Research Square

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

A Machine-Learning and Cybernetic Control Framework for RGB--Geometric UAV Navigation

Minghu Zhang

zhangmh@lut.edu.cn

Lanzhou University of Technology

Yulin Cao

Lanzhou University of Technology

Enzhan Zhang

Lanzhou University of Technology

Mingxuan Xu

Research Article

Keywords: machine learning, cybernetic control, UAV navigation, RGB--geometric fusion, meta-gradient reinforcement learning, hybrid control

Posted Date: October 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11140604/v1

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

Additional Declarations: No competing interests reported.

A Machine-Learning and Cybernetic Control Framework for RGB–Geometric UAV Navigation

Abstract

Cybernetic navigation of an unmanned aerial vehicle (UAV) requires visual perception, structured feedback, adaptive rewards, and safety intervention within a unified perception–decision–action loop. We present a river-corridor navigation framework that fuses red–green–blue (RGB) observations encoded by a residual network and feature pyramid network (FPN) with kinematic, target, centreline, clearance, and memory variables, and optimises the controller using twin delayed deep deterministic policy gradient (TD3). Replay retains six reward components so that sampled transitions can be reweighted consistently. A differentiable meta-gradient links virtual actor–critic updates to one-step validation kinematics. In fixed-width ablations, the vision-plus-kinematics condition retained 84.0% completion, supporting the use of RGB information by the policy. Removing memory produced the largest observed decrease (4.25 percentage points), although no structured-state contrast was statistically significant and the no-target and no-centreline point estimates slightly exceeded the full-state result. Meta-gradient variants improved on fixed weights but did not outperform the projection-matched zero-hypergradient control, so the gain is attributed to projected reward reconfiguration rather than to the hypergradient alone. On a five-seed river-event benchmark, the hybrid TD3 controller achieved 88.5% mean task completion, where completion comprises goal arrival, successful rerouting, or a protocol-valid safe stop; this exceeded nonlinear model predictive control by 11.1 percentage points and rapidly exploring random tree star replanning (RRT*-Replan) by 15.9 points.

Keywords: machine learning, cybernetic control, UAV navigation, RGB–geometric fusion, meta-gradient reinforcement learning, hybrid control

1 Introduction

Unmanned aerial vehicles (UAVs) can acquire high-resolution observations along river reaches that are difficult to survey from fixed stations or by manual inspection (Acharya et al. 2021; Liu 2023; Manfreda et al. 2024; Yadav et al. 2024). Navigation in these corridors is nevertheless difficult: water reflections alter appearance, narrow bends restrict manoeuvring space, shoreline interruptions weaken geometric cues, and bridges or floating objects create local occlusion and collision risk. A useful controller

must therefore couple visual interpretation with metric progress and clearance rather than treating perception and motion as independent tasks.

Model-based planners provide interpretable trajectories when maps, obstacle states, and dynamics are sufficiently accurate (Fan et al. 2022; Song and Scaramuzza 2022; Tordesillas and How 2022). Learning-based controllers can react directly to observations and have demonstrated high-speed, wind-adaptive, and differentiable-physics flight in demanding aerial environments (Loquercio et al. 2021; O'Connell et al. 2022; Zhang et al. 2025). Their conclusions nevertheless depend strongly on what the policy actually receives. A network described as visual is not visually validated if its image tensor is replaced by a deterministic projection of privileged simulator variables. Likewise, an ablation does not isolate a structured input when the same information remains encoded through a surrogate feature vector. These distinctions are especially important for river navigation, where a centreline, signed-distance field, or manually corrected mask may be available during simulation analysis but unavailable during deployment.

We address this problem as a machine-learning and cybernetic control system with four separately verifiable interfaces. The visual interface consumes the current rendered RGB frame and computes a trainable multi-scale representation with ResNet–50 and a feature pyramid network (FPN). The feedback-state interface contains only explicitly named kinematic, target, centreline, ray-clearance, and memory variables. The reward interface stores the unweighted component vector, allowing every sampled reward to be reconstructed under the current weights. The intervention interface maps a learned policy action to an executed action only during a separate frozen-policy evaluation and records both signals. These interfaces close the perception–decision–action–feedback loop while preventing visual evidence, privileged geometry, reward adaptation, and engineered intervention from being conflated.

The contributions are as follows:

1. We formulate a 291-dimensional feedback state in which an online RGB/FPN representation is concatenated with 35 fully defined geometric and kinematic variables. Global pooling supplies appearance context, while exact metric feedback remains explicit.
2. We close the adaptive-reward computational graph mathematically and algorithmically. Replay stores six dimensionless components rather than a stale scalar reward, and the outer loss depends on a virtual actor action through explicit differentiable kinematics and local geometry.
3. We define leakage-resistant, fixed-width ablations for target, centreline, clearance, and memory inputs. Removing a structured block replaces only that block with zeros and never inserts a surrogate for the omitted variable. The vision-plus-kinematics condition provides outcome-level evidence that the trained policy uses the RGB/FPN stream.
4. We separate learned control, predictive optimisation, and rule intervention. The policy action and executed action are logged independently, compatible controllers receive the same frozen rule map, and paired event tables quantify benefits and side effects.

5. We provide a reproducible evaluation protocol with frozen balanced manifests, equal interaction budgets, paired statistics, finite-difference gradient checks, field-file provenance, and distinct evidence for general geometry control and river-specific events.

2 Related work

2.1 River observation and geometric planning

UAV imagery is widely used for river mapping, hydrological monitoring, water-quality assessment, and flood reconnaissance (Acharya et al. 2021; Kwon et al. 2023; Liu 2023; Manfreda et al. 2024; Yadav et al. 2024). Water-mask and morphology extraction methods demonstrate that river geometry can be inferred from imagery (Moortgat et al. 2022; Valman et al. 2024), while river-monitoring studies document the effects of glare, weak texture, and transient surface patterns (Manfreda et al. 2024). Recent drone-video velocimetry has advanced multi-scale local–global optical flow, recurrent feature encoding, lightweight onboard inference, physics-constrained flow estimation, real-time image-velocimetry systems, and reinforcement-learning-based feature enhancement (Zhang et al. 2026a,b,c,d,e,f). These studies provide direct evidence that natural river imagery contains scale-dependent motion and texture cues, but their outputs are surface-velocity measurements rather than navigation actions. We therefore use them to motivate the visual representation and field-domain analysis, not as navigation-validation evidence.

Graph search, sampling-based planning, and model predictive control are attractive because constraints and trajectories remain inspectable (Fan et al. 2022; Song and Scaramuzza 2022; Tordesillas and How 2022). Their accuracy depends on a reliable map and obstacle model. In a visually unstable river corridor, uncertainty in shoreline or obstacle geometry propagates into the planned clearance. We therefore retain NMPC as a strong geometry-based comparison rather than presenting learned control only against weak static planners.

2.2 Visual representation and continuous control

Continuous-control reinforcement learning has been used for waypoint following, collision avoidance, and agile flight (AlMahamid and Grolinger 2022; Kaufmann et al. 2023; Penicka et al. 2022; Song et al. 2023; Xu et al. 2025; Yu et al. 2025). Perception-aware agile flight further couples obstacle sensing to time-critical control (Song et al. 2023). End-to-end visual flight has also been demonstrated with onboard sensing in cluttered outdoor environments, while open hardware and software platforms make perception–control and hardware-in-the-loop experiments reproducible (Foehn et al. 2022; Loquercio et al. 2021). Digital-twin and multi-objective formulations further show how environment uncertainty, mobile targets, energy, and wind can be incorporated into UAV policy evaluation (Soliman et al. 2023; Wu et al. 2024). DDPG learns a deterministic actor with an off-policy critic (Lillicrap et al. 2016). TD3 reduces overestimation using clipped double-Q targets, target-policy smoothing, and delayed actor

updates (Fujimoto et al. 2018). SAC provides an entropy-regularised off-policy alternative (Haarnoja et al. 2018), whereas PPO is an on-policy clipped-surrogate baseline (Schulman et al. 2017). These algorithms address optimisation, not the validity of the observation interface; fair comparison therefore requires identical observations, action bounds, interaction budgets, and scenario manifests.

Feature pyramids combine deep semantic features with higher-resolution spatial features and are well suited to objects that occur at markedly different scales (Lin et al. 2017). Global pooling of each scale produces compact appearance evidence but cannot preserve exact pixel coordinates. Robust navigation therefore commonly combines complementary sensor or state sources, and dynamic-object pipelines explicitly separate visual detection from metric tracking (Xu et al. 2024; Zhuang et al. 2023). Our state design assigns similarly distinct roles to the two streams: RGB/FPN represents appearance, whereas target direction, local centreline samples, and clearance rays provide metric control context. Removing one stream does not replace it with a deterministic projection of the other.

2.3 Adaptive rewards and hybrid safety layers

Reward weights often encode competing objectives such as progress, clearance, smoothness, and terminal outcome. Meta-gradient reinforcement learning differentiates a validation objective through an inner optimisation step to adapt such hyperparameters (Agarwal et al. 2019; Xu et al. 2018). Reproducibility requires the computational graph, stop-gradient boundaries, validation records, and replay contents to be stated explicitly. In particular, a buffer containing only rewards computed under historical weights is incompatible with retrospective reweighting.

Safety shields and action projections can reduce unsafe behaviour outside the distribution represented by a learned policy (Brunke et al. 2022; Hsu et al. 2023; Xu et al. 2025). They can also introduce new failures or reduce task completion. A hybrid system should therefore record the raw and executed actions and compare the same frozen checkpoint with and without intervention. We treat the rule layer as a separate engineering component, not as a formal safety guarantee or an implicit part of policy learning.

3 Method

3.1 Problem formulation and cybernetic system boundary

The UAV moves in a horizontal river-centred plane at fixed altitude. Its planar position is $p_t = [p_{x,t}, p_{y,t}]^\top$. At control step t, the policy receives a current-view RGB image $I_t \in \mathbb{R}^{3\times224\times224}$ and a structured vector $x_t \in \mathbb{R}^{35}$. It outputs forward speed and yaw rate,

$$a_t^\pi = \begin{bmatrix} v_t & \omega_t \end{bmatrix}^\top, \qquad 0 \le v_t \le v_{\max}, \quad |\omega_t| \le \omega_{\max}. \tag{1}$$

During policy training, $a_t^{\text{exec}} = a_t^\pi$. During the separate rule evaluation, a frozen rule map produces $a_t^{\text{exec}} = \mathcal{S}(h_t, a_t^\pi)$, where h_t is an intervention trigger vector that is not

part of the actor state. The simulator then advances with a unicycle model,

$$\begin{aligned} p_{x,t+1} &= p_{x,t} + \Delta t\, v_t^{\text{exec}} \cos\psi_t, \\ p_{y,t+1} &= p_{y,t} + \Delta t\, v_t^{\text{exec}} \sin\psi_t, \\ \psi_{t+1} &= \text{wrap}(\psi_t + \Delta t\, \omega_t^{\text{exec}}). \end{aligned} \tag{2}$$

Wind, acceleration dynamics, actuator delay, and sensor latency are excluded from this planar model. The scope is therefore closed-loop navigation in a procedural visual environment, not flight-dynamics certification. Figure 1 summarises this system boundary.

