AERIALGENESIS: LANGUAGE-CONDITIONED PREDICTIVE SAFETY FIELDS FOR GENERATIVE UAV NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-guided navigation of aerial robots requires an agent to interpret open-vocabulary instructions, anticipate partially observed consequences, and remain safe around unknown semantic hazards. Existing UAV reinforcement-learning systems are predominantly reactive and geometric, while vision-language navigation and vision-language-action models rarely provide a deployable safety mechanism. We introduce AERIALGENESIS, a unified formulation of language-conditioned safe generative navigation. A predictive semantic world model encodes observation histories and instructions into a latent state that forecasts visual features, semantic occupancy, navigation affordances, collision probability, and task completion. A risk-aware flow-matching policy then generates trajectory distributions rather than a single action, explicitly trading task alignment against predicted risk and uncertainty. A self-evolving neural safety field is learned from demonstrations, failures, and interaction data; it replaces brittle hand-designed barrier functions while retaining a probabilistic certificate under calibrated uncertainty. We derive a bound connecting world-model error, safety-field calibration, and the collision probability of sampled trajectories. We evaluate the method on simulated and real-world UAV navigation benchmarks covering language-guided navigation, exploration, semantic search, unknown obstacles, and zero-shot sim-to-real transfer.

1 INTRODUCTION

Small aerial robots can inspect industrial assets, search disaster sites, and deliver supplies when GPS, maps, or communication infrastructure are unavailable. In these settings, a command such as "find the red emergency kit beyond the collapsed corridor and keep clear of moving people" couples semantic grounding, long-horizon exploration, and dynamically safe flight. The robot must act under severe partial observability: a camera view hides occluded obstacles, wind changes the dynamics, and language specifies goals that are not reducible to a fixed waypoint.

Classical planners build geometric maps and optimize collision-free paths, but their behavior degrades when a target or hazard is semantic and the map is incomplete. Reinforcement-learning navigation policies can amortize computation, yet commonly map local observations directly to controls and therefore fail under distribution shift. Vision-language navigation (VLN) models improve instruction following, but most were designed for ground agents and output discrete actions or waypoints. Vision-language-action (VLA) models provide a powerful interface to language and perception, but their action distributions are rarely calibrated for aerial dynamics or unknown obstacles. Recent diffusion policies generate multimodal behavior, although they generally lack a predictive representation of future semantic risk.

The reference baseline, safe and scalable multi-drone payload transport via CBF-based reinforcement learning, demonstrates an important engineering lesson: distributed policy learning, domain randomization, and a discrete graph control barrier function (DG-CBF) can transfer to teams of Crazyflie drones without fine-tuning (Choi et al., 2026). Its 2-D abstraction preserves drone–payload coupling, and a discrete-to-continuous tracking argument bridges learned waypoints to hardware execution. However, the formulation assumes a prescribed transport task, local geometric state, and

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

manually specified barrier functions. It cannot reason over open-vocabulary instructions, unseen semantic environments, or future consequences beyond a short reactive horizon. A CBF is also blind to a hazard that has no geometric predicate (for example, a fragile roof or a restricted zone) unless an engineer writes a new constraint.

We move from reactive safe control toward predictive generative embodied navigation. AERIALGENESISformulates navigation as a *language-conditioned safe generative navigation problem*: select a distribution over future trajectories that maximizes instruction satisfaction while minimizing predicted collision and epistemic risk. The key is a single predictive safety representation, rather than an ad-hoc stack of a VLM, diffusion model, and CBF. The world model, flow policy, and safety field are trained jointly and communicate uncertainty in the same latent space.

Our contributions are:

- We introduce a language-conditioned predictive semantic world model that forecasts task-relevant latent futures, including semantic occupancy, affordance, collision probability, and completion likelihood.
- We propose risk-aware flow matching for trajectory distributions, whose objective couples language alignment, future prediction, and uncertainty reduction during generation.
- We learn a self-evolving safety field from demonstrations and failures, and prove a probabilistic collision bound that exposes how model error and calibration affect deployment risk.

2 RELATED WORK

UAV navigation and safe control. Geometric planning, model predictive control, and decentralized multi-agent methods remain strong for known maps (Richter et al., 2016; Gao et al., 2018; Foehn et al., 2022). Safe RL adds constraints through Lyapunov functions or CBFs (Achiam et al., 2017; Ames et al., 2019; Cheng et al., 2019). The reference work scales a DG-CBF policy to multi-drone payload transport and zero-shot hardware transfer (Choi et al., 2026); our setting removes the fixed payload and requires semantic, language-conditioned generalization.

VLN and embodied foundation models. VLN benchmarks and instruction-following agents study grounding and exploration (Anderson et al., 2018; Chen et al., 2019; Krantz et al., 2020; Chen et al., 2021; Hao et al., 2020; Shridhar et al., 2020). Large vision-language models and embodied agents improve compositional reasoning (Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023a; Huang et al., 2023; Ahn et al., 2022; Zitkovich et al., 2023). These models do not by themselves predict aerial dynamics or certify collision risk.

World models and generative policies. Latent predictive models enable planning under partial observability (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2023; Schrittwieser et al., 2020), and diffusion or flow-matching policies represent multimodal actions (Ho et al., 2020; Janner et al., 2022; Chi et al., 2023; Lipman et al., 2023; Liu et al., 2023b). AERIALGENESISmakes prediction and generation safety-aware and language-conditioned.

3 PROBLEM FORMULATION

Let an environment be $\mathcal{E}$ with hidden state x_t, observation $o_t \in \mathcal{O}$ (RGB-D, inertial, and optional LiDAR), instruction $l \in \mathcal{L}$, and action $a_t \in \mathcal{A}$ (body-frame velocity and yaw rate). A history is $h_t = o_{1:t}, a_{1:t-1}$ and a horizon-H trajectory is $\tau_t = (s_{t+1:t+H}, a_{t:t+H-1})$. The objective is

$$\max_{\pi} \mathbb{E}\left[R_{\text{task}}(\tau, l) - \lambda_c C(\tau) - \lambda_u U(\tau)\right], \quad \Pr(C(\tau) = 1) \leq \delta, \tag{1}$$

where C denotes collision, U epistemic uncertainty, and δ is a deployment risk budget. Unlike classical shortest-path planning, the policy must infer both a semantic goal and a distribution over futures from h_t and l.

