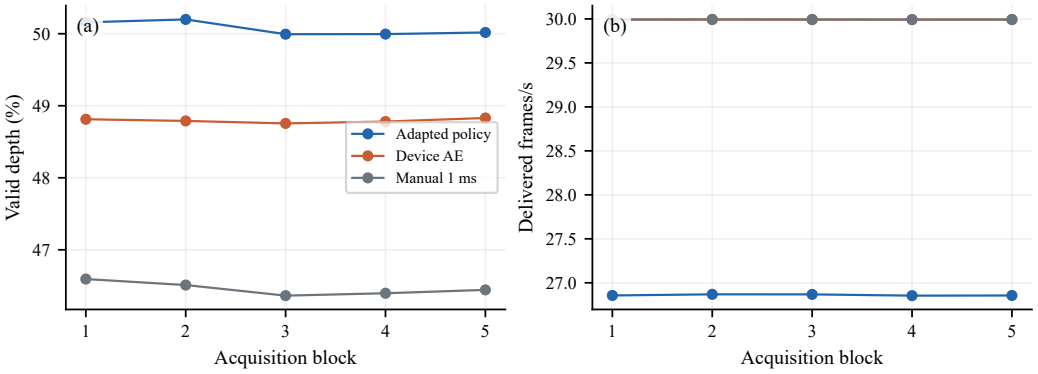


Figure 10. Physical D455 results by acquisition block: (**a**) full-frame valid-depth coverage and (**b**) delivered frame rate. Each point summarizes one 8 s clip; every block contains the adapted camera policy, device auto-exposure, and illustrative manual reference in permuted order. The two panels show the coverage gain and throughput cost within the same stationary setup.

The frozen navigation policy completes 44/60 SIH crossings (73.33%) versus 47/60 (78.33%) in matched pure simulation (Table 12). There are 44 joint successes, 13 joint failures, and three pure-simulation-only successes. All trajectories appear in Figure 11. The observed 5.00-point execution loss is descriptive, not an equivalence result. Across 2741 SIH steps, host processing, inference, and state-age 95th percentiles are 30.24, 8.63, and 20.91 ms; no scheduled slots are skipped. Per-trajectory scheduling spans 50.59–112.15 ms with jitter. Requiring 15 mm additional clearance retains 43/60 successes.

Table 12. Matched pure-simulation and PX4 SIH navigation with the frozen policy. Each regime has 20 paired episodes with shared geometry and seeds. Counts retain all outcomes; these supplementary results are separate from the primary benchmark.

Sensing Regime	Successful Crossings		SIH Failures	
	Pure sim.	PX4 SIH	Collision	Timeout
Glare	12/20	11/20	9	0
Specular proxy	19/20	18/20	2	0
Dark-material proxy	16/20	15/20	4	1
All regimes	47/60	44/60	15	1
Success rate	78.33%	73.33%		