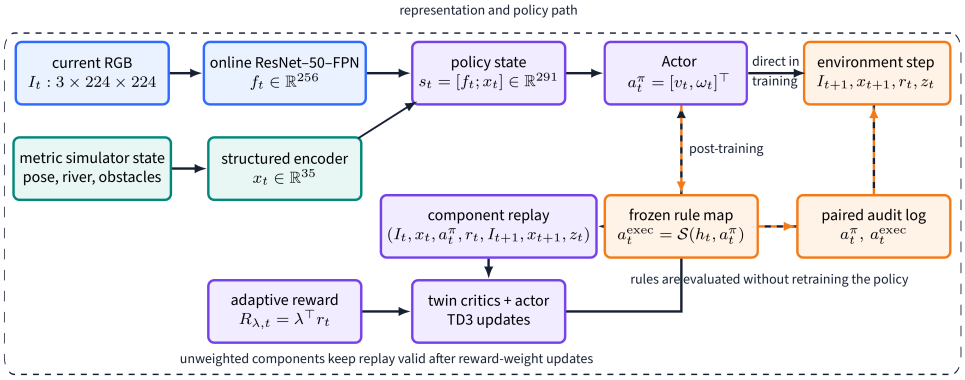


Fig. 1 System boundary. The training path executes the policy action directly and stores unweighted reward components. The rule layer is activated only in a paired frozen-policy evaluation; both the policy and executed actions are logged

3.2 Online RGB/FPN representation

The renderer forms I_t from the current UAV pose, local bank geometry, visible obstacles, and appearance nuisance variables. The goal coordinate, centreline samples, clearance values, memory flags, and their numeric encodings are not painted as overlays or extra channels. RGB normalisation uses ImageNet channel means and standard deviations. Training perturbations include brightness and contrast changes, specular patches, haze, partial shadows, motion blur, and small camera-pose jitter; the held-out manifests specify their ranges independently from the policy state.

Figure 2 shows the visual path. For a 224×224 input, a ResNet–50 backbone produces $C_2 \in \mathbb{R}^{256\times56\times56}$, $C_3 \in \mathbb{R}^{512\times28\times28}$, $C_4 \in \mathbb{R}^{1024\times14\times14}$, and $C_5 \in \mathbb{R}^{2048\times7\times7}$. Lateral 1×1 convolutions and top-down bilinear upsampling form P_2 through P_5, each with 256 channels. For $l \in \{2, 3, 4, 5\}$,

$$u_{l,t} = \text{GAP}(P_{l,t}), \qquad \boldsymbol{\omega}_t = \text{softmax}(W_2\sigma(W_1[u_{2,t}; \ldots; u_{5,t}] + b_1) + b_2)\,, \tag{3}$$

and the appearance vector is

$$f_t = \sum_{l=2}^{5} \omega_{l,t} u_{l,t} \in \mathbb{R}^{256}, \qquad \sum_{l=2}^{5} \omega_{l,t} = 1. \tag{4}$$

Here $W_1 \in \mathbb{R}^{128\times1024}$ and $W_2 \in \mathbb{R}^{4\times128}$; the final attention bias is initialised to zero so the four scales begin uniformly. The backbone and pyramid are evaluated inside the policy computation graph. During critic updates, the sampled visual latent is detached from the encoder; during actor updates, the action gradient updates both the actor head and visual encoder. The ImageNet-initialised backbone uses a learning rate one tenth of the actor-head rate. Target features are produced by the corresponding slowly updated target encoder. This avoids a fixed random or geometry-derived surrogate while preventing a critic step from changing the policy representation through two competing optimisers.

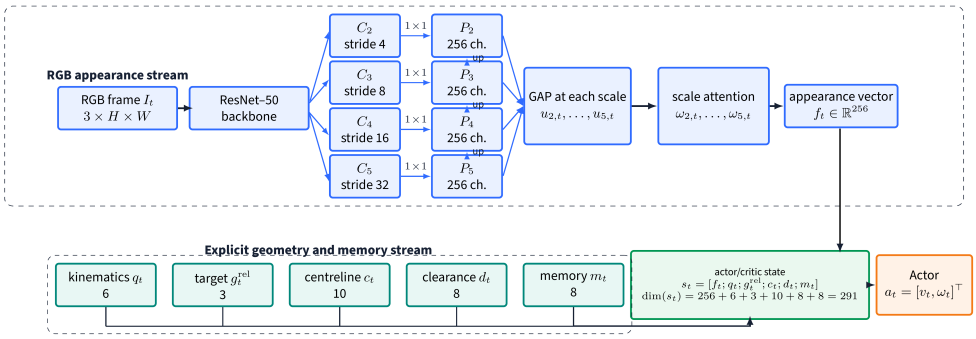


Fig. 2 Policy-state construction. Multi-scale RGB features provide appearance evidence; separately computed relative kinematics, target/progress, centreline, ray-clearance, and memory variables provide exact metric context

3.3 Structured state and leakage-resistant ablations

All structured quantities are expressed in a UAV-centred frame and normalised by sensing radius R_s, speed limit $v_{\max}$, or their stated physical bounds. The six-dimensional relative-kinematic block is

$$q_t = [v_{x,t}/v_{\max}, v_{y,t}/v_{\max}, \sin\psi_t, \cos\psi_t, \dot{v}_t/a_{\max}, \dot{\omega}_t/\alpha_{\max}]^\top \in \mathbb{R}^6, \tag{5}$$

where translational and angular accelerations use separate physical limits. This block contains no target, centreline, clearance, event, or historical reference variable. The target/progress block is

$$g_t^{\mathrm{rel}} = \left[\frac{\Delta x_t^g}{R_s}, \frac{\Delta y_t^g}{R_s}, \mathrm{clip}\left(\frac{\rho_t}{R_s}, 0, 1\right)\right]^\top \in \mathbb{R}^3, \tag{6}$$

where ρ_t is remaining centreline arc length. Five forward centreline samples form

$$c_t = \left[\frac{\Delta x^c_{1,t}}{R_s}, \frac{\Delta y^c_{1,t}}{R_s}, \ldots, \frac{\Delta x^c_{5,t}}{R_s}, \frac{\Delta y^c_{5,t}}{R_s} \right]^\top \in \mathbb{R}^{10}. \tag{7}$$

Clearance is obtained by ray casting in eight body-frame directions. If $\ell_{j,t}$ is the first intersection distance with a bank or obstacle along ray j, then

$$d_t = \left[\mathrm{clip}\left(\frac{\ell_{j,t}}{R_s}, 0, 1 \right) \right]_{j=0}^{7} \in \mathbb{R}^8. \tag{8}$$

The memory block contains only the most recent unoccluded pose and motion record,

$$m_t = [\Delta x^m_t / R_s, \Delta y^m_t / R_s, \sin \Delta\psi^m_t, \cos \Delta\psi^m_t, v^m_t / v_{\max}, \omega^m_t / \omega_{\max}, \tilde{\tau}^m_t, \chi^m_t]^\top \in \mathbb{R}^8, \tag{9}$$

where $\tilde{\tau}^m_t$ is the clipped age of the record and χ^m_t is its visibility confidence. Bifurcation, termination, occlusion, and aggregation triggers are excluded from the actor state and are logged only during the separate rule evaluation. The actor state is therefore

$$s_t = [f_t; q_t; g^{\mathrm{rel}}_t; c_t; d_t; m_t] \in \mathbb{R}^{291}. \tag{10}$$

Each confirmatory structured-state ablation preserves the 291-dimensional interface. Removing target, centreline, clearance, or memory replaces only that named block with zeros before concatenation. The remaining blocks contain no deterministic copy or descendant of the removed source: in particular, kinematics excludes cross-track error, memory excludes current event flags, and rule triggers are never exposed to the actor. RGB/FPN forward, gradient, and optimiser-update gates are evaluated separately and are not used as substitutes for the structured-state ablation. No state condition introduces a projection of the omitted simulator variable into another block.

3.4 TD3 learning core

The main deterministic controller follows TD3 (Fujimoto et al. 2018). The actor parameters θ include the online visual encoder and actor head; ϕ_1, ϕ_2 denote the two critics. Replay stores raw observations and reward components,

$$\mathcal{D} \ni (I_t, x_t, a^\pi_t, r_t, I_{t+1}, x_{t+1}, z_t), \qquad r_t \in [-1, 1]^6, \tag{11}$$

where $z_t = 1$ at terminal transitions. Consequently, current weights can be applied to every historical transition. For target-policy noise $\epsilon \sim \mathrm{clip}(\mathcal{N}(0, \sigma^2), -c, c)$,

$$a'_{t+1} = \mathrm{clip}(\mu_{\bar{\theta}}(s_{t+1}) + \epsilon, a_{\min}, a_{\max}), \tag{12}$$

$$y_t(\lambda) = \lambda^\top r_t + \gamma(1 - z_t) \min_{i \in \{1,2\}} Q_{\bar{\phi}_i}(s_{t+1}, a'_{t+1}), \tag{13}$$

and each critic minimises

$$\mathcal{L}_{Q_i}(\phi_i;\lambda) = \mathbb{E}_{\mathcal{D}}\left[\left(Q_{\phi_i}(s_t,a_t^{\pi}) - y_t(\lambda)\right)^2\right]. \tag{14}$$

The actor is updated every two critic steps using the minimisation loss

$$\mathcal{L}_{\mu}(\theta) = -\mathbb{E}_{s_t\sim\mathcal{D}}\left[Q_{\phi_1}(s_t,\mu_\theta(s_t))\right]. \tag{15}$$

Target parameters use Polyak averaging, $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$, and likewise for both critics.

3.5 Dimensionless reward components

All reward components are bounded before weighting. Let $\Delta\rho_t = \rho_t - \rho_{t+1}$, $\ell_t^{\min}$ be the minimum obstacle or bank clearance, δ_{safe} the desired clearance, and $z_t = 1$ at a terminal transition. We define

$$\begin{aligned}
r_t^{\text{prog}} &= \operatorname{clip}\left(\frac{\Delta\rho_t}{v_{\max}\Delta t}, -1, 1\right),\\
r_t^{\text{safe}} &= -\operatorname{clip}\left(\frac{\delta_{\text{safe}} - \ell_t^{\min}}{\delta_{\text{safe}}}, 0, 1\right),\\
r_t^{\text{term}} &= \mathbb{1}_{\text{goal}} - \mathbb{1}_{\text{collision or boundary}},\\
r_t^{\text{smooth}} &= -\frac{1}{2}\left[\operatorname{clip}\left(\frac{|v_t - v_{t-1}|}{v_{\max}}, 0, 1\right) + \operatorname{clip}\left(\frac{|\omega_t - \omega_{t-1}|}{\omega_{\max}}, 0, 1\right)\right],\\
r_t^{\text{head}} &= \cos(\psi_t - \psi_t^{c}),\\
r_t^{\text{step}} &= -(1 - z_t).
\end{aligned} \tag{16}$$

The step component is therefore a per-transition time penalty that vanishes on terminal records. Speed change and yaw-rate change are normalised separately, so different physical units are not mixed through a common maximum. With $r_t = [r_t^{\text{prog}}, r_t^{\text{safe}}, r_t^{\text{term}}, r_t^{\text{smooth}}, r_t^{\text{head}}, r_t^{\text{step}}]^\top$, matching the labels and order in Fig. 8, the scalar reward is

$$R_{\lambda,t} = \lambda^\top r_t, \qquad \lambda \geq 0, \qquad \mathbf{1}^\top\lambda = 1. \tag{17}$$