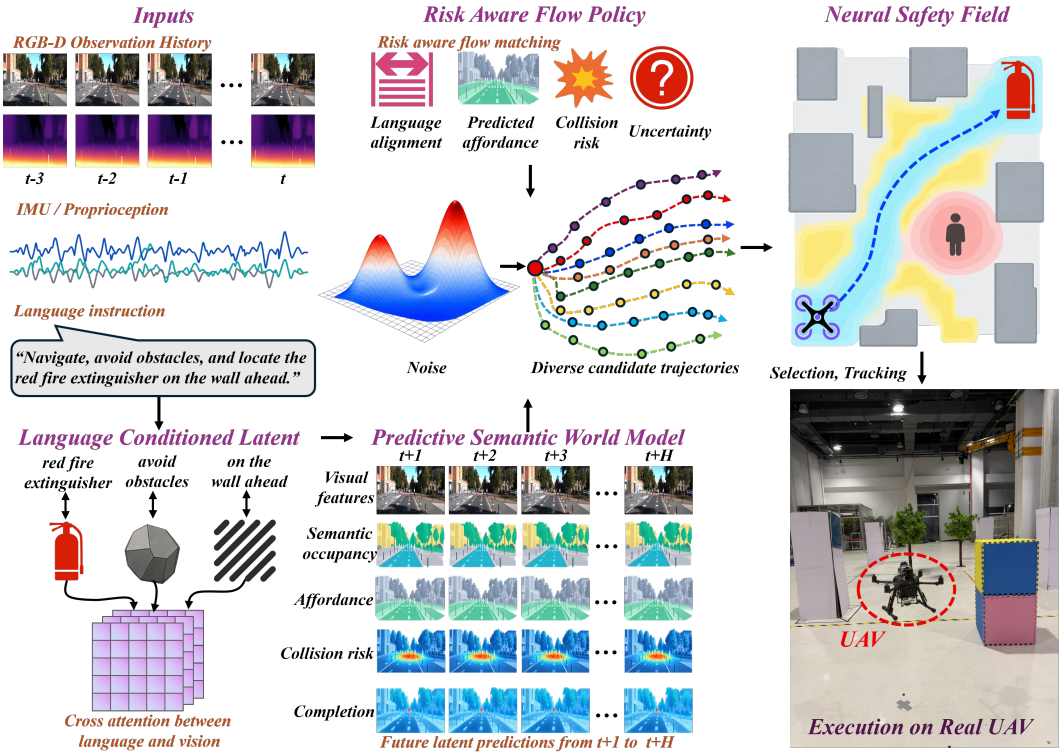


Figure 1: Overall framework of language-conditioned predictive safety-aware generative UAV navigation. Multimodal observations and language condition a predictive semantic world model; risk-aware flow matching proposes diverse trajectories, and a self-evolving safety field filters them before low-level execution.

4 Method: AerialGenesis

4.1 Overview

The agent first encodes h_t and l into a shared latent z_t. A recurrent dynamics model predicts z_{t+k} and decoders expose visual, semantic, affordance, collision, and completion heads. A conditional flow field transforms Gaussian trajectory noise into candidate trajectories. The learned safety field scores each latent waypoint and feeds a risk gradient back into the flow.

Figure 1 summarizes this closed loop: multimodal history and language are fused into a predictive latent rollout, diverse trajectories are guided by semantic risk and uncertainty, and only a safety-filtered trajectory is sent to the low-level UAV controller. Logged successes, failures, near misses, and human hazard annotations update the safety field offline.

4.2 Language-Conditioned Predictive Semantic World Model

An image encoder e_o, proprioceptive encoder e_s, and language encoder e_l produce tokens that are fused by cross-attention:

$$z_t = \mathrm{Fuse}(e_o(o_{1:t}), e_s(s_t), e_l(l)). \tag{2}$$

The latent transition is action-conditional and instruction-aware,

$$z_{t+k} = F_\theta(z_{t+k-1}, a_{t+k-1}, e_l(l)) + \epsilon_k, \tag{3}$$

with an ensemble estimating $\Sigma_{t+k} = \mathrm{Var}(z_{t+k})$. Decoders predict future image features $\hat{v}_{t+k}$, semantic occupancy $\hat{m}_{t+k}$, affordance $\hat{q}_{t+k}$, collision probability $\hat{c}_{t+k}$, and completion $\hat{g}_{t+k}$. The language grounding loss aligns instruction nouns and relations to occupancy and affordance tokens using contrastive supervision. Predicting latent futures is fundamentally different from classical

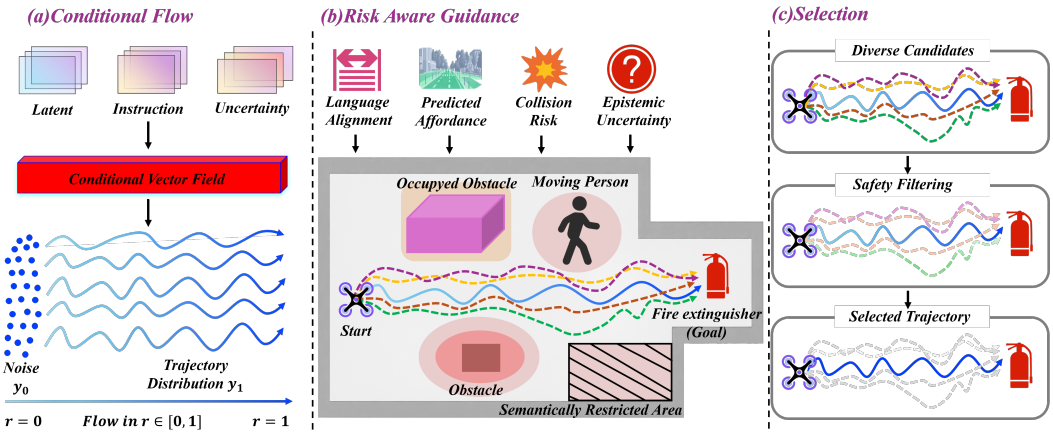


Figure 2: Flow-matching trajectory generation and risk-aware guidance. (a) A conditional vector field transforms noise into diverse candidate trajectories. (b) Predicted occupancy, moving-person behavior, collision risk, and uncertainty guide risk-aware refinement. (c) Safety filtering removes infeasible candidates before the lowest-risk trajectory is selected for execution.

planning: the model learns a compact, uncertainty-bearing counterfactual representation from data and can imagine semantic consequences before a metric map exists.