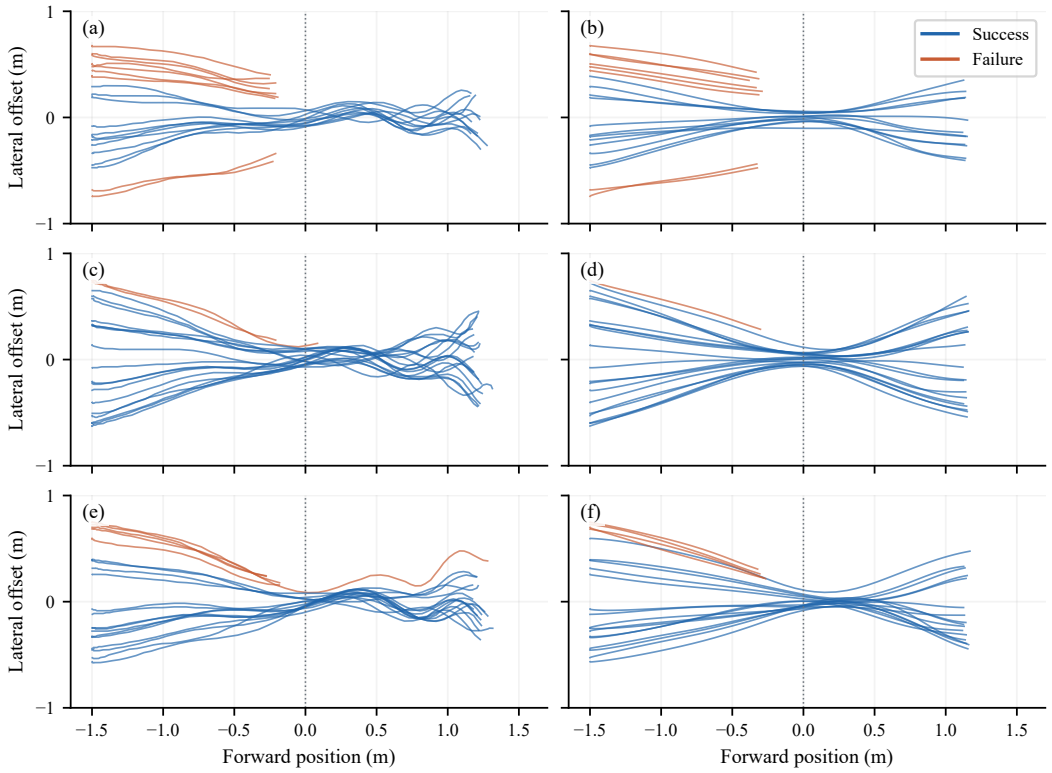


Figure 11. All 60 PX4 SIH trajectories (left column) and their 60 matched pure-simulation trajectories (right). Panels (**a**,**b**) show glare, (**c**,**d**) the specular proxy, and (**e**,**f**) the dark-material proxy; (**a**,**c**,**e**) are PX4 SIH and (**b**,**d**,**f**) are matched pure simulation. Each panel contains 20 episodes. Coordinates use the scene frame, with lateral position relative to the episode's aperture center and the dotted line indicating the front-wall plane. Blue and orange distinguish successful and failed episodes. SIH traces use ground truth for evaluation; the policy acts on estimated state. The depth observations in both runs are synthetic.

Together, the experiments connect differentiable acquisition to navigation gains and exercise physical sensing and flight-controller interfaces. The evidence concerns aperture crossing under the stated sensing and state-estimation models. D455 evaluates a separately

adapted camera branch in a stationary setup; SIH evaluates the frozen navigation policy with synthetic depth, simulated dynamics, and hardware-based estimated-state feedback. These complementary tests establish acquisition feasibility and control-interface compatibility. The smaller glare gain and the severe-bias failure boundary delimit the navigation benefit: improved depth availability must be coupled with flight adaptation and sufficiently accurate ranges.

6. Conclusions

This study presents a staged learning framework that improves UAV navigation by coupling differentiable depth sensing with flight adaptation. Reliability-based supervision and student-state relabeling train the camera policy, while a separate flight branch adapts to the resulting observations without changing the learned camera mapping. In the primary aperture-crossing benchmark, success increases from 66.7% with nominal fixed acquisition to 75.1%. Ablation studies support the learning design, while additional layouts and depth-bias tests characterize performance under varied observation conditions. Physical D455 acquisition and PX4 SIH experiments provide complementary evidence on sensing and controller integration.

The evaluation focuses on aperture crossing under controlled sensing conditions, allowing the relationship between depth acquisition and navigation performance to be examined systematically. The findings, therefore, apply to the scene and sensor configurations studied, with broader applicability across environments, platforms, and operating conditions requiring further investigation.

Building on these findings, future work will extend the framework to more general navigation tasks and sensing platforms. This will involve learning from physical sensor data and integrating adaptive depth acquisition more closely with state estimation and flight control, with the aim of supporting navigation in complex environments.

Author Contributions: Conceptualization, W.G. and Z.M.; methodology, W.G.; software, W.G.; formal analysis, G.Z.; resources, Z.M.; data curation, G.Z.; writing—original draft preparation, W.G. and G.Z.; writing—review and editing, W.G., G.Z., and J.J.; visualization, G.Z.; project administration, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the "Vice President of Science and Technology" Project of Changping District, Beijing (No. 202603002020).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code, configuration files, evaluation scripts, and trained model weights are openly available in the public GitHub repository at https://github.com/Gaiwqqq/ActiveDepthSensingForUAV (accessed on 5 October 2026). The supporting data for the simulation, bounded-depth-bias, physical D455 acquisition, and PX4 SIH experiments will be made openly available in the same repository upon publication.