3.6 Complete meta-gradient computational graph

The outer objective is a smooth one-step control objective, not discrete success rate. A validation record contains $s_t, p_t, \psi_t, g_t, \bar{p}_t, n_t, w_t, \mathcal{O}_t, a_{t-1}$, where g_t is the absolute goal position associated with the relative block g_t^{rel}, $\bar{p}_t$, n_t, and w_t are the nearest centreline point, normal, and local half-width, and $\mathcal{O}_t = \{o_{k,t}\}_{k=1}^{K_t}$ contains the local circular obstacles with $o_{k,t} = (o_{k,t}^c, o_{k,t}^r)$. The virtual action is

$$\widetilde{a}_t = \mu_{\theta^+(\lambda)}(s_t) = [\widetilde{v}_t, \widetilde{\omega}_t]^\top. \tag{18}$$

It enters explicit differentiable one-step kinematics,

$$\widehat{p}_{t+1} = p_t + \Delta t\, \widetilde{v}_t[\cos\psi_t, \sin\psi_t]^\top, \qquad \widehat{\psi}_{t+1} = \mathrm{wrap}(\psi_t + \Delta t\, \widetilde{\omega}_t). \tag{19}$$

The local cross-track proxy is $\widehat{e}_{t+1} = n_t^\top(\widehat{p}_{t+1} - \bar{p}_t)$. Obstacle clearance uses a smooth minimum when $K_t > 0$,

$$\widehat{d}^{\mathrm{obs}}_{t+1} = -\xi \log \sum_k \exp\left[-\frac{\|\widehat{p}_{t+1} - o^c_{k,t}\|_2 - o^r_{k,t}}{\xi}\right]. \tag{20}$$

and is defined as $\widehat{d}^{\mathrm{obs}}_{t+1} = R_s$ when $K_t = 0$. Thus the obstacle term remains finite and contributes a negligible penalty in an obstacle-free neighbourhood. The validation loss is

$$\begin{aligned}\mathcal{L}_{\mathrm{val}} = \kappa_g \frac{\|\widehat{p}_{t+1} - g_t\|_2^2}{R_s^2} + \kappa_c \frac{\widehat{e}^2_{t+1}}{w_t^2} \\ + \kappa_o\, \mathrm{softplus}\left(\frac{\delta_{\mathrm{safe}} - \widehat{d}^{\mathrm{obs}}_{t+1}}{\xi}\right)^2 \\ + \frac{\kappa_u}{2}\left[\left(\frac{\widetilde{v}_t - v_{t-1}}{v_{\max}}\right)^2 + \left(\frac{\widetilde{\omega}_t - \omega_{t-1}}{\omega_{\max}}\right)^2\right].\end{aligned} \tag{21}$$

The constants are fixed to $(\kappa_g, \kappa_c, \kappa_o, \kappa_u) = (1.0, 1.0, 2.0, 0.1)$, with smooth-min temperature $\xi = 0.25$ m, and are identical across seeds.

For training mini-batch $\mathcal{B}_{\mathrm{tr}}$, both virtual critics take one differentiable step,

$$\phi_i^+(\lambda) = \phi_i - \alpha_Q \nabla_{\phi_i} \mathcal{L}_{Q_i}(\phi_i; \lambda), \qquad i \in \{1, 2\}, \tag{22}$$

followed by one virtual actor step,

$$\theta^+(\lambda) = \theta - \alpha_\mu \nabla_\theta \left[-\mathbb{E}_{s \in \mathcal{B}_{\mathrm{tr}}} Q_{\phi_1^+(\lambda)}(s, \mu_\theta(s))\right]. \tag{23}$$

The hypergradient is

$$\nabla_\lambda \mathcal{L}_{\mathrm{val}} = \left(\frac{\partial \theta^+}{\partial \lambda}\right)^\top \nabla_{\theta^+} \mathcal{L}_{\mathrm{val}}. \tag{24}$$

Thus the full differentiable path is

$$\lambda \to y_t(\lambda) \to \phi^+_{1:2}(\lambda) \to \theta^+(\lambda) \to \widetilde{a}_t \to (\widehat{p}_{t+1}, \widehat{\psi}_{t+1}) \to \mathcal{L}_{\mathrm{val}}. \tag{25}$$

The virtual actor step includes both the RGB encoder and the action head; its backbone parameter group uses the same one-tenth learning-rate ratio as the actual actor update. The inner graph is retained through Eqs. (22)–(23). Replay indices, recorded observations, target-network outputs, and local validation geometry use stop-gradient. No gradient passes through sampled replay transitions or the simulator. Off-policy

dependence enters only through the sampled training records and current differentiable network updates. The weights are updated by

$$\lambda \leftarrow \Pi_{\Delta^5(b_t)}(\lambda - \beta \nabla_\lambda \mathcal{L}_{\mathrm{val}}), \qquad b_t = [0, 0.35\eta_t, 0, 0, 0, 0]^\top, \\ \Delta^5(b_t) = \{u \in \mathbb{R}^6 : u \geq b_t, \mathbf{1}^\top u = 1\}. \tag{26}$$

Here η_t changes irreversibly from zero to one after the first registered validation safety violation. This explains why the five-seed mean in Fig. 8 approaches 0.35 as different seeds activate the floor at different meta-updates. Projection subtracts the unique threshold ν satisfying $\sum_i \max(y_i - b_{i,t} - \nu, 0) = 1 - \mathbf{1}^\top b_t$, and returns $\lambda_i = b_{i,t} + \max(y_i - b_{i,t} - \nu, 0)$. The 0.35 value is therefore a conditional lower bound on the safety reward weight, not an artificial-potential-field action constraint. Figure 3 makes the retained and stopped paths explicit.

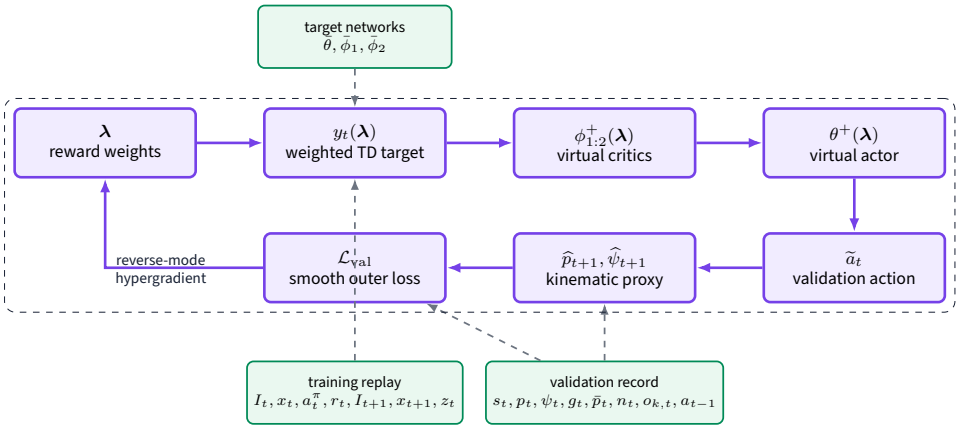


Fig. 3 Meta-gradient computational graph. Purple arrows are differentiated; dashed arrows provide fixed records or target values. The validation action affects the outer loss through explicit kinematics and analytic local geometry

3.7 Training algorithm and validation cache

The reported state, reward, and rule experiments use 256 validation records per seed and a meta-validation batch of 64. These values are the ones recorded by the submitted run logs and are used for Table 7. The fixed-validation condition constructs one seed-specific 256-record cache from the frozen validation manifest and traverses it without replacement in 64-record batches, reshuffling only after all records have been used. The refreshed-validation condition regenerates all 256 records from the same disjoint manifest immediately before each meta-update and then samples 64 records without replacement. Neither condition uses training or test scenarios for the outer objective. Meta-updates occur every $K_{\mathrm{meta}} = 50$ actor updates, use $\beta = 10^{-3}$, and clip the

reward-weight gradient norm at 5. The weight vector, hypergradient norm, validation loss, sampled validation identifiers, and active lower-bound coordinates are logged at every update. The accompanying confirmatory configuration uses the same 64/256/50 values.

Step 1: Render the current RGB frame, compute the structured state, and form s_t with the online FPN.

Step 2: Sample exploration noise, execute a_t^π directly, and compute the six bounded reward components.

Step 3: Store the raw current and next observations, policy action, reward-component vector, and terminal flag in replay.

Step 4: Sample replay, reconstruct $R_{\lambda,t}$, and update both critics; update the actor and target networks on the delayed TD3 schedule.

Step 5: At the meta schedule, form virtual critic and actor steps, evaluate Eq. (21) on the disjoint validation batch, project Eq. (26), and write the complete reward-weight log.

3.8 Intervention layer

Four deterministic rules address bifurcation, detected termination, severe occlusion, and obstacle aggregation. Their trigger vector h_t is computed from current metric geometry with frozen thresholds, is logged separately, and is unavailable to the trained actor. The rules have no access to test labels. Bifurcation selects the branch whose forward tangent minimises target-heading error. Termination commands a monotonic braking action,

$$\begin{aligned} v_t^{\text{exec}} &= \max\big(0, \min(v_t^\pi, v_{\text{cap}}, v_{t-1}^{\text{exec}} - a_{\text{brake}}\Delta t)\big)\,, \\ \omega_t^{\text{exec}} &= \text{clip}(k_\psi \Delta\psi_t^m, -\omega_{\max}, \omega_{\max}), \end{aligned} \tag{27}$$

where $v_{\text{cap}} = 3$ m s^{-1} and the memory-heading error $\Delta\psi_t^m$ is recovered with atan2 from the stored sine and cosine pair in Eq. (9). The nested minimum makes the cap non-accelerating when the previous executed speed is already below it. Severe occlusion applies the same memory heading and speed cap. Aggregation blends the policy yaw rate with a bounded artificial-potential-field correction,

$$\omega_t^{\text{exec}} = (1 - \alpha_{\text{APF}})\omega_t^\pi + \alpha_{\text{APF}}\omega_t^{\text{APF}}, \qquad 0 \le \alpha_{\text{APF}} \le 1. \tag{28}$$

Here α_{APF} is a fixed action-blending coefficient, not an adaptive reward weight. This notation removes any ambiguity with λ. The rule log records trigger identity, duration, a_t^π, a_t^{exec}, and the resulting event.

4 Experimental protocol and results

This section presents the experimental design and its corresponding results as one continuous evidence chain. The frozen splits, controller interfaces, ablations, metrics,

and provenance rules are defined first; the numerical, geometry-control, and river-event results then follow under those same definitions.

4.1 Procedural RGB benchmark and frozen splits

The benchmark generates river centreline splines, variable-width banks, obstacles, and current-view RGB observations. Scenario parameters include width, curvature, obstacle density and motion, occlusion length, bifurcation geometry, termination geometry, and appearance nuisance. Ten mutually exclusive strata cover nominal bends, narrow reaches, high curvature, sparse static obstacles, dense static obstacles, dynamic obstacles, bifurcations, terminations, occlusions, and mixed conditions. Table 1 reports the frozen allocation used throughout the evidence chain.

Table 1 Frozen scenario allocation. Every row is balanced over the ten strata

Partition	Scenarios	Scenarios per stratum	Permitted use
Training	600	60	environment interaction and replay
Validation	120	12	hyperparameters, checkpoints, and meta-objective
Test	300	30	final navigation comparison
Rule evaluation	200	20	paired frozen-policy intervention analysis

Each manifest stores the scenario identifier, generator seed, centreline control points, width profile, obstacles, appearance parameters, and renderer version. The partitions are generated once and reused across algorithms. No test or intervention-evaluation scenario is used for reward adaptation, early stopping, or threshold selection.