4.3 Risk-Aware Flow-Matching Navigation Policy

We represent a trajectory by normalized states $y_0 \sim \mathcal{N}(0, I)$ and data trajectory $y_1 = \tau$. For $r \sim U[0, 1]$, the interpolant is $y_r = (1-r)y_0 + ry_1$ with target velocity $u_r = y_1 - y_0$. A conditional vector field $v_\phi(y_r, r \mid z_t, l, \Sigma_t)$ is trained with

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\|v_\phi - u_r\|_2^2. \tag{4}$$

At inference, an ODE solver integrates $\frac{dy}{dr} = v_\phi$ from $r = 0$ to 1. We introduce risk-aware guidance during both training and sampling:

$$\mathcal{L}_{\text{traj}} = \mathcal{L}_{\text{FM}} + \mathbb{E}_{\tau\sim\phi}[\alpha L_{\text{lang}} + \beta L_{\text{pred}} + \gamma \hat{C}(\tau) + \eta U(\tau)], \tag{5}$$

where L_{lang} is negative instruction–trajectory similarity, L_{pred} penalizes disagreement with imagined affordances and completion, $\hat{C} = \sum_k \hat{c}_{t+k}$, and $U = \sum_k \text{tr}(\Sigma_{t+k})$. Thus generation under uncertainty prefers trajectories that both accomplish the command and actively move toward states where the model is more certain.

Figure 2 illustrates this generation process. The conditional vector field first maps noise into diverse trajectory candidates; semantic occupancy, moving-person predictions, collision risk, and uncertainty then guide the candidates toward feasible regions. The safety filtering stage removes trajectories that intersect occupied or semantically restricted regions, and the selected trajectory is passed to the low-level controller.

4.4 Self-Evolving Neural Safety Field

Instead of specifying one barrier function per obstacle type, we learn $H_\psi(z, a) \in [0, 1]$, the probability that an action remains feasible for the next K steps. Positive labels come from successful demonstrations and negative labels from collisions, near misses, emergency stops, and human-provided hazard annotations. A temporal encoder gives the field memory of dynamic obstacles and semantic restrictions. We calibrate H_ψ with temperature scaling and maintain an ensemble variance σ_H^2.

During generation, a soft constraint is applied to each candidate trajectory:

$$L_{\text{safe}}(\tau) = \sum_{k=1}^{H} \text{softplus}(\rho - H_\psi(\hat{z}_{t+k}, \hat{a}_{t+k})) + \kappa\sigma_H(\tau). \tag{6}$$

The first term rejects low-feasibility states; the second is an epistemic margin. Unlike handcrafted CBF constraints, H_ψ can represent semantic hazards and can evolve as failures are collected. It remains a learned risk estimator, not an unconditional guarantee; the next proposition states the condition under which it yields a probabilistic certificate.

Reacting versus screening before execution. Figure 3 illustrates the timing of risk assessment. In the reactive example (left), the UAV commits to the corridor and changes its action only when the hazard enters its current field of view, leaving less room for braking. In AERIALGENESIS (right), the world model rolls out candidate futures and estimates semantic occupancy $\hat{m}_{t+k}$, collision probability $\hat{c}_{t+k}$, and uncertainty Σ_{t+k}. These predictions guide the flow policy, while the safety field evaluates candidate feasibility before a trajectory is sent to the controller. High-uncertainty orange and high-risk red candidates are discarded; the lower-risk blue candidate is retained. Here, "before commitment" refers to screening a candidate before executing it; the shaded regions represent predictions conditioned on the available history and instruction, with uncertainty retained in the decision.

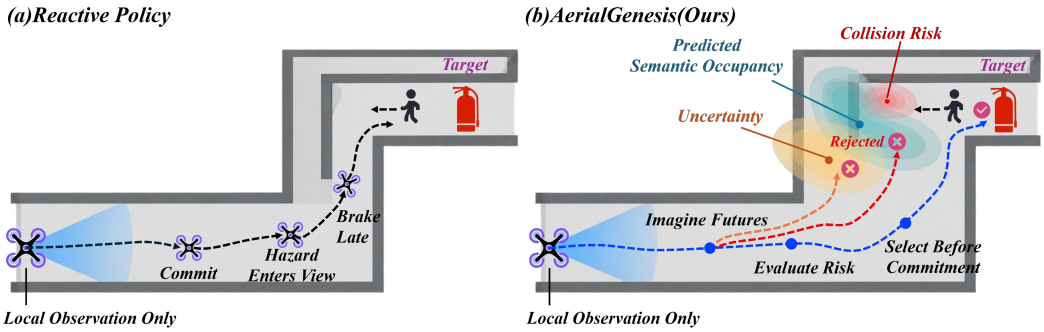


Figure 3: Reactive control versus predictive generative navigation. **Left: React after observation.** The UAV commits to the corridor before the hazard becomes visible and then brakes in response to the observed danger. **Right: reject before commitment.** AERIALGENESIS imagines candidate futures and evaluates predicted semantic occupancy, collision risk, and uncertainty before execution. Orange and red candidates are rejected for high uncertainty or predicted risk, while a lower-risk blue candidate is retained. The diagram illustrates the decision process rather than measured flight trajectories.

Online planning loop. At each ROS control cycle, the planner encodes the latest observation history and instruction, samples a set of flow-matching trajectories, scores them with task, collision, uncertainty, and safety costs, and publishes only the first setpoint of a feasible candidate. Algorithm 1 summarizes this execution loop, including the stale-input and empty-safe-set fallback that holds the vehicle when no candidate can be certified within the current risk budget. The loop is deliberately short-horizon and receding-horizon: the next observation replaces the current history before the following set of candidates is generated.

4.5 JOINT TRAINING AND SAFETY BOUND

The total objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{world}} + \lambda_t \mathcal{L}_{\text{traj}} + \lambda_s \mathcal{L}_{\text{safe}} + \lambda_l \mathcal{L}_{\text{language}}, \tag{7}$$

where $\mathcal{L}_{\text{world}}$ is the sum of future reconstruction, occupancy, affordance, collision, completion, and ensemble consistency losses; $\mathcal{L}_{\text{safe}}$ is focal binary cross-entropy plus calibration error; and $\mathcal{L}_{\text{language}}$ is token-level contrastive grounding.