Acknowledgments: During preparation of this revision, the authors used OpenAI Codex (OpenAI) for language editing, organization of reviewer responses, and assistance with scripts for checking experimental records and preparing figures. The authors take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

DAgger	Dataset Aggregation
EMA	Exponential Moving Average
GRU	Gated Recurrent Unit
ROI	Region of Interest
UAV	Unmanned Aerial Vehicle
P/E/G	Projector power/exposure/gain
SIH	Simulation-in-Hardware

Appendix A. Occlusion Layouts and Evaluation Protocol

The same frozen split-stem policy is evaluated across all layouts. Table A1 reports the condition-specific outcomes. Layout selection uses a separate validation set of 20 episodes per regime, ranked by Ours success, then baseline advantage, worst-regime success, and terminal distance. Testing uses 100 independent episodes per regime, paired with the original-scene control. Overall paired success changes for A–D are +5.00, +1.67, +4.33, and +0.67 percentage points, with 95% intervals [2.00, 8.00], [−1.00, 4.33], [0.00, 8.67], and [−1.00, 2.33]. Intervals use 10,000 paired regime-stratified bootstrap resamples, without multiple-comparison correction, and are conditional on the selected layouts.

Table A1. All condition-specific success rates (SR); 100 episodes per cell. Δ: change from the same-regime original control in percentage points (pp). The dark-material decrease for C is retained.

Layout	Glare		Specular Proxy		Dark-Material Proxy	
	SR (%)	Δ (pp)	SR (%)	Δ (pp)	SR (%)	Δ (pp)
Original	65.0	+0.0	76.0	+0.0	83.0	+0.0
A: staggered pair	73.0	+8.0	82.0	+6.0	84.0	+1.0
B: one-sided	67.0	+2.0	79.0	+3.0	83.0	+0.0
C: offset wide	75.0	+10.0	84.0	+8.0	78.0	-5.0
D: stepped	65.0	+0.0	77.0	+1.0	84.0	+1.0

Foreground boxes share scene yaw and remain fixed within an episode. Their lateral coordinates follow the reset-sampled aperture center y_a:

$$y_o = \alpha_o y_a + \delta_o, \qquad \alpha_o = \frac{x_o - x_{\text{start}}}{x_{\text{wall}} - x_{\text{start}}} = \frac{x_o + 1.5}{1.5}. \tag{A1}$$

The offset δ_o distinguishes the layouts. Table A2 gives all box centers and half-extents; the boxes participate in both depth rendering and the original collision queries.

Table A2. Local geometry of the additional occluders. Each box has center $(x_o, \alpha_o y_a + \delta_o, z_o)$ and the listed half-extents. Here y_a is the reset-sampled aperture center. All boxes share the scene yaw; their geometry remains static within an episode.

Layout	Obj.	x_o (m)	α_o	δ_o (m)	z_o (m)	Half-Extents (h_x, h_y, h_z) (m)
A: staggered pair	1	−0.90	0.4000	−0.080	0.600	(0.080, 0.100, 0.600)
A: staggered pair	2	−0.40	0.7333	+0.080	0.575	(0.090, 0.110, 0.575)
B: one-sided	1	−0.75	0.5000	+0.100	0.625	(0.080, 0.100, 0.625)
C: offset wide	1	−0.40	0.7333	+0.070	0.625	(0.100, 0.150, 0.625)
D: stepped	1	−1.10	0.2667	−0.035	0.655	(0.080, 0.065, 0.655)
D: stepped	2	−1.04	0.3067	+0.080	0.635	(0.100, 0.120, 0.635)

Appendix B. Supplementary Hardware Protocols

D455 optical-axis depth is projected to 96 × 72 using nearest-neighbor pinhole mapping, preserving invalid pixels. Motion inputs represent the stationary camera. A physical parameter search supplies the register targets for 300 camera-head imitation updates (Table A3). This adapted shared-stem camera model is separate from the frozen split-stem navigation policy used in SIH. The illustrative 1 ms manual reference is deliberately underexposed; device auto-exposure provides the substantive physical-acquisition comparison.