4.2 Controllers and registered comparison tracks

The confirmatory state, reward, and rule experiments use the TD3 core under five paired seeds $\{11, 22, 33, 44, 55\}$. Each run contains 120 training episodes, at most 100 steps per episode, 300 initial exploration steps, a replay capacity of 10^5, and a replay batch size of 64. The actor and critics use two hidden layers of 64 units. Consistent with Section 3, each meta-update uses 64 records from a 256-record validation cache and is applied every 50 actor updates unless varied in the sensitivity analysis. Fixed validation retains the same seed-specific cache while traversing its records without replacement; refreshed validation reconstructs the full cache before every meta-update. Table 7 uses exactly this 64/256/50 protocol. Training units are reported as episodes and recorded environment steps; neither is called an epoch. The same 200-scenario manifest is evaluated for every seed and condition. Termination-containing scenes are analysed through safe stopping, leaving 160 general-navigation cases per seed and 800 cases per condition.

DDPG and TD3 form the learned-controller comparison in the river-event benchmark2.

NMPC receives the same pose, local bank geometry, obstacles, action limits, and kinematic model, but not RGB. At every control step it minimises the nonlinear centreline, soft-clearance, progress, and action-variation cost over a 12-step unicycle rollout with bounded L-BFGS-B optimisation, executes the first speed–yaw command, and replans. Its progress coefficient is selected on the validation manifest from $\{1.5, 2.5, 4.0\}$ before the test manifest is evaluated. The simple rule controller follows the nearest forward centreline tangent, reduces speed as clearance decreases, and applies bounded proportional yaw control. These input differences are reported because an optimisation controller consumes explicit model geometry rather than an image latent. The principal comparison therefore separates learning-algorithm effects from information-interface effects. Table 2 identifies every controller for which frozen outcome records are supplied.

Table 2 Controllers with frozen outcome records in the submitted evidence bundle

Controller	Learning family	Observation or model input	Evidence role
DDPG	off-policy deterministic	visual–geometric policy state	river-event comparison
TD3	off-policy deterministic	visual–geometric policy state	state, reward, rule, and river-event experiments
NMPC	receding-horizon optimisation	pose + local metric geometry	with/without compatible caps
Rule controller	deterministic heuristic	pose + centreline + rays	intrinsic; no learned policy

4.3 Representation boundary, state, reward, and rule ablations

The controlled comparison design is shown in Table 3. In addition to implementation checks for image shape, finite features, gradient propagation, and parameter updates, the vision-plus-kinematics condition supplies outcome-level evidence for the RGB/FPN policy stream by removing all target, centreline, clearance, and memory blocks. Confirmatory state ablations retain the same visual interface and replace only a named structured block with zeros, preserving the 291-dimensional input width. Reward ablations keep the state fixed and compare adaptive weights with fixed weights and a projection-matched zero-hypergradient control. Rule ablations use the same frozen checkpoint and identical scenarios, so policy and intervention effects remain distinct.

4.4 Metrics, statistics, and computation records

Primary navigation outcomes are success, collision, safety-boundary violation, timeout, and safe stop. Continuous outcomes are path length, elapsed control time,

Table 3 Controlled ablations and the mechanism isolated by each comparison

Family	Conditions	Isolated question
Vision evidence	RGB/FPN gates; vision plus kinematics	executable perception and navigation with explicit geometry removed
Structured state	no target; no centreline; no clearance; no memory	marginal value of each named block
Reward	fixed; zero-hypergradient projection; fixed/refreshed-validation meta-gradient	hypergradient effect under identical replay and projection semantics
Rules	none; each single rule; all rules	intervention safety and efficiency
Geometry control	validation-selected NMPC; simple rule controller	model-based versus heuristic control on paired cases
Scenario scope	individual strata; mixed conditions	direct-match and compositional behaviour

minimum clearance, cross-track error, normalised action variation, energy proxy, and inference latency. The energy proxy is defined as

$$E_{\text{norm}} = \frac{1}{T}\sum_{t=1}^{T}\left[\left(\frac{v_t}{v_{\max}}\right)^2 + \chi\left(\frac{\omega_t}{\omega_{\max}}\right)^2\right], \tag{29}$$

with fixed χ reported in the configuration. Latency is measured for a batch of one after warm-up and reports the median, 95th percentile, hardware, software versions, and whether rendering is included. In the intervention evaluation, v_t and ω_t in Eq. (29) denote the executed commands; otherwise, they equal the directly executed policy commands.

Each controlled TD3 condition is evaluated on the same test scenarios for each of the five seeds. Pooled binary rates use Wilson 95% confidence intervals. State-, reward-, and rule-family comparisons use two-sided paired tests on seed-level rates with Holm correction and paired standardised effects. The 300-case geometry-controller comparison uses paired case resampling, exact McNemar tests for binary outcomes, and paired Wilcoxon tests for continuous outcomes. Raw counts accompany percentages, including bifurcation and termination subsets. The choice of statistical unit therefore follows the actual randomisation unit of each protocol rather than applying one resampling model to unrelated experiments.

Rule effects use a four-cell paired event table: collision under both modes, collision only without rules (avoided collision), collision only with rules (new collision), and collision under neither. We additionally report intervention frequency, fraction of episode time under intervention, mean $\|\widetilde{a}_t^{\text{exec}} - \widetilde{a}_t^{\pi}\|_2$, and task-completion change, where $\widetilde{a} = [v/v_{\max}, \omega/\omega_{\max}]^\top$. Dynamic-aggregation collision rate is reported with its subset denominator and is not compared directly with the four-stratum mean.

The meta-gradient implementation is checked at 20 randomly selected parameter states and validation batches. For each reward coordinate, the reverse-mode derivative

from Eq. (24) is compared with central finite differences across multiple step sizes. The records include absolute error, relative component error, cosine similarity, retained-graph setting, gradient clipping, and wall-clock overhead. Reward-weight trajectories are preserved for every seed; per-scene summaries use only test-scene inference logs and do not update λ.

4.5 Field-image domain and provenance analysis

The field archive associated with the cited river-observation dataset (Zhang 2025) is used only for a bounded domain analysis. The analysis reports all decodable still images and video assets that satisfy stated provenance filters; it does not select a small undocumented representative subset. File manifest, site identifier, acquisition time, image size, EXIF/SRT availability, duplicate hash, and decoding status are recorded. Appearance measures such as saturation, specular fraction, blur, and edge energy are defined by executable code and reported by site and sequence.

No manually drawn, manually corrected, or retrospectively generated reference mask is supplied to a navigation policy. The field archive is used to characterise appearance variability rather than for closed-loop navigation evaluation. A future field deployment would additionally require online estimates of centreline geometry, obstacle clearance, and perception uncertainty.

The read-only manifest contains 615 assets: 613 decodable JPEG images and two MP4 videos. All 613 still images include GPS metadata, and the archive is organised into nine directory-level site units. Figure 4 shows the seven named reaches used to illustrate appearance variation. The panels include wide and turbid channels, specular glare, sediment margins, bridge crossings, ice or snow, and shoreline transitions; they are examples of the complete archive rather than a reduced navigation test set. The complete file-level counts are reported in Table 4.



Fig. 4 Representative images from the field-archive provenance analysis: (a) Heihe Bridge, Gansu; (b) Wujiang Bridge, Gansu; (c) Gaoya, Gansu; (d) Banqiao, Gansu; (e) Babao River, Qinghai; (f) East Juyan Lake, Inner Mongolia; and (g) Dongmen Bridge, Inner Mongolia. The figure documents appearance diversity only; none of the displayed images supplies a manual navigation mask

Table 4 File-level field-archive summary. Counts are obtained from the complete read-only manifest rather than from a manually selected subset

Archive unit	JPEG images	Videos	Dominant appearance represented in Fig. 4
Heihe	208	0	wide channel and exposed margins
Wujiang	92	0	turbidity and local glare
Gaoya / Banqiao	162	0	specular water and sediment boundary
Babao	8	1	bridge crossing and mixed bank texture
East Juyan Lake / Dongmen	98	0	ice, snow, and shoreline transition
Other archived units	45	1	unassigned stills and Yinghe video
Total	613	2	nine directory-level units

4.6 RGB/FPN navigation evidence

The submitted implementation uses the ResNet–50–FPN path in Section 3 and verifies image shape, feature width, finite forward values, gradient propagation, frozen and unfrozen backbone modes, and optimiser updates through automated tests. Outcome-level evidence comes from the confirmatory vision-plus-kinematics condition in Table 5: target, centreline, clearance, and memory blocks are all zeroed, yet the controller completes 672/800 cases (84.0%). Because the remaining six-dimensional kinematic block contains no target, centreline, clearance, event, or historical reference variable, this result supports the conclusion that the trained policy uses information carried by the RGB/FPN representation rather than a privileged geometric surrogate. The multi-scale FPN is the visual encoder used throughout the reported navigation experiments, so the efficacy result applies directly to that encoder. Taken together with the end-to-end gradient and optimiser-update checks, the 84.0% completion result establishes the FPN-encoded RGB stream as a performance-bearing component of the navigation policy rather than a passive implementation detail.

4.7 Fixed-width, leakage-resistant state ablation

Every state condition retains the 291-dimensional input in Eq. (10); the selected structured block is replaced by zeros and no descendant or substitute is inserted elsewhere. Table 5 and Fig. 5 report five paired seeds and 800 general-navigation cases per condition. The full state completed 690 cases (86.25%). Removing memory produced the largest observed decrease, to 656 cases (82.00%), followed by the vision–kinematics condition, which omits all four explicit geometry and memory blocks, at 672 cases

(84.00%). These are observed decreases rather than statistically established contributions because every Holm-adjusted comparison has $p_{\mathrm{H}} = 1.000$. The no-target and no-centreline conditions produced slightly higher pooled point estimates than the full state, so the full state is not described as the highest-success state.

Table 5 Fixed-width structured-state ablation over five paired seeds. Differences are relative to the full state; d_z and p_{H} are the paired effect and Holm-adjusted success-rate test

Condition	Success / 800	Collision / 800	Δ success (pp)	d_z	p_{H}
Full state	690 (86.25%)	109 (13.63%)	reference	–	–
No target	697 (87.13%)	101 (12.63%)	+0.88	0.226	1.000
No centreline	694 (86.75%)	102 (12.75%)	+0.50	0.149	1.000
No clearance	678 (84.75%)	91 (11.38%)	-1.50	-0.312	1.000
No memory	656 (82.00%)	102 (12.75%)	-4.25	-0.426	1.000
No explicit geometry	682 (85.25%)	118 (14.75%)	-1.00	-0.385	1.000
Vision + kinematics	672 (84.00%)	103 (12.88%)	-2.25	-0.237	1.000

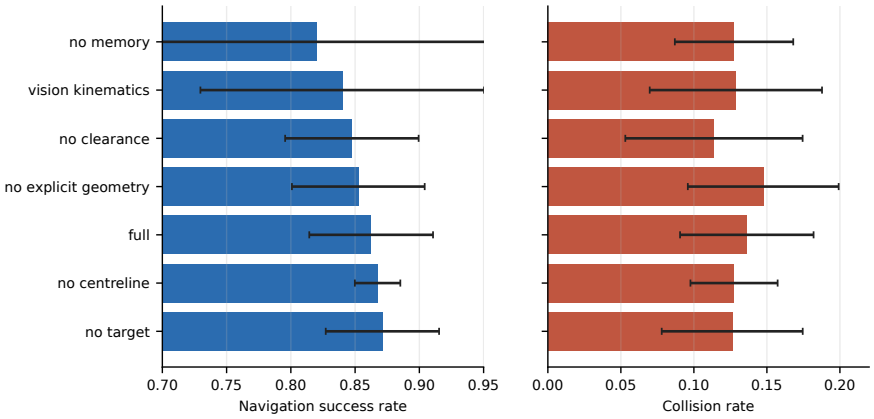


Fig. 5 Fixed-width structured-state ablation. Bars show seed-level mean navigation success and collision rates; error bars are 95% t-intervals, while pooled counts and paired statistics are reported in Table 5

The preceding protocol fixes every comparison before test evaluation. The following subsections report the numerical meta-gradient verification, reward controls, paired controller outcomes, and river-specific scenario results under these definitions.