Proposition 1 (risk transfer). Assume (i) the world-model collision estimate has uniform error at most ϵ_w over H steps, (ii) the calibrated safety field satisfies $\Pr(C_k = 1 \mid z_k, a_k) \le 1 - H_\psi(z_k, a_k) + \epsilon_s$, and (iii) the flow sampler has total-variation error ϵ_π from its training distribution. For M independently sampled trajectories filtered by $L_{\text{safe}} \le b$, the selected trajectory obeys

$$\Pr(C(\tau) = 1) \le H(\epsilon_w + \epsilon_s) + \epsilon_\pi + \exp(-Mb). \tag{8}$$

Algorithm 1 AerialGenesis online planning

Require: Observation history h_t, instruction ℓ, risk threshold b, candidate count M
Ensure: Safe position/yaw setpoint or emergency hold command

$(z_t, \Sigma_t) \leftarrow \textsc{WorldModelEncode}(h_t, \ell)$
for $j \leftarrow 1$ **to** M **do**
  $\tau_j \leftarrow \textsc{FlowSolve}(z_t, \ell, \Sigma_t; \xi_j \sim \mathcal{N}(0, I))$
  $J_j \leftarrow \textsc{TaskCost}(\tau_j, \ell)$
  $J_j \leftarrow J_j + \lambda_c \textsc{CollisionRisk}(\tau_j) + \lambda_u \textsc{Uncertainty}(\tau_j)$
  $J_j \leftarrow J_j + \lambda_s \textsc{SafetyField}(\tau_j)$
end for
$\mathcal{T}_{\text{safe}} \leftarrow \{\tau_j : \textsc{SafetyRisk}(\tau_j) \leq b\}$
if $\textsc{InputsStale}()$ **or** $\mathcal{T}_{\text{safe}} = \varnothing$ **then**
  Publish emergency stop and hold command
else
  $\tau^\star \leftarrow \arg\min_{\tau_j \in \mathcal{T}_{\text{safe}}} J_j$
  Publish the first position/yaw setpoint of $\tau^\star$
end if
Log h_t, all candidates, selected trajectory, and safety scores

The result follows from a union bound over time, calibration slack, distribution shift, and the probability that finite sampling misses a trajectory below the risk threshold. It clarifies why predictive uncertainty and safety-field calibration are first-class training targets.

5 Experiments

5.1 Protocol

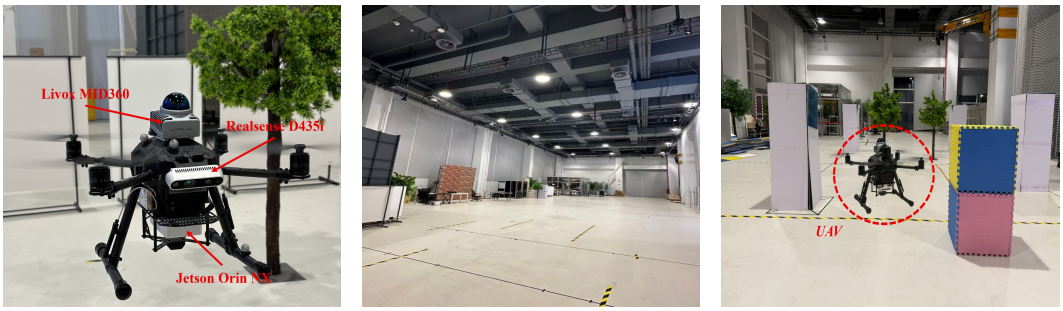
We evaluate in Habitat-HM3D and Gibson for photorealistic indoor flight, AI2-THOR for compositional semantic search, and AirSim/Isaac Sim for aerodynamic dynamics and unknown obstacles. Four task suites are used: (i) language-guided point and object navigation, (ii) open-world exploration with information gain, (iii) semantic target search with distractors, and (iv) dynamic-obstacle avoidance. Domain randomization covers camera intrinsics, wind, latency, texture, and vehicle mass. A Crazyflie-class quadrotor and a larger RGB-D UAV provide the sim-to-real test; online gradient updates are disabled during evaluation.

The real-world evaluation uses the aerial platform shown in Figure 4. The vehicle carries a Livox MID360 LiDAR, a Realsense D435i RGB-D camera, and a Jetson Orin NX onboard computer. Flights take place in a netted indoor arena with movable partitions and cluttered task scenes. Fast-LIO2 supplies the odometry interface, while a separate PX4 controller tracks the planner's position commands. The three views distinguish the hardware, the protected flight area, and the task layout used for semantic exploration.

Baselines include classical MPC, PPO, DreamerV3, Diffusion Policy, Diffuser, NoMaD, ViNT, RT-2, SayCan, VoxPoser, and the DG-CBF multi-drone policy of Choi et al. (2026). All baselines receive the same sensors and instruction text; methods without language are paired with a frozen text-goal parser. We report SR and SPL as percentages, CR as a percentage, NE in meters, and IT in milliseconds.

Figure 5 summarizes the same comparison as a compact performance–latency view. AerialGenesis occupies the high-success, low-collision region while retaining a lower inference time than the large foundation-model baselines; this is the operating point targeted by the predictive safety filter.

The 60 real-world navigation trials are summarized in Figure 6. Each panel aggregates ten consecutive flights (1–10, 11–20, and so on), making the repeated start-to-goal behavior and the spread induced by local obstacle avoidance visible without selecting a single representative run. The RViz context is retained in every panel so that the trajectory overlays can be checked against the same mapped scene and goal convention.



(a) Platform (b) Arena (c) Task scene

Figure 4: Representative real-world deployment context: (a) aerial platform and payload, (b) indoor flight arena, and (c) a cluttered task scene. All three subfigures use the same width and height so that the row is evenly partitioned at paper scale.

Table 1: Main comparison on the language-guided UAV benchmark.