SIH and its paired pure-simulation evaluation share scene geometry, randomization, and policy weights. The integration configuration is selected in preliminary trials, then evaluated on 60 independent paired episodes. The 44/60 SIH success rate has a 95% Wilson interval [60.99, 82.87]%, conditional on this configuration. PX4 estimates motion from simulated inertial and navigation signals; rendering and outcome evaluation use simulator ground truth. An additional 15 mm clearance test assesses sensitivity to the resolution of this feedback. SIH involves neither physical vehicle motion nor physical depth-camera feedback.

Table A3. Reproducible settings of the two supplementary hardware experiments. Physical and virtual acquisition settings are separate interfaces.

Item	Setting
Physical D455 acquisition	
Camera input/capture	96 × 72 projected depth/640 × 480 raw depth
Policy register ranges	Power: 0–360 SDK units; exposure: 0.5–20 ms; gain: 16–64
Register updates	2 Hz; smoothing coefficient 0.7; power step 30
Camera adaptation	300 camera-head imitation updates; 48 training clips, 12 held-out imitation clips
Hardware teacher target	Power 360; exposure 5 ms; gain 16
Evaluation	Five blocks; three methods; 8 s per clip
PX4 SIH navigation	
Flight controller	Open-source NxtPX4 board; PX4 v1.15.4; onboard SIH dynamics and sensor simulation
Control interface	USB MAVLink; host acceleration and yaw setpoints; PX4 estimated-state feedback
Acquisition/policy	96 × 72 synthetic depth; frozen split-stem policy
Execution	60 actions, including the final interval; per-trajectory exposure-dependent scheduling
Host control adapter	Drag compensation; 50 ms state prediction
Outcome thresholds	Goal tolerance 0.35 m; clearance threshold 1.1 mm; additional 15 mm sensitivity check
Final evaluation	20 episodes per regime; 60 paired SIH/pure-simulation episodes

Appendix C. Bounded-Bias Evaluation Details

The bias test uses the same frozen split-stem policy as Appendix A, selected using unperturbed validation before bias evaluation. Scene geometry is paired across bias levels, while perturbations are sampled independently of scene randomization. The perturbation affects all valid pixels and preserves invalid pixels within each frame. Policy weights remain fixed, and success, collision, and timeout are mutually exclusive outcomes. Table A4 reports every regime; the zero-bias condition serves as the matched reference for degradation under systematic range error.

Table A4. Condition-specific success rates (SR), 100 episodes per cell, and overall mean terminal distance over all outcomes. The 300-episode zero-bias control is distinct from the 900-episodeprimary benchmark.

Bias Bound	Glare SR (%)	Specular SR (%)	Dark SR (%)	Overall Distance (m)
0 m	61	80	83	0.562
0.10 m	62	74	74	0.633
0.30 m	36	41	34	1.138