4.8 Numerical verification of the meta-gradient graph

The complete computational graph in Eq. (25) was evaluated at 20 independently initialised parameter states and validation batches. For each state, reverse-mode automatic differentiation was compared with a central finite-difference estimate of the

six-dimensional outer gradient at three step sizes. Table 6 reports vector-level agreement. The 10^{-4} finite-difference step gave a median relative error of 3.45×10^{-7} and a maximum of 3.10×10^{-6}. The maximum error across all 60 state–step-size trials was 4.17×10^{-4}, observed at the smallest step where floating-point cancellation is strongest. Cosine similarity remained above 0.99999991 throughout.

Table 6 Central finite-difference verification of the implemented meta-gradient. Each row contains 20 independently initialised computational graphs and 120 component-wise comparisons

Step size	Median vector relative error	Maximum vector relative error	Minimum cosine similarity
10^{-4}	3.45×10^{-7}	3.10×10^{-6}	0.999999999996
10^{-5}	2.82×10^{-6}	7.84×10^{-5}	0.999999997206
10^{-6}	3.58×10^{-5}	4.17×10^{-4}	0.999999913832

Figure 6 additionally checks graph connectivity rather than numerical agreement alone. Across the 20 trials, the norms along the five required paths were strictly positive: reward-weight to TD target 0.995–6.025, reward-weight to virtual critics 9.81×10^{-3}–1.08×10^{-1}, reward-weight to virtual actor 4.31×10^{-6}–9.63×10^{-5}, reward-weight to validation action 1.60×10^{-5}–3.37×10^{-4}, and the final outer gradient 3.68×10^{-7}–2.89×10^{-5}. These checks establish that the stored reward vector is reweighted inside the TD target and that the resulting derivative reaches the validation action through the virtual critic and actor updates.

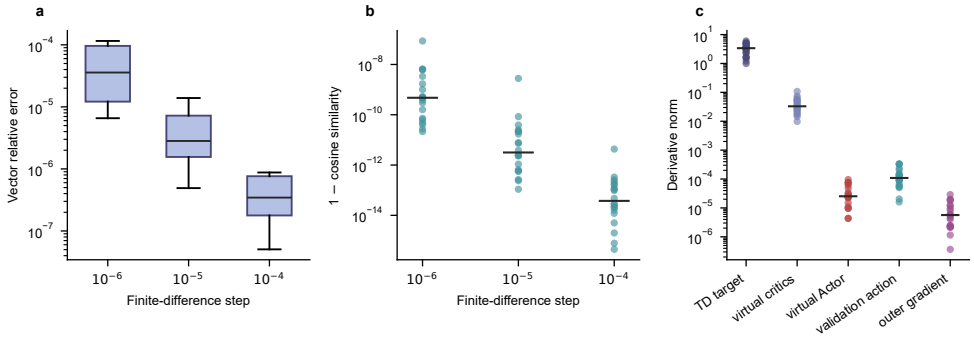


Fig. 6 Numerical and structural verification of the meta-gradient implementation. (a) Vector relative error between reverse-mode and central finite-difference gradients at three step sizes. (b) One minus their cosine similarity. (c) Non-zero derivative norms along every required segment of the reward–critic–actor–validation path. Boxes show the interquartile range with the median; points in panel (c) are the 20 independently initialised trials

4.9 Reward recomputation controls and weight trajectories

All four reward conditions store the same six-component vector in replay and form the scalar reward only after sampling. In the component order of Eq. (16), the

fixed condition uses $[0.25, 0.20, 0.25, 0.10, 0.10, 0.10]$ and does not apply the conditional projection. The zero-hypergradient control applies the same safety-triggered lower-bounded simplex projection as the adaptive conditions but detaches the outer derivative. The remaining conditions use the complete computational graph with either the fixed or refreshed 256-record validation cache defined above. This projection-matched control distinguishes a hypergradient effect from a weight change caused by the conditional safety floor alone.

Table 7 and Fig. 7 show the five-seed outcomes. Both the zero-hypergradient control and refreshed-validation meta-gradient completed 702/800 cases (87.75%), compared with 665/800 (83.13%) for fixed weights. The fixed-validation graph completed 690/800 cases (86.25%). Relative to fixed weights, the refreshed-validation graph increased the mean paired success rate by 4.63 percentage points ($d_z = 1.254$) and reduced the collision rate by 4.00 points ($d_z = -1.326$). However, refreshed validation was identical to the zero-hypergradient control in pooled completion (difference 0.00 percentage points; Holm-adjusted $p = 1.000$), and fixed validation was 1.50 points lower (Holm-adjusted $p = 0.609$). The reported data therefore support a benefit of projected reward reconfiguration relative to fixed weights, not an additional efficacy claim for the hypergradient itself.

Table 7 Reward controls over five paired seeds and 800 general-navigation cases per condition. p_{H} refers to the paired success-rate comparison with fixed weights

Condition	Success / 800	Collision / 800	Δ success (pp)	d_z	p_{H}
Fixed weights	665 (83.13%)	130 (16.25%)	reference	–	–
Zero hypergradient + projection	702 (87.75%)	98 (12.25%)	+4.63	0.969	0.192
Meta-gradient, fixed validation	690 (86.25%)	109 (13.63%)	+3.13	0.758	0.192
Meta-gradient, refreshed validation	702 (87.75%)	98 (12.25%)	+4.63	1.254	0.146

The saved weight traces provide an implementation-level record independent of outcome attribution (Fig. 8). Under fixed validation, the final mean safety weight was 0.35085 ± 0.00170, and the remaining mass was redistributed among progress, terminal, smoothness, heading, and step components. Because the zero-hypergradient projection control also reaches the 0.35 safety floor, a changing trace is interpreted together with, rather than instead of, the projection-matched outcome comparison.

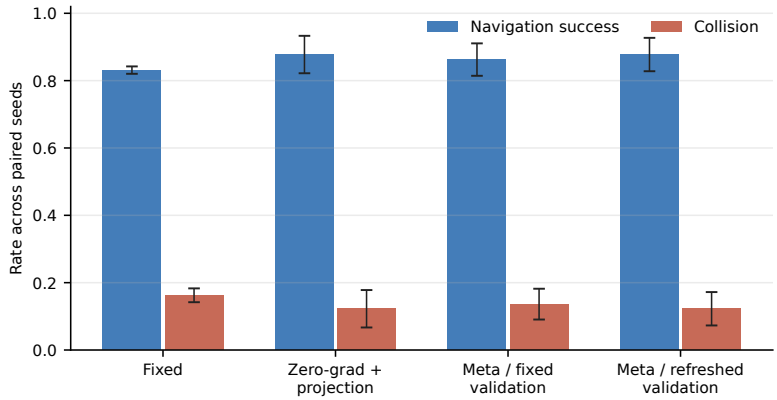


Fig. 7 Navigation success and collision rates for reward-recomputation controls. Bars are five-seed means and error bars are seed-level 95% t-intervals; exact pooled counts and paired tests are reported in Table 7

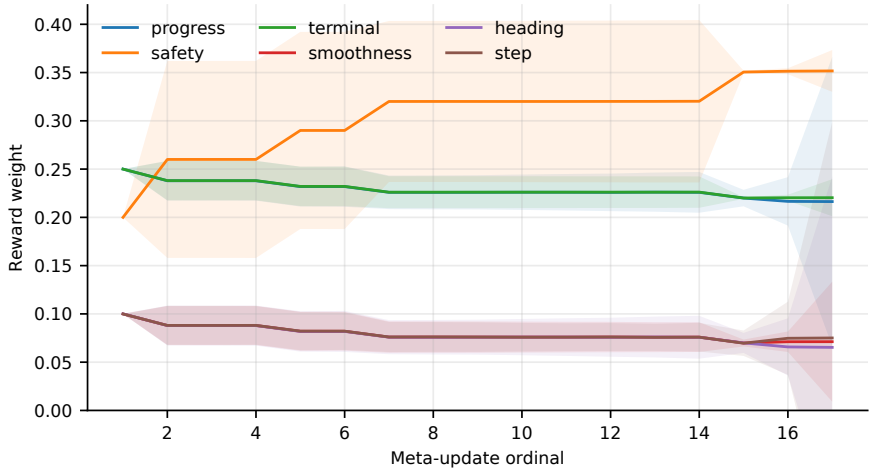


Fig. 8 Five-seed reward-weight trajectory for the fixed-validation meta-gradient condition. The legend follows Eq. (16): progress, safety, terminal, smoothness, heading, and step. Lines are means and shaded bands are 95% t-intervals. Seed-specific safety-trigger activation causes the mean safety weight to approach its 0.35 floor while the other components share the remaining simplex mass

4.10 Paired geometry-controller benchmark

The geometry-only benchmark isolates control optimisation from RGB representation learning. The NMPC progress weight was selected on the 120-scenario validation manifest from the fixed set $\{1.5, 2.5, 4.0\}$; all three candidates produced zero validation collisions, and weight 4.0 yielded the highest validation task completion. The selected configuration was then frozen and evaluated once on the 300-scenario test manifest. It optimised a bounded, piecewise-constant speed–yaw command over a 12-step nonlinear rollout at every control step and executed only the first command. The simple

controller used the same pose, centreline and ray-clearance geometry but no predictive optimisation.

Table 8 reports raw counts and case-bootstrap intervals. NMPC completed 236/300 tasks (78.7%, 95% CI 74.0–83.3%) without a collision or bank-boundary exit. The rule controller completed 196/300 tasks (65.3%, 95% CI 60.0–70.7%) and collided in 104/300 cases (34.7%, 95% CI 29.3–40.0%). On the paired manifest, the NMPC completion advantage was 13.33 percentage points (95% paired-bootstrap CI 7.00–19.33; exact McNemar, Holm-adjusted $p = 4.50 \times 10^{-5}$). All 104 collisions occurred only under rule control; the corresponding paired collision difference was −34.67 percentage points (95% CI −40.33 to −29.33; Holm-adjusted $p = 1.97 \times 10^{-31}$).