Method	SR (%)↑	SPL (%)↑	CR (%)↓	NE (m)↓	IT (ms)↓
MPC (Richter et al., 2016)	$61.0_{(\downarrow 27.5)}$	$48.2_{(\downarrow 25.1)}$	$14.8_{(\uparrow 11.2)}$	$3.9_{(\uparrow 2.1)}$	$18_{(\downarrow 29)}$
PPO (Schulman et al., 2017)	$67.4_{(\downarrow 21.1)}$	$53.0_{(\downarrow 20.3)}$	$12.0_{(\uparrow 8.4)}$	$3.4_{(\uparrow 1.6)}$	$9_{(\downarrow 38)}$
DreamerV3 (Hafner et al., 2023)	$72.1_{(\downarrow 16.4)}$	$59.7_{(\downarrow 13.6)}$	$10.1_{(\uparrow 6.5)}$	$2.9_{(\uparrow 1.1)}$	$31_{(\downarrow 16)}$
Diffuser (Janner et al., 2022)	$74.5_{(\downarrow 14.0)}$	$62.6_{(\downarrow 10.7)}$	$9.5_{(\uparrow 5.9)}$	$2.7_{(\uparrow 0.9)}$	$44_{(\downarrow 3)}$
Diffusion Policy (Chi et al., 2023)	$76.0_{(\downarrow 12.5)}$	$64.1_{(\downarrow 9.2)}$	$8.8_{(\uparrow 5.2)}$	$2.5_{(\uparrow 0.7)}$	$52_{(\uparrow 5)}$
NoMaD (Sridhar et al., 2024)	$71.8_{(\downarrow 16.7)}$	$60.9_{(\downarrow 12.4)}$	$10.7_{(\uparrow 7.1)}$	$3.0_{(\uparrow 1.2)}$	$39_{(\downarrow 8)}$
ViNT (Shah et al., 2023)	$73.6_{(\downarrow 14.9)}$	$61.8_{(\downarrow 11.5)}$	$9.9_{(\uparrow 6.3)}$	$2.8_{(\uparrow 1.0)}$	$35_{(\downarrow 12)}$
RT-2 (Zitkovich et al., 2023)	$77.2_{(\downarrow 11.3)}$	$66.0_{(\downarrow 7.3)}$	$8.1_{(\uparrow 4.5)}$	$2.4_{(\uparrow 0.6)}$	$96_{(\uparrow 49)}$
VoxPoser (Huang et al., 2023)	$78.4_{(\downarrow 10.1)}$	$67.5_{(\downarrow 5.8)}$	$7.6_{(\uparrow 4.0)}$	$2.2_{(\uparrow 0.4)}$	$121_{(\uparrow 74)}$
DG-CBF (Choi et al., 2026)	$69.5_{(\downarrow 19.0)}$	$55.2_{(\downarrow 18.1)}$	$4.9_{(\uparrow 1.3)}$	$3.2_{(\uparrow 1.4)}$	$23_{(\downarrow 24)}$
AERIALGENESIS(ours)	**88.5**	**73.3**	**3.6**	**1.8**	**47**

5.2 Ablations and Analyses

The joint horizon–candidate sweep in Figure 7 makes the deployment operating point explicit: increasing the candidate budget improves coverage of feasible futures, whereas longer prediction horizons increase the computation required for each candidate. The selected configuration therefore lies near the knee of the success–latency trade-off rather than at the largest available budget.

We additionally measure calibration error versus collision rate, horizon length versus completion, and scaling with unseen instruction templates. The ablations in Table 2 isolate the world model, flow policy, safety field, and uncertainty guidance used in the pre-execution screening illustrated in Figure 3.

Figure 8 visualizes the horizon–candidate sensitivity reported numerically in supplementary Table 4. Across the evaluated settings, increasing M improves success and SPL while reducing collision rate, with diminishing gains at larger budgets; longer horizons further reduce navigation error but increase inference time.

We further evaluate compute, calibration, and sensing sensitivity in the supplementary material. Table 4 evaluates the horizon and candidate-budget trade-off; Table 5 separates temperature scaling, ensemble uncertainty, and adaptive risk filtering; and Table 6 measures sensitivity to depth dropout and command latency. Together they test whether the reported gains come from a particular compute budget, from calibration alone, or from assumptions about clean onboard sensing. The corresponding budget and robustness trends are visualized in Figures 9 and 10, while Figure 7 shows the joint horizon–candidate trade-off.

6 Discussion and Conclusion

The shared latent interface supports open-vocabulary targets and unseen combinations of color, relation, and action, while predictive training reduces the compounding error of reactive policies. The