References

1. Servi, M.; Mussi, E.; Profili, A.; Furferi, R.; Volpe, Y.; Governi, L.; Buonamici, F. Metrological Characterization and Comparison of D415, D455, L515 RealSense Devices in the Close Range. *Sensors* **2021**, *21*, 7770. [CrossRef]
2. Jang, M.; Yoon, H.; Lee, S.; Kang, J.; Lee, S. A Comparison and Evaluation of Stereo Matching on Active Stereo Images. *Sensors* **2022**, *22*, 3332. [CrossRef]
3. Song, Y.; Shi, K.; Penicka, R.; Scaramuzza, D. Learning Perception-Aware Agile Flight in Cluttered Environments. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2023; pp. 1989–1995. [CrossRef]
4. Tordesillas, J.; How, J.P. Deep-PANTHER: Learning-Based Perception-Aware Trajectory Planner in Dynamic Environments. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1399–1406. [CrossRef]
5. Zhao, G.; Wu, T.; Chen, Y.; Gao, F. Learning Speed Adaptation for Flight in Clutter. *IEEE Robot. Autom. Lett.* **2024**, *9*, 7222–7229. [CrossRef]
6. Bégin, M.A.; Hunter, I. Auto-Exposure Algorithm for Enhanced Mobile Robot Localization in Challenging Light Conditions. *Sensors* **2022**, *22*, 835. [CrossRef]
7. Yan, C.; Dansereau, D.G. TaCOS: Task-Specific Camera Optimization with Simulation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*; IEEE: New York, NY, USA, 2025; pp. 2052–2062.
8. Yan, C.; Bryson, M.; Dansereau, D.G. JOCA: Task-Driven Joint Optimisation of Camera Hardware and Adaptive Camera Control Algorithms. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*; IEEE: New York, NY, USA, 2026; pp. 3928–3938.
9. Hsieh, T.L.; Jhan, Z.S.; Yeh, N.J.; Chen, C.Y.; Chuang, C.T. An Unmanned Aerial Vehicle Indoor Low-Computation Navigation Method Based on Vision and Deep Learning. *Sensors* **2024**, *24*, 190. [CrossRef]
10. Łuczak, P.; Granosik, G. Autonomous UAV Landing and Collision Avoidance System for Unknown Terrain Utilizing Depth Camera with Actively Actuated Gimbal. *Sensors* **2025**, *25*, 6165. [CrossRef]
11. Qayyum, U.; Kim, J. Depth-Camera-Aided Inertial Navigation Utilizing Directional Constraints. *Sensors* **2021**, *21*, 5913. [CrossRef]
12. Bajcsy, R. Active Perception. *Proc. IEEE* **1988**, *76*, 996–1005. [CrossRef]
13. Aloimonos, J.; Weiss, I.; Bandyopadhyay, A. Active Vision. *Int. J. Comput. Vis.* **1988**, *1*, 333–356. [CrossRef]
14. Bajcsy, R.; Aloimonos, Y.; Tsotsos, J.K. Revisiting Active Perception. *Auton. Robot.* **2018**, *42*, 177–196. [CrossRef]
15. Florence, P.; Carter, J.; Tedrake, R. Integrated Perception and Control at High Speed: Evaluating Collision Avoidance Maneuvers Without Maps. In *Algorithmic Foundations of Robotics XII*; Goldberg, K., Abbeel, P., Bekris, K., Miller, L., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 13, pp. 304–319. [CrossRef]
16. Hu, Y.; Zhang, Y.; Song, Y.; Deng, Y.; Yu, F.; Zhang, L.; Lin, W.; Zou, D.; Yu, W. Seeing Through Pixel Motion: Learning Obstacle Avoidance from Optical Flow with One Camera. *IEEE Robot. Autom. Lett.* **2025**, *10*, 5871–5878. [CrossRef]
17. Loquercio, A.; Kaufmann, E.; Ranftl, R.; Müller, M.; Koltun, V.; Scaramuzza, D. Learning High-Speed Flight in the Wild. *Sci. Robot.* **2021**, *6*, eabg5810. [CrossRef]
18. Foehn, P.; Kaufmann, E.; Romero, A.; Penicka, R.; Sun, S.; Bauersfeld, L.; Laengle, T.; Cioffi, G.; Song, Y.; Loquercio, A.; et al. Agilicious: Open-Source and Open-Hardware Agile Quadrotor for Vision-Based Flight. *Sci. Robot.* **2022**, *7*, eabl6259. [CrossRef]
19. Kaufmann, E.; Bauersfeld, L.; Loquercio, A.; Müller, M.; Koltun, V.; Scaramuzza, D. Champion-Level Drone Racing Using Deep Reinforcement Learning. *Nature* **2023**, *620*, 982–987. [CrossRef]
20. Newbury, R.; Collins, J.; He, K.; Pan, J.; Posner, I.; Howard, D.; Cosgun, A. A Review of Differentiable Simulators. *IEEE Access* **2024**, *12*, 97581–97604. [CrossRef]
21. Zhang, X.; Wang, R.; Ren, Y.; Sun, J.; Fang, H.; Chen, J.; Wang, G. DiffAero: A GPU-Accelerated Differentiable Simulation Framework for Efficient Quadrotor Policy Learning. *arXiv* **2025**, arXiv:2509.10247. [CrossRef]
22. Zhang, Y.; Hu, Y.; Song, Y.; Zou, D.; Lin, W. Learning Vision-Based Agile Flight via Differentiable Physics. *Nat. Mach. Intell.* **2025**, *7*, 954–966. [CrossRef]