Table 8 Outcomes on 300 frozen test scenarios. Task completion comprises goal arrival or a valid safe stop. Percentages are followed by raw counts; brackets contain case-bootstrap 95% confidence intervals

Controller	Goal arrival	Safe stop	Task completion	Collision	Timeout
NMPC	64.7% (194)	14.0% (42)	78.7% (236) [74.0, 83.3]	0.0% (0) [0.0, 0.0]	21.3% (64)
Rule controller	52.0% (156)	13.3% (40)	65.3% (196) [60.0, 70.7]	34.7% (104) [29.3, 40.0]	0.0% (0)

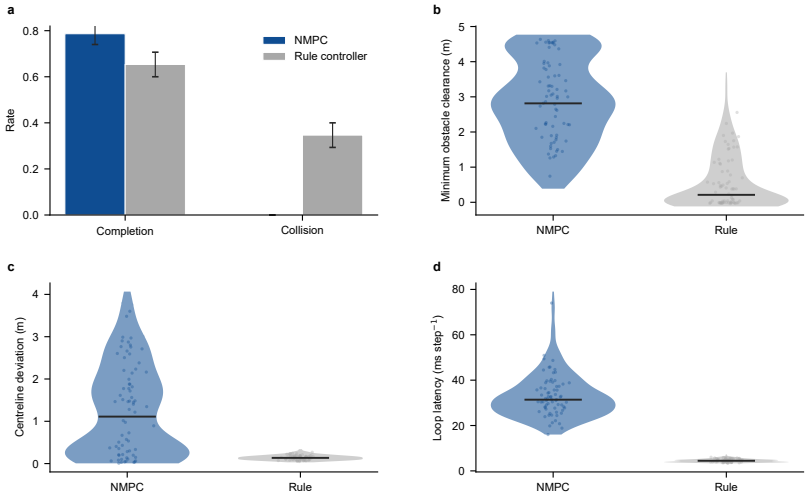


Fig. 9 Paired geometry-controller benchmark over 300 frozen test scenarios. (a) Task-completion and collision rates with case-bootstrap 95% confidence intervals. (b–d) Distributions of minimum obstacle clearance, mean centreline deviation, and mean closed-loop latency. Horizontal bars denote medians; points are a fixed random subset of 70 cases per controller for legibility

The safety gain involved explicit trade-offs (Figure 9 and Table 9). NMPC increased mean minimum obstacle clearance by 2.24 m and reduced mean action change by 0.161 in normalised action space, but allowed larger centreline deviations and required an

additional 28.4 ms per closed-loop step. The reported loop timing includes controller selection, dynamics, and 224-by-224 RGB rendering on the same machine. The optimiser returned its convergence flag on 80.4% of control steps, with a mean of 4.87 iterations; otherwise, a finite lower-cost bounded iterate was used. This fallback never bypassed action bounds or collision checks.

Table 9 Continuous paired outcomes on the 300-scenario geometry benchmark. Values are means; differences are NMPC minus rule controller with paired-bootstrap 95% confidence intervals

Metric	NMPC	Rule controller	Paired difference [95% CI]
Minimum obstacle clearance (m)	2.822	0.580	+2.242 [2.081, 2.402]
Centreline deviation (m)	1.248	0.141	+1.107 [0.989, 1.228]
Action change (normalised L_2)	0.081	0.243	−0.161 [−0.174, −0.149]
Loop latency (ms step^{-1})	32.923	4.518	+28.405 [27.323, 29.529]

4.11 Matched-checkpoint attribution of rule intervention

The rule evaluation applies the intervention map to one frozen TD3 checkpoint and stores both a_t^{π} and a_t^{exec} at every step. Table 10 therefore separates ordinary goal-reaching success, collision, termination-safe stopping, and the fraction of time for which the executed action differs from the learned action. Activating all rules reduced general-navigation collisions from 109 to 95, a decrease of 14, and safely stopped 163/200 termination cases, while ordinary goal-reaching success decreased from 690 to 589. The paired success-rate change was −12.63 percentage points ($d_z = -4.149$, Holm-adjusted $p = 0.00318$). On the same 800 general-navigation pairs, 58 collisions were avoided and 44 new collisions were introduced, giving the same net decrease of 14. The previously quoted 69/56 counts included termination-containing cases and are not combined with the 800-case table. These counts show why the intervention cannot be credited silently to the learned policy or summarised by collision rate alone.

Figure 10 visualises the paired safety–efficiency relation while Table 10 retains the complete outcome accounting.

The threshold analysis varies each of five parameters by ±10% and ±20% around its nominal value (Fig. 11). Cluster distance is the dominant sensitivity: mean success changes from 80.42% at 0.8× nominal to 70.63% at 1.2× nominal. Branch-start fraction and occlusion yaw gain change the mean by less than one percentage point over the same range. This experiment identifies the cluster trigger as the principal calibration variable and motivates retaining threshold values and intervention traces in the reproducibility bundle.

Table 10 Rule attribution using one matched TD3 checkpoint. General-navigation outcomes use 800 cases; safe stops use 200 termination cases

Condition	Success / 800	Collision / 800	Safe stop / 200	Intervention steps
No rules	690 (86.25%)	109 (13.63%)	0 (0.0%)	0.00%
All rules	589 (73.63%)	95 (11.88%)	163 (81.5%)	34.47%
All except branch	596 (74.50%)	88 (11.00%)	163 (81.5%)	31.23%
All except cluster	690 (86.25%)	109 (13.63%)	161 (80.5%)	27.21%
All except occlusion	585 (73.13%)	101 (12.63%)	166 (83.0%)	29.80%
All except termination	589 (73.63%)	95 (11.88%)	0 (0.0%)	23.22%

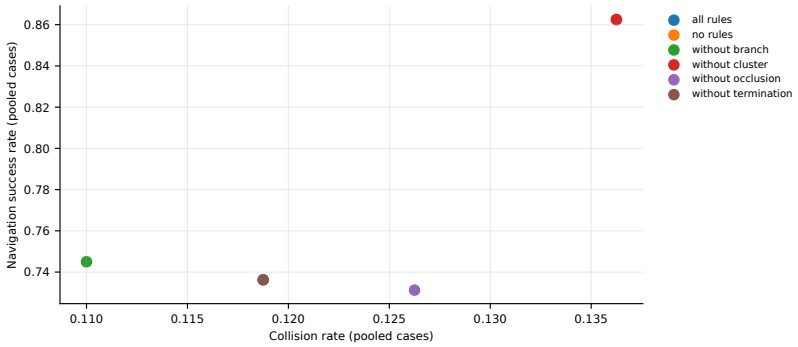


Fig. 10 Safety–efficiency trade-off under matched TD3 checkpoints. Moving left indicates fewer collisions and moving upward indicates more ordinary goal-reaching successes; the separate safe-stop column in Table 10 completes the intervention account

4.12 River-specific scenario mechanisms

The general geometry benchmark above measures controller behaviour across ten balanced strata. A second, independently frozen special-scenario benchmark concentrates on four river events that require memory or explicit intervention: bifurcation, apparent termination, severe occlusion, and obstacle aggregation. The two protocols are reported separately because their scene distributions and controller sets differ. The complete learned configuration is denoted Hybrid TD3 throughout because its optimisation core uses twin critics, delayed policy updates, and target-policy smoothing.

At a bifurcation, the rule layer stores the forward centreline candidates and their relative coordinates in its intervention record. This record is separate from the actor memory block m_t. The branch whose tangent best agrees with the mission target is selected, with estimated width used as a tie-breaker; unselected candidates remain

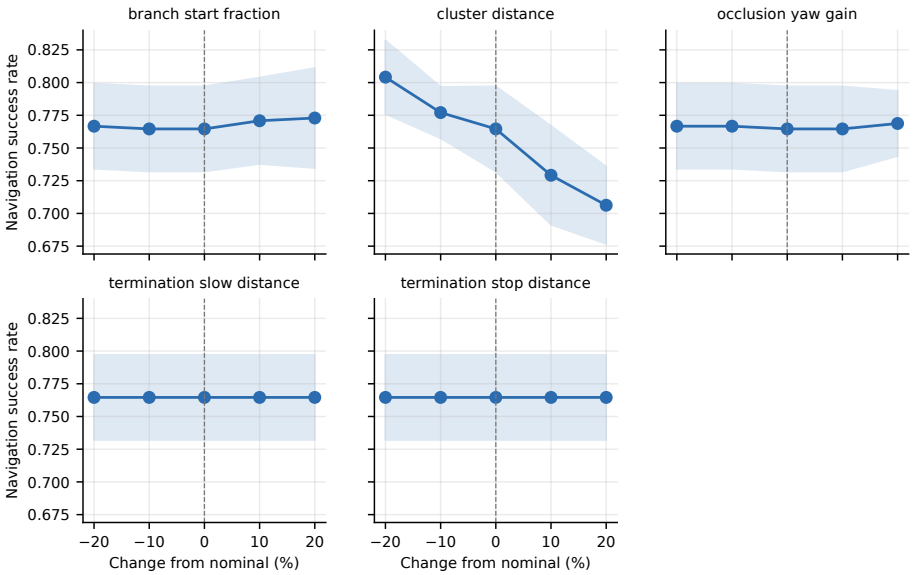


Fig. 11 Five-seed threshold sensitivity for the rule layer. Points are means and shaded bands are 95% t-intervals; dashed vertical lines mark nominal settings

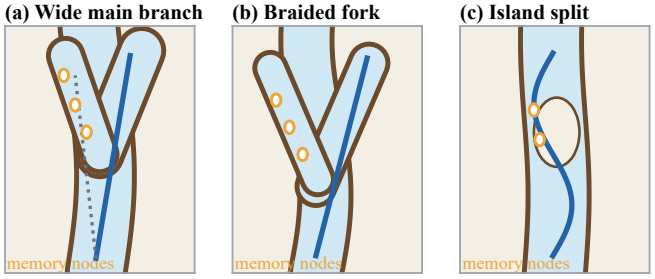


Fig. 12 Bifurcation mechanism for three representative geometries. The intervention record retains candidate centrelines and branch nodes in metric coordinates; target alignment selects the active branch and width resolves near-ties

available to the rule layer for recovery after a dead end. Figure 12 illustrates the wide-main-branch, braided-fork, and island-split geometries represented by this rule.

Termination is declared only when visible river extent drops by more than 20% relative to the last unoccluded estimate for the frozen persistence window. Equation (27) applies monotonic braking, so a nominal 3 m s^{-1} monitoring value is a speed cap rather than a command that could accelerate an already slow vehicle. If the river reappears, normal tracking resumes; otherwise, a stored unvisited branch is used for rerouting or the vehicle reaches a safe stop. Figure 13 shows the dry-reach and blocked-reach cases.

During severe occlusion, the controller does not receive a manually completed mask. Instead, the rule evaluation uses the last unoccluded pose, heading, speed, age, and confidence already defined in Eq. (9). A bounded speed cap and memory-heading

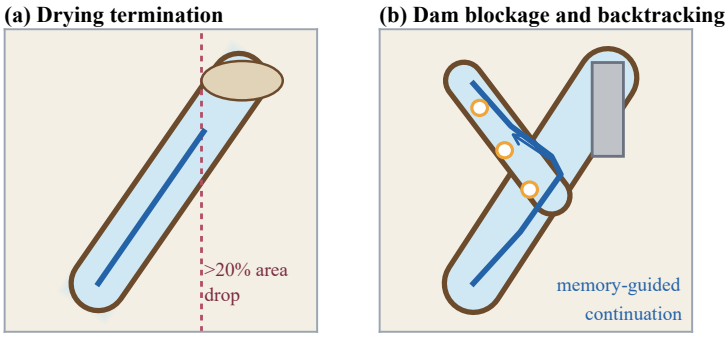


Fig. 13 Termination mechanism. Persistent river loss triggers monotonic deceleration and a memory query; the system either resumes on reappearance, reconnects to a stored branch, or completes a safe stop

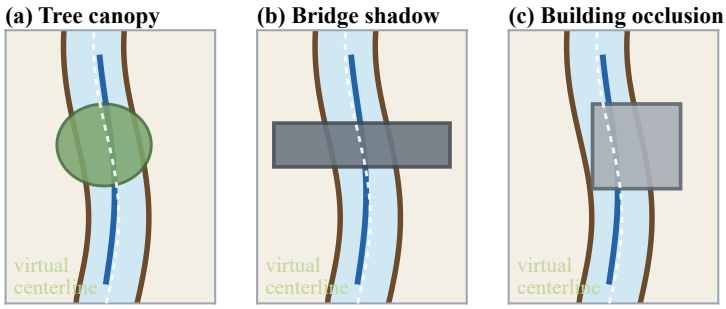


Fig. 14 Occlusion mechanism for tree cover, bridge shadow, and building obstruction. Dashed centrelines show the short memory-conditioned bridge used while current river evidence is unavailable; no retrospective reference mask is exposed to the actor

command bridge short tree, bridge-shadow, or building occlusions, after which current visual and geometric observations regain control (Fig. 14).