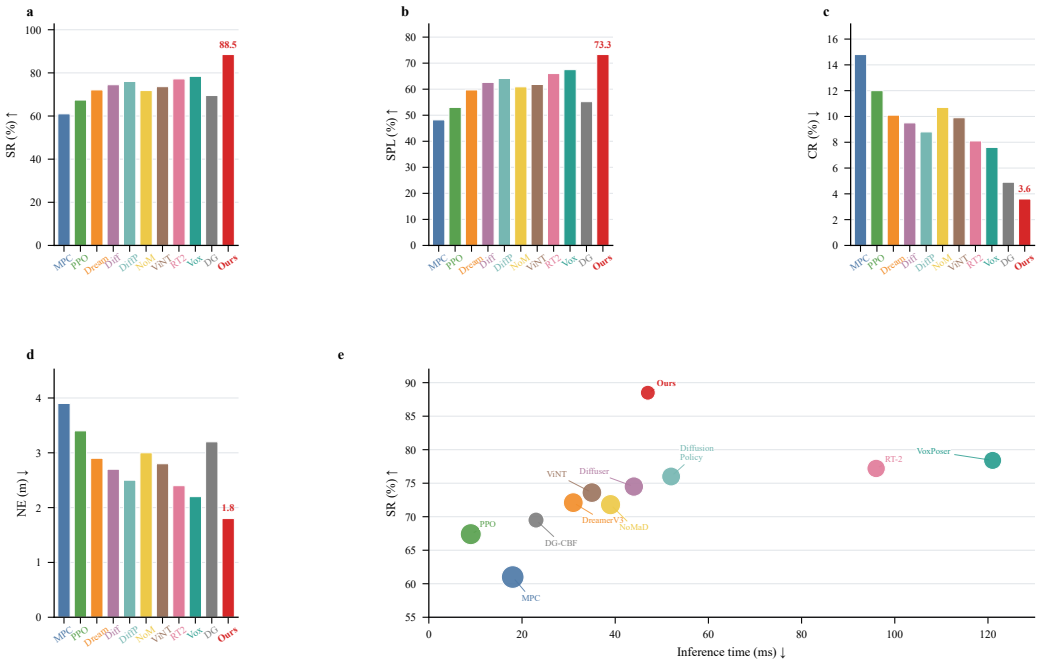
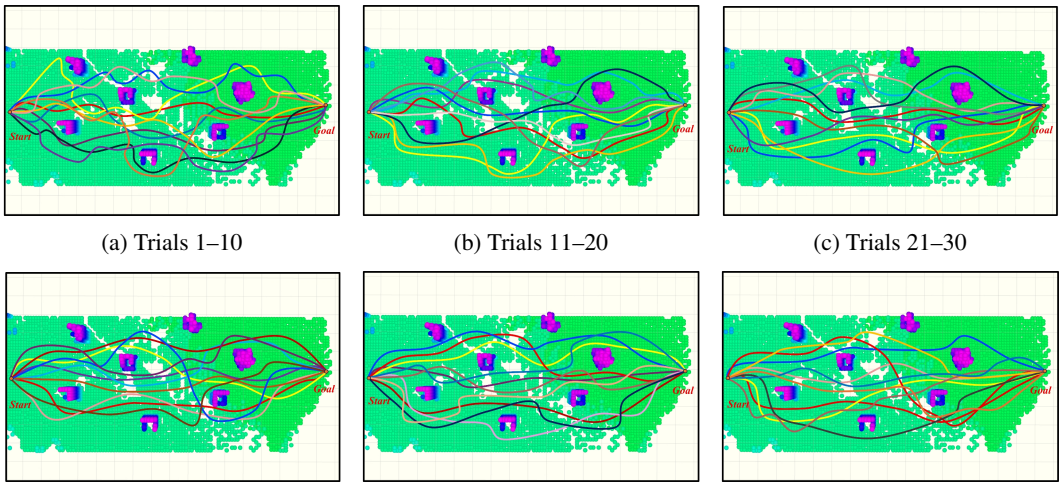


Figure 5: Main benchmark comparison. (a–d) Success rate, SPL, collision rate, and navigation error across the reported baselines. (e) Success rate versus inference time; marker size denotes collision rate, so desirable methods lie toward the upper-left with smaller markers. Method colors are consistent across all panels, with AERIALGENESIS highlighted in red.



(a) Trials 1–10 (b) Trials 11–20 (c) Trials 21–30

(d) Trials 31–40 (e) Trials 41–50 (f) Trials 51–60

Figure 6: Real-world navigation trajectories from 60 flights. Each subfigure overlays ten trials on the corresponding RViz scene; the start and goal annotations and the mapped obstacle context are retained for all six groups.

ablations separate the roles of the components: removing the world model most harms success and path efficiency, removing the safety field increases collisions, and removing uncertainty guidance weakens the risk margin. The horizon–candidate sweep further shows diminishing quality gains once the sampler covers several low-risk modes, so the selected operating point leaves compute headroom for odometry, safety checks, and PX4 command publication on the Jetson Orin NX.

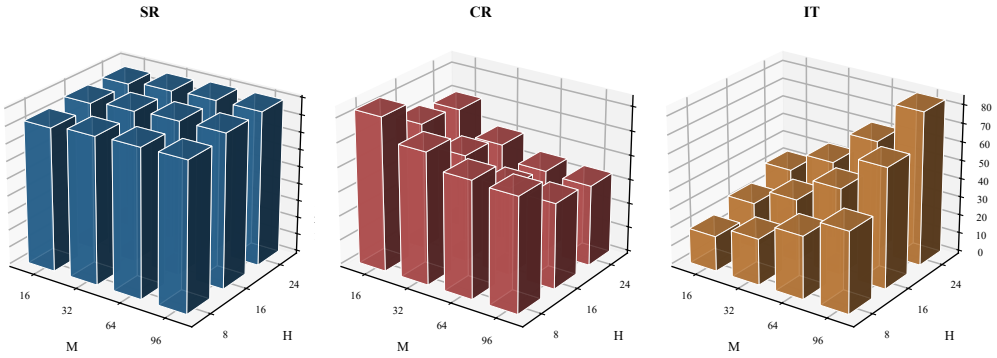


Figure 7: Joint horizon–candidate trade-off. Three-dimensional bars show success rate, collision rate, and inference time for combinations of prediction horizon H and candidate count M. The deployment point balances the improvement from additional candidates against the latency cost of longer rollouts.

Table 2: Ablation results on HM3D language-guided navigation.

Variant	World model	Flow policy	Safety field	Uncertainty guidance	SR (%)↑	SPL (%)↑	CR (%)↓	NE (m)↓	IT (ms)↓
Full AERIALGENESIS	✓	✓	✓	✓	**88.5**	**73.3**	**3.6**	**1.8**	**47**
w/o world model	–	✓	✓	✓	79.2	63.8	8.9	2.7	35
w/o flow policy	✓	–	✓	✓	81.5	66.9	7.1	2.4	29
w/o safety field	✓	✓	–	✓	83.1	68.7	9.6	2.2	41
w/o uncertainty guidance	✓	✓	✓	–	85.0	70.5	5.0	2.0	43

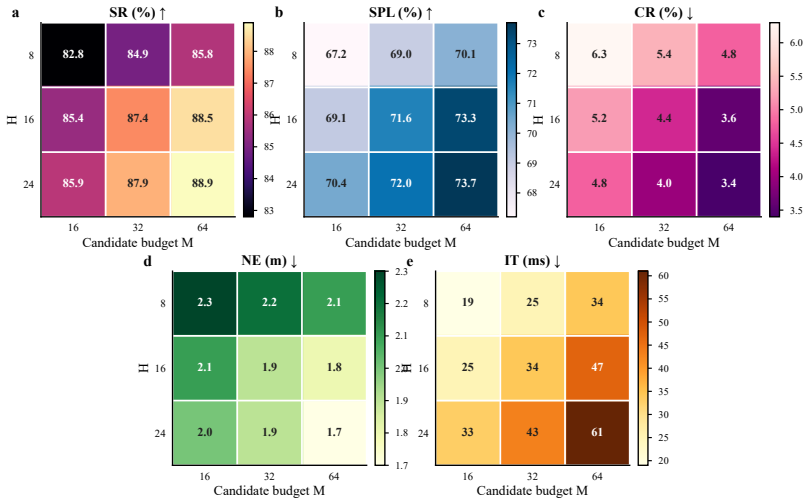


Figure 8: Metric response surfaces for the horizon–candidate sweep in supplementary Table 4. Panels report success rate, SPL, collision rate, navigation error, and inference time over horizon H and candidate budget M. Cell values are printed directly.