23. Heeg, J.; Song, Y.; Scaramuzza, D. Learning Quadrotor Control from Visual Features Using Differentiable Simulation. In *Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2025; pp. 4033–4039. [CrossRef]
24. Romero, A.; Aljalbout, E.; Song, Y.; Scaramuzza, D. Actor–Critic Model Predictive Control: Differentiable Optimization Meets Reinforcement Learning for Agile Flight. *IEEE Trans. Robot.* **2026**, *42*, 673–692. [CrossRef]
25. Denninger, M.; Winkelbauer, D.; Sundermeyer, M.; Boerdijk, W.; Knauer, M.; Strobl, K.H.; Humt, M.; Triebel, R. BlenderProc2: A Procedural Pipeline for Photorealistic Rendering. *J. Open Source Softw.* **2023**, *8*, 4901. [CrossRef]
26. Amini, A.; Wang, T.H.; Gilitschenski, I.; Schwarting, W.; Liu, Z.; Han, S.; Karaman, S.; Rus, D. VISTA 2.0: An Open, Data-Driven Simulator for Multimodal Sensing and Policy Learning for Autonomous Vehicles. In *Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2022; pp. 2419–2426. [CrossRef]
27. Puig, X.; Undersander, E.; Szot, A.; Dallaire Cote, M.; Yang, T.Y.; Partsey, R.; Desai, R.; Clegg, A.W.; Hlavac, M.; Min, S.Y.; et al. Habitat 3.0: A Co-Habitat for Humans, Avatars, and Robots. In *Proceedings of the International Conference on Learning Representations (ICLR)*; ICLR: Appleton, WI, USA, 2024.
28. Elmquist, A.; Serban, R.; Negrut, D. A Sensor Simulation Framework for Training and Testing Robots and Autonomous Vehicles. *J. Auton. Veh. Syst.* **2021**, *1*, 021001. [CrossRef]
29. Blasinski, H.; Farrell, J.; Lian, T.; Liu, Z.; Wandell, B. Optimizing Image Acquisition Systems for Autonomous Driving. *Electron. Imaging* **2018**, *2018*, art00002. [CrossRef]
30. Liu, Z.; Lian, T.; Farrell, J.; Wandell, B.A. Neural Network Generalization: The Impact of Camera Parameters. *IEEE Access* **2020**, *8*, 10443–10454. [CrossRef]
31. Lyu, Z.; Goossens, T.; Wandell, B.A.; Farrell, J. Validation of Physics-Based Image Systems Simulation with 3-D Scenes. *IEEE Sens. J.* **2022**, *22*, 19400–19410. [CrossRef]
32. Carlson, A.; Skinner, K.A.; Vasudevan, R.; Johnson-Roberson, M. Modeling Camera Effects to Improve Visual Learning from Synthetic Data. In *Proceedings of the Computer Vision—ECCV 2018 Workshops, Cham, 2019*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11129, pp. 505–520. [CrossRef]
33. Carlson, A.; Skinner, K.A.; Vasudevan, R.; Johnson-Roberson, M. Sensor Transfer: Learning Optimal Sensor Effect Image Augmentation for Sim-to-Real Domain Adaptation. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2431–2438. [CrossRef]
34. Ouyang, H.; Shi, Z.; Lei, C.; Law, K.L.; Chen, Q. Neural Camera Simulators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2021; pp. 7696–7705. [CrossRef]
35. Chen, B.H.; Batagoda, N.M.; Negrut, D. DiffPhysCam: Differentiable Physics-Based Camera Simulation for Inverse Rendering and Embodied AI. *arXiv* **2025**, arXiv:cs.GR/2508.08831. [CrossRef]
36. Ross, S.; Gordon, G.; Bagnell, D. 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, Fort Lauderdale, FL, USA, 11–13 April 2011*; PMLR: Cambridge, MA, USA, 2011; Volume 15, pp. 627–635. Available online: https://proceedings.mlr.press/v15/ross11a.html (accessed on 13 September 2026).

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