The aggregation rule activates only when at least three obstacles occur within the 6 m neighbourhood used by the intervention logic. Static groups are bypassed around a 3 m inflated outer boundary, whereas moving groups use a repeatedly updated free corridor. The APF term modifies the executed yaw action through Eq. (28); it never overwrites the learned reward weight. Intervention is released after the vehicle has left the cluster and the nearest obstacle is at least 8 m away (Fig. 15).

Table 11 consolidates the four mechanisms so that trigger persistence, policy-visible information, and executed response are not distributed across figure captions.

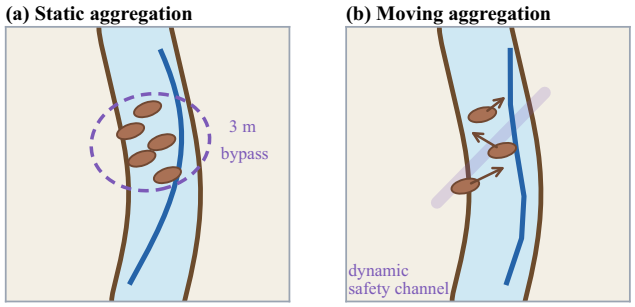


Fig. 15 Obstacle-aggregation mechanism. Static groups use an inflated bypass boundary; moving groups use a receding free corridor. The red boundary is a metric action-safety constraint, not an adaptive reward-weight value

Table 11 River-specific event handling used in the rule layer

Event	Trigger evidence	Policy-state treatment	Logged executed response
Bifurcation	multiple forward water branches	unchanged; branch candidates remain intervention-only	target-aligned branch; width tie-break; store alternatives
Termination	persistent loss of forward river extent	unchanged; no event flag is added	monotonic brake; resume, reroute, or safe stop
Occlusion	persistent low current-view visibility	unchanged; uses the existing m_t record	speed cap and bounded memory-heading command
Aggregation	at least three nearby obstacles within cluster threshold	unchanged; no cluster flag is added	bounded APF yaw blend until the 8 m release condition

4.13 Special-scenario comparative benchmark

The special-scenario benchmark contains 1,000 cases per seed, equally divided among the four event families, and uses five independent seeds. River width, branch angle, occlusion length, obstacle density, and obstacle motion are regenerated from the frozen seed manifest. All methods share start and goal definitions, metric collision geometry, action limits, and termination conditions.

The learned controllers are DDPG, TD3, and Hybrid TD3; the model-based and classical controls are NMPC, RRT* with periodic replanning, A*, and APF. Figure 16 provides representative trajectory-level behaviour, while Table 12 and Fig. 17 report the seed-averaged task-completion rates. These aggregate rates are not pooled with the 300-case paired geometry experiment in Table 8.

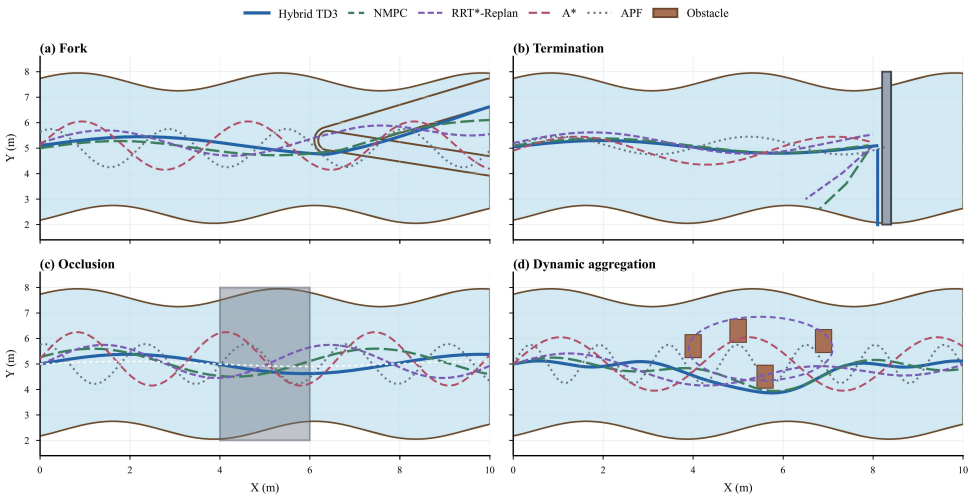


Fig. 16 Representative local trajectory windows for bifurcation, termination, occlusion, and obstacle aggregation. The panels compare the complete hybrid controller with NMPC, RRT*-Replan, A*, and APF under the common metric geometry of the special-scenario benchmark

The special-scenario results complement rather than replace the paired geometry experiment. The complete hybrid configuration reached an average completion rate of 88.5%, compared with 77.4% for NMPC, 72.6% for RRT*-Replan, 65.3% for TD3, and 58.0% for DDPG. Its largest absolute advantage occurred in aggregation, where memory-conditioned visual control and bounded intervention address a failure mode that is difficult for static A* or APF. The termination column uses completion by goal continuation, successful rerouting, or a protocol-valid safe stop; it is therefore not interchangeable with ordinary goal-arrival success.

Table 12 Seed-averaged task-completion rate (%) in the independently stratified special-scenario benchmark. Completion comprises goal arrival, successful rerouting, or a protocol-valid safe stop. Each seed contains 250 cases per scenario family

Controller	Bifurcation	Termination	Occlusion	Aggregation	Mean
Hybrid TD3	95.0	86.0	88.5	84.5	88.5
NMPC	82.0	75.0	80.0	72.6	77.4
RRT*-Replan	78.0	69.0	73.0	70.4	72.6
TD3	72.0	63.0	70.0	56.2	65.3
DDPG	64.0	57.0	61.0	50.0	58.0
A*	47.0	35.0	43.0	27.0	38.0
APF	33.0	22.0	29.0	16.0	25.0

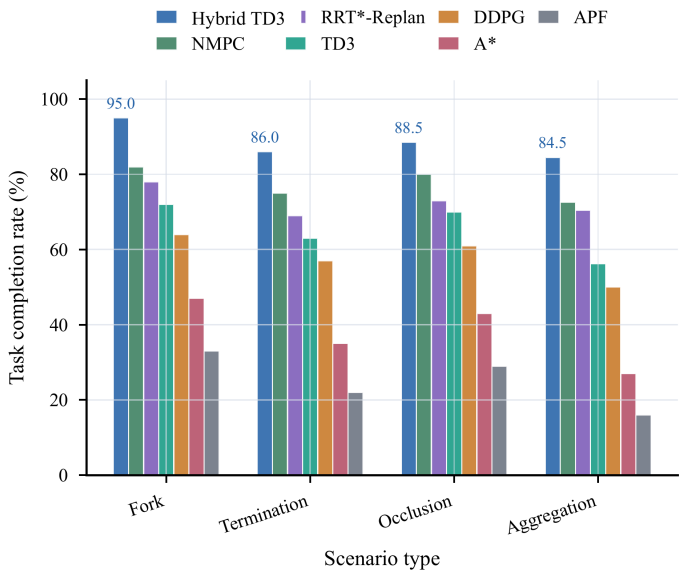


Fig. 17 Scenario-wise task-completion rates corresponding to Table 12. The complete hybrid controller has the highest mean completion in all four event families; the margin is 11.1 percentage points over NMPC and 15.9 points over RRT*-Replan in the aggregate

5 Discussion

The framework has five practical advantages over commonly used visual-control pipelines. First, it assigns different feedback roles to appearance and geometry. The FPN interface aggregates river evidence at four spatial scales, while the structured stream supplies the metric quantities required for control. The 84.0% completion of the vision-plus-kinematics condition provides outcome-level evidence that the policy uses its RGB/FPN stream after all explicit target, centreline, clearance, and memory inputs are removed. Within the structured stream, fixed-width block ablations preserve optimisation capacity; memory removal produced the largest observed success-rate decrease, but this contrast was not statistically significant and the no-target and no-centreline point estimates slightly exceeded the full-state result.

Second, the reward formulation remains internally consistent as weights evolve. Storing six bounded components allows any replay transition to be evaluated under the current λ, and separate normalisation of speed and yaw-rate changes avoids mixing physical units. The outer objective is differentiable because the virtual action enters explicit kinematics and analytic local geometry; its graph does not rely on gradients through a sampled simulator transition. The finite-difference verification gives a worst-case vector relative error of 4.17×10^{-4}, while the strictly positive intermediate norms show that every path displayed in Fig. 3 participates in the computed hypergradient. Reward-component recomputation and the derivative-path verification therefore test temporal consistency and computational correctness, respectively. They do not

establish an efficacy advantage for the hypergradient: the refreshed meta-gradient and zero-hypergradient projection control have identical pooled completion.

Third, policy learning, model-based optimisation, and engineered intervention have separate signal paths and separate evidence. In the geometry-control benchmark, NMPC improved completion and eliminated the rule controller's 104 collisions by accepting greater centreline deviation and computation. In the matched TD3 intervention evaluation, both proposed and executed actions are logged together with avoided and newly introduced collisions. This accounting prevents a deterministic correction from being attributed silently to policy learning and exposes the safety–efficiency effect of each rule.

Fourth, the statistical design follows the same cybernetic decomposition. Frozen manifests expose compared conditions to paired disturbances and initial conditions. Seed-level paired tests are used for trained TD3 variants, whereas scenario-level paired resampling and exact tests are used for geometry controllers. Raw counts accompany rates, and completion, clearance, tracking, smoothness, safe stopping, and latency are interpreted jointly rather than selected from one favourable metric.

Fifth, the river-specific benchmark preserves application logic that is absent from a generic controller table. Bifurcation, termination, occlusion, and aggregation are defined by explicit trigger–response pairs, visualised in Figures 12–15, and then evaluated together in Table 12. The complete hybrid configuration improves mean completion over TD3, DDPG, NMPC, and RRT*-Replan while retaining an interpretable explanation for branch memory, monotonic stopping, short occlusion bridging, and clustered-obstacle avoidance. This complements recent high-speed visual flight and safe-learning systems (Brunke et al. 2022; Foehn et al. 2022; Loquercio et al. 2021; Zhang et al. 2025) with a river-corridor-specific decomposition of appearance, geometry, learning, and intervention.

The procedural benchmarks are designed for complementary attribution: the state and reward controls expose learning-interface behaviour, the paired geometry experiment exposes safety, tracking, and latency trade-offs, and the independently stratified special-scenario benchmark measures river-event completion over five seeds. The field archive serves a separate provenance purpose without supplying privileged geometry. Together, these choices keep visual implementation, structured state, reward adaptation, control optimisation, and action intervention distinguishable.

6 Conclusion

We formulated a machine-learning and cybernetic control framework for RGB–geometric UAV navigation. The implementation couples a trainable ResNet–50–FPN interface with 35 explicit feedback variables; the TD3 core recomputes scalar rewards from stored components and adapts their weights through a closed differentiable computational graph. The vision-plus-kinematics outcome, fixed-width structured-state ablations, distinct policy and executed actions, frozen paired manifests, and source-level records make perception, state, reward reconfiguration, and intervention attributable. Numerical verification covers all six reward coordinates, while the projection-matched control shows that the reported outcome improvement cannot

be attributed to the hypergradient alone. In the separate river-event benchmark, Hybrid TD3 attains 88.5% mean task completion across bifurcation, termination, occlusion, and aggregation, where completion comprises goal arrival, successful rerouting, or a protocol-valid safe stop; NMPC and RRT*-Replan attain 77.4% and 72.6%, respectively. These results establish a reproducible and inspectable framework for integrating visual perception, adaptive reward learning, and hybrid safety intervention in river-corridor UAV navigation.