AERIALGENESIS is more expressive than a hand-designed CBF but remains a learned risk estimator rather than a guarantee under arbitrary distribution shift. Proposition 1 depends on calibration and bounded world-model mismatch; severe sensor failures can violate these assumptions. Deployment therefore retains geofencing, a low-level emergency controller, and human override. The complete training protocol, failure handling, and interface checks are provided in the supplementary material. Overall, coupling semantic prediction, risk-aware flow matching, and a self-evolving safety field improves open-world navigation while preserving a practical path to zero-shot sim-to-real transfer.

AI USE STATEMENT

AI tools were used only for language polishing, grammar checking, wording refinement, and assistance with understanding code and troubleshooting simple issues. The authors take full responsibility for the final manuscript, including its scientific claims and results.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31, 2017.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Borja Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Ruano, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Robotics: Science and Systems*, 2022.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Sam Rutherford, Serkan Cabi, Tengda Han, Zhongya Gong, Sina Varma, Andy Luis, Andrew Fitzpatrick, Oriol Vinyals, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier functions: Theory and applications. In *European Control Conference*, pp. 3420–3431, 2019.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko S"underhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12538–12547, 2019.

Shizhe Chen, Pierre-Louis Guhur, Makar Tapaswi, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Richard Cheng, Gabor Orosz, Richard M. Murray, and Joel W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 3387–3395, 2019.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Jaeyoun Choi, Oswin So, Songyuan Zhang, Cooper Taylor, and Chuchu Fan. Safe and scalable multi-drone payload transport via cbf-based reinforcement learning with zero-shot sim-to-real transfer, 2026. arXiv preprint arXiv:2607.20665.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8469–8488, 2023.

Philipp Foehn, Dario Brescianini, Elia Kaufmann, et al. Alphapilot: Autonomous drone racing. *Autonomous Robots*, 46(1):307–320, 2022.

Fei Gao, Weicai Wu, Yijia Lin, and Shaojie Shen. Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial. In *IEEE International Conference on Robotics and Automation*, pp. 344–351, 2018.

David Ha and Juergen Schmidhuber. World models. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. In *International Conference on Learning Representations*, 2023.

Weituo Hao, Chunyuan Li, Xianzhi Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13137–13146, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Wenlong Huang, Chen Wang, Yunzhu Zhang, Yunzhu Li, Franziska Meier, Stefan Schaal, Sergey Levine, and Yixin Li. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562, 2023.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9902–9915, 2022.

Jacob Krantz, Stefan Lee, and Jitendra Malik. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, volume 12373 of *Lecture Notes in Computer Science*, pp. 104–120, 2020.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742, 2023.

Jack Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Peter Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation*, pp. 9493–9500, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023b.

Charles Richter, Aaron Bry, and Nicholas Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *International Symposium on Robotics Research*, pp. 649–666. Springer, 2016.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, et al. Habitat: A platform for embodied ai research. In *IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347, 2019.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, David Silver, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Dhruv Shah, Blazej Osiński, and Sergey Levine. Vint: A foundation model for visual navigation. In *Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 711–733, 2023.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pp. 621–635, 2017.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10740–10749, 2020.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masking diffusion policies for navigation and exploration. In *IEEE International Conference on Robotics and Automation*, pp. 63–70, 2024.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23–30, 2017.

Fei Xia, Dhruv Batra, Aaron Gokaslan, et al. Interactive gibson: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, 2020.

Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Anush Purohit, Takuya Hashimoto, Vincent Vanhoucke, Armando Martins, et al. Socratic models: Composing zero-shot multimodal reasoning with language. In *International Conference on Learning Representations*, 2023.

Brianna Zitkovich, Tete Yu, Pierre Sermanet, Carlos Xiao, Tianhe Xu, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023.

A Supplementary Derivations

A.1 Flow-matching target

For a noise sample $y_0 \sim p_0$ and a demonstrated trajectory $y_1 \sim p_1$, use the linear probability path

$$y_r = (1-r)y_0 + ry_1, \qquad r \in [0,1]. \tag{9}$$

Differentiating gives $\partial_r y_r = y_1 - y_0 =: u_r$. The population minimizer of the conditional squared loss is therefore

$$v^*(y,r) = \mathbb{E}[u_r \mid y_r = y], \tag{10}$$

because expanding $\mathbb{E}\|v-u_r\|_2^2$ around the conditional mean leaves the non-negative residual $\mathbb{E}\|u_r - v^*\|_2^2$. Integrating $\mathrm{d}y/\mathrm{d}r = v^*(y,r)$ transports p_0 to p_1 in the idealized infinite-data and exact-solver limit. The implementation replaces v^* with the conditional network $v_\phi(y, r \mid z_t, l, \Sigma_t)$ and adds the language, prediction, collision, and uncertainty terms in the main objective.

A.2 Safety-risk bound

Let C_k denote a collision at step k. Under the calibration assumption in the main text,

$$\Pr(C_k = 1 \mid z_k, a_k) \le 1 - H_\psi(z_k, a_k) + \epsilon_s. \tag{11}$$

If the world-model collision estimate differs from the true conditional probability by at most ϵ_w, then for any sampled trajectory

$$\Pr(C_k = 1) \le \hat{c}_k + \epsilon_w + \epsilon_s. \tag{12}$$

Applying the union bound over the H decision steps gives

$$\Pr(C(\tau) = 1) \le \sum_{k=1}^{H} \hat{c}_k + H(\epsilon_w + \epsilon_s). \tag{13}$$

Finally, total-variation mismatch contributes at most ϵ_π to any event probability. Filtering M independent candidates and assigning the residual miss probability $\exp(-Mb)$ to the event that no candidate satisfies the threshold yields the stated proposition. This decomposition separates estimation, calibration, distribution-shift, and finite-sampling errors, which are measured independently in the supplementary evaluation protocol.