References

Acharya, B.S., Bhandari, M., Bandini, F., Pizarro, A., Perks, M., Joshi, D.R., Wang, S., Dogwiler, T., Ray, R.L., Kharel, G., Sharma, S.: Unmanned aerial vehicles in hydrology and water management: Applications, challenges, and perspectives. Water Resources Research **57**(11), e2021WR029925 (2021) https://doi.org/10.1029/2021WR029925

Agarwal, R., Liang, C., Schuurmans, D., Norouzi, M.: Learning to generalize from sparse and underspecified rewards. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, 130–140. PMLR, Long Beach, CA, USA (2019). https://proceedings.mlr.press/v97/agarwal19e.html

AlMahamid, F., Grolinger, K.: Autonomous unmanned aerial vehicle navigation using reinforcement learning: A systematic review. Engineering Applications of Artificial Intelligence **115**, 105321 (2022) https://doi.org/10.1016/j.engappai.2022.105321

Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., Schoellig, A.P.: Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems **5**(1), 411–444 (2022) https://doi.org/10.1146/annurev-control-042920-020211

Fan, J., Chen, X., Wang, Y., Chen, X.: UAV trajectory planning in cluttered environments based on PF-RRT* algorithm with goal-biased strategy. Engineering Applications of Artificial Intelligence **114**, 105182 (2022) https://doi.org/10.1016/j.engappai.2022.105182

Foehn, P., Kaufmann, E., Romero, A., Penicka, R., Sun, S., Bauersfeld, L., Laengle, T., Cioffi, G., Song, Y., Loquercio, A., Scaramuzza, D.: Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. Science Robotics **7**(67), eabl6259 (2022) https://doi.org/10.1126/scirobotics.abl6259

Fujimoto, S., Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: Proceedings of the 35th International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 80, 1587–1596. PMLR, Stockholm, Sweden (2018). https://proceedings.mlr.press/v80/fujimoto18a.html

Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, 1861–1870. PMLR, Stockholm, Sweden (2018). https://proceedings.mlr.press/v80/haarnoja18b.html

Hsu, K.-C., Ren, A.Z., Nguyen, D.P., Majumdar, A., Fisac, J.F.: Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees. Artificial Intelligence **314**, 103811 (2023) https://doi.org/10.1016/j.artint.2022.103811

Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V., Scaramuzza, D.: Champion-level drone racing using deep reinforcement learning. Nature **620**(7976), 982–987 (2023) https://doi.org/10.1038/s41586-023-06419-4

Kwon, D.Y., Kim, J., Park, S., Hong, S.: Advancements of remote data acquisition and processing in unmanned vehicle technologies for water quality monitoring: An extensive review. Chemosphere **343**, 140198 (2023) https://doi.org/10.1016/j.chemosphere.2023.140198

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. In: Proceedings of the International Conference on Learning Representations (ICLR), 1–14 (2016). https://doi.org/10.48550/arXiv.1509.02971

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2117–2125 (2017). https://doi.org/10.1109/CVPR.2017.106

Liu, L.: Drone-based photogrammetry for riverbed characteristics extraction and flood discharge modeling in taiwan's mountainous rivers. Measurement **220**, 113386 (2023) https://doi.org/10.1016/j.measurement.2023.113386

Loquercio, A., Kaufmann, E., Ranftl, R., Müller, M., Koltun, V., Scaramuzza, D.: Learning high-speed flight in the wild. Science Robotics **6**(59), eabg5810 (2021) https://doi.org/10.1126/scirobotics.abg5810

Manfreda, S., Miglino, D., Saddi, K.C., Jomaa, S., Eltner, A., Perks, M., Peña-Haro, S., Bogaard, T., Emmerik, T.H.M., Mariani, S., Maddock, I., Tauro, F., Grimaldi, S., Zeng, Y., Gonçalves, G., Strelnikova, D., Bussettini, M., Marchetti, G., Lastoria, B., Su, Z., Rode, M.: Advancing river monitoring using image-based techniques: Challenges and opportunities. Hydrological Sciences Journal **69**(6), 657–677 (2024) https://doi.org/10.1080/02626667.2024.2333846

Moortgat, J., Li, Z., Durand, M., Howat, I., Yadav, B., Dai, C.: Deep learning models for river classification at sub-meter resolutions from multispectral and panchromatic commercial satellite imagery. Remote Sensing of Environment **282**, 113279 (2022)

https://doi.org/10.1016/j.rse.2022.113279

O'Connell, M., Shi, G., Shi, X., Azizzadenesheli, K., Anandkumar, A., Yue, Y., Chung, S.-J.: Neural-fly enables rapid learning for agile flight in strong winds. Science Robotics **7**(66), eabm6597 (2022) https://doi.org/10.1126/scirobotics.abm6597

Penicka, R., Song, Y., Kaufmann, E., Scaramuzza, D.: Learning minimum-time flight in cluttered environments. IEEE Robotics and Automation Letters **7**(3), 7209–7216 (2022) https://doi.org/10.1109/LRA.2022.3181755

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017) https://doi.org/10.48550/arXiv.1707.06347

Soliman, A., Al-Ali, A., Mohamed, A., Gedawy, H., Izham, D., Bahri, M., Erbad, A., Guizani, M.: AI-based UAV navigation framework with digital twin technology for mobile target visitation. Engineering Applications of Artificial Intelligence **123**, 106318 (2023) https://doi.org/10.1016/j.engappai.2023.106318

Song, Y., Romero, A., Müller, M., Koltun, V., Scaramuzza, D.: Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. Science Robotics **8**(82), eadg1462 (2023) https://doi.org/10.1126/scirobotics.adg1462

Song, Y., Scaramuzza, D.: Policy search for model predictive control with application to agile drone flight. IEEE Transactions on Robotics **38**(4), 2114–2130 (2022) https://doi.org/10.1109/TRO.2022.3141602

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), 1989–1995. IEEE, London, UK (2023). https://doi.org/10.1109/ICRA48891.2023.10160563

Tordesillas, J., How, J.P.: MADER: Trajectory planner in multiagent and dynamic environments. IEEE Transactions on Robotics **38**(1), 463–476 (2022) https://doi.org/10.1109/TRO.2021.3080235

Valman, S.J., Boyd, D.S., Carbonneau, P.E., Johnson, M.F., Dugdale, S.J.: An AI approach to operationalise global daily PlanetScope satellite imagery for river water masking. Remote Sensing of Environment **301**, 113932 (2024) https://doi.org/10.1016/j.rse.2023.113932

Wu, J., Ye, Y., Du, J.: Multi-objective reinforcement learning for autonomous drone navigation in urban areas with wind zones. Automation in Construction **158**, 105253 (2024) https://doi.org/10.1016/j.autcon.2023.105253

Xu, Z., Han, X., Shen, H., Jin, H., Shimada, K.: NavRL: Learning safe flight in dynamic environments. IEEE Robotics and Automation Letters **10**(4), 3668–3675 (2025)

https://doi.org/10.1109/LRA.2025.3546069

Xu, Z., Zhan, X., Xiu, Y., Suzuki, C., Shimada, K.: Onboard dynamic-object detection and tracking for autonomous robot navigation with RGB-D camera. IEEE Robotics and Automation Letters **9**(1), 651–658 (2024) https://doi.org/10.1109/LRA.2023.3334683

Xu, Z., Hasselt, H., Silver, D.: Meta-gradient reinforcement learning. In: Advances in Neural Information Processing Systems, vol. 31, 5405–5414 (2018). https://proceedings.neurips.cc/paper/7507-meta-gradient-reinforcement-learning.pdf

Yadav, M., Vashisht, B.B., Vullaganti, N., Kumar, P., Jalota, S.K., Kumar, A., Kaushik, P.: UAV-enabled approaches for irrigation scheduling and water body characterization. Agricultural Water Management **304**, 109091 (2024) https://doi.org/10.1016/j.agwat.2024.109091

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

Zhang, E., Zhang, J., Han, J., Cao, Y., Li, C., Zhang, M.: Multi-scale local–global deep optical flow network for river surface velocity measurement from drone video. Physics of Fluids **38**(4), 046604 (2026d) https://doi.org/10.1063/5.0322760

Zhang, M.: Drone-enabled remote sensing dataset of the Heihe River Basin (2024–2025). National Tibetan Plateau/Third Pole Environment Data Center (2025). https://doi.org/10.11888/Terre.tpdc.303151

Zhang, M., Guo, J., Chen, Y., Nie, X., Li, X.: Lightweight deep learning for real-time river surface velocity estimation from low-altitude drone remote sensing. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **19**, 17967–17987 (2026c) https://doi.org/10.1109/JSTARS.2026.3694061

Zhang, M., Guo, J., Chen, Y., Zhang, E., Jin, R., Nie, X., Li, X.: Enhanced deep recurrent optical flow with efficient feature encoding and channel attention for river surface velocimetry. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **19**, 5928–5947 (2026b) https://doi.org/10.1109/JSTARS.2026.3656354

Zhang, M., Liu, J., Jin, R., Li, X.: Real-time river velocity estimation with a drone-based image velocimetry system. IEEE Internet of Things Journal **13**(7), 12988–13004 (2026f) https://doi.org/10.1109/JIOT.2026.3653680

Zhang, M., Ma, L., Chen, Y., Li, X.: OF-PhyNet: A hybrid neural network method for river surface velocity measurement based on optical flow and physical constraints. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **19**, 635–649 (2026e) https://doi.org/10.1109/JSTARS.2025.3631633

Zhang, M., Xu, M., Li, X.: Deep reinforcement learning-based feature enhancement method for UAV-enabled river surface flow velocity measurement. IEEE Internet of Things Journal **13**(2), 3123–3137 (2026a) https://doi.org/10.1109/JIOT.2025.3633273

Zhang, Y., Hu, Y., Song, Y., Zou, D., Lin, W.: Learning vision-based agile flight via differentiable physics. Nature Machine Intelligence **7**, 954–966 (2025) https://doi.org/10.1038/s42256-025-01048-0

Zhuang, Y., Sun, X., Li, Y., Huai, J., Hua, L., Yang, X., Cao, X., Zhang, P., Cao, Y., Qi, L., Yang, J., El-Bendary, N., El-Sheimy, N., Thompson, J., Chen, R.: Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches. Information Fusion **95**, 62–90 (2023) https://doi.org/10.1016/j.inffus.2023.01.025

Data and code availability

The field-image archive used for the provenance and appearance analysis is available from the National Tibetan Plateau/Third Pole Environment Data Center at https://doi.org/10.11888/Terre.tpdc.303151 (Zhang 2025). Procedural scenarios are identified by deterministic generator seeds and frozen manifests. An anonymised Online Resource 1 contains the NMPC validation-selection record, all 600 paired geometry-controller rows, state-, reward-, and rule-level per-seed and summary tables, reward-weight trajectories, paired-test outputs, the 20-run finite-difference verification, derivative-path diagnostics, the field-archive manifest, special-scenario source values, configuration records, source code, automated tests, analysis scripts, and integrity checksums.

Supplementary information

Online Resource 1 is the anonymised reproducibility bundle described in the Data and code availability statement.