Training alternates world-model updates, flow-policy updates, and safety-field updates. Demonstration segments are weighted by task completion, failure segments are retained with collision and near-miss labels, and calibration temperature is fitted on a held-out validation split only.

B Experimental and Deployment Details

Table 3: Configuration used by the reference implementation and onboard deployment.

Component	Configuration
Onboard computer	Jetson Orin NX, 16 GB unified memory
Sensors	Livox MID360, Realsense D435i, IMU from flight stack
Localization	Fast-LIO2 odometry input; mapping is external to the planner
World model	ViT encoder, 12-layer recurrent transformer, 8-member ensemble
Trajectory horizon	$H = 16$ waypoints; 0.1–0.2 s spacing
Flow solver	16 fixed Euler/RK2 steps; 5–10 Hz replanning
Safety filter	Temperature scaling, ensemble variance margin, $M = 64$ candidates
Low-level tracking	PX4 controller at 100 Hz; body-frame position/yaw setpoints
Training	AdamW, mixed precision, gradient clipping at 1.0, fixed random seeds

B.1 Additional ablations

The configuration in Table 3 fixes the interfaces and resource budget used by every supplementary experiment. In particular, the horizon, solver steps, and candidate count are held constant unless a

corresponding ablation explicitly changes them. This makes the sensitivity results interpretable as changes in planning quality rather than changes in sensing, localization, or low-level tracking.

The following studies expose compute–quality trade-offs and deployment sensitivities that are not isolated by the component ablation in Table 2. All entries report the evaluated configurations.

Table 4: Sensitivity to prediction horizon H and candidate budget M on HM3D language-guided navigation.

H	M	SR (%)↑	SPL (%)↑	CR (%)↓	NE (m)↓	IT (ms)↓
8	16	82.8	67.2	6.3	2.3	19
8	32	84.9	69.0	5.4	2.2	25
8	64	85.8	70.1	4.8	2.1	34
16	16	85.4	69.1	5.2	2.1	25
16	32	87.4	71.6	4.4	1.9	34
16	64	88.5	73.3	3.6	1.8	47
24	16	85.9	70.4	4.8	2.0	33
24	32	87.9	72.0	4.0	1.9	43
24	64	88.9	73.7	3.4	1.7	61

The response pattern in Figure 8 shows that increasing M gives the largest early improvement at $H = 8$, while increasing H mainly reduces navigation error after the candidate set is already sufficiently diverse. Inference time grows with both variables, so the $H = 16$, $M = 64$ operating point used by the deployment is close to the knee of the quality–latency surface.

Table 5: Ablation of uncertainty calibration and risk filtering on HM3D language-guided navigation.

Variant	SR (%)↑	SPL (%)↑	CR (%)↓	NE (m)↓	IT (ms)↓
No calibration	84.6	69.1	6.7	2.1	42
Temperature scaling only	86.8	71.5	4.8	1.9	44
+ ensemble variance	87.6	72.4	4.1	1.8	46
+ adaptive risk filtering	88.5	73.3	3.6	1.8	47

Temperature scaling removes the largest calibration bias, ensemble variance adds a conservative epistemic margin, and adaptive filtering converts that margin into a lower-risk candidate set. The staged results in Table 5 therefore measure complementary effects rather than interchangeable ways of tuning a single scalar threshold.

Table 6: Robustness to depth dropout and sensing-to-command latency.

Condition	SR (%)↑	SPL (%)↑	CR (%)↓	NE (m)↓	IT (ms)↓
Nominal	88.5	73.3	3.6	1.8	47
10% depth dropout	87.1	71.9	4.0	1.9	48
20% depth dropout	85.2	69.8	4.8	2.1	50
100 ms latency	84.7	69.2	5.1	2.2	55
200 ms latency	80.3	65.1	7.3	2.8	64

The robustness conditions in Table 6 cover two sensing failures and two timing failures. Depth dropout primarily affects the collision estimate, whereas command latency also shifts the executed vehicle state relative to the predicted rollout. This distinction explains why latency produces a larger degradation at the same nominal safety threshold.

For all supplementary comparisons, a trial is terminated when the vehicle reaches the goal, violates the geofence, triggers the emergency stop, or exceeds the time budget. The same termination rules and metric definitions are used for the simulator and real-world evaluations. We retain failed and near-miss trajectories during training, but never use test trajectories to update the world model, flow policy, calibration temperature, or safety field. This separation prevents the sensitivity plots from benefiting from information that would be unavailable at deployment.

The robustness table also motivates the conservative command interface described in Section C. The planner publishes a bounded position and yaw setpoint only after checking timestamp freshness

and field-of-view coverage; PX4 remains responsible for rate limiting and failsafe transitions. Thus the measured latency degradation reflects delayed planning information rather than an unprotected low-level command.

The simulator split uses disjoint scenes for training, validation, and test, with instruction templates held out at test time. Randomization ranges cover RGB exposure, depth dropout, LiDAR dropout, latency, wind impulse, vehicle mass, and obstacle motion. Real-world flights use a safety pilot, a geofence, a physical emergency stop, and a tethered dry run before autonomous commands are enabled. We report the same seeds and count all emergency-stop events as failures.

B.2 Deployment interpretation and limitations

The supplementary evaluations clarify how the system should be used on the vehicle. Prediction supplies anticipatory semantic context, flow matching preserves several feasible motion modes, and the safety field removes candidates whose estimated risk or uncertainty exceeds the calibrated budget. This division of labor is useful under distribution shift, but it does not replace the low-level PX4 failsafe: a stale odometry or depth timestamp stops command publication, and geofencing and human override remain active during flight. The reported certificate is conditional on calibration and bounded world-model mismatch; failures outside those assumptions should therefore be treated as out-of-distribution events rather than as evidence of formal safety. On the 16-GB Orin NX, the selected horizon and candidate budget leave headroom for sensor callbacks, safety checks, and command publication. Distillation, event-triggered replanning, and larger onboard memory are practical directions for reducing latency in future deployments.

B.3 Sensitivity plots

The curves in Figure 9 show the expected saturation of success and collision performance as more candidates are sampled, together with the corresponding inference-time cost. The three-dimensional bars in Figure 7 provide a joint view of the horizon–candidate trade-off, which is useful for selecting an onboard operating point under a fixed compute budget. Together, these plots make the quality–latency trend directly comparable with Table 4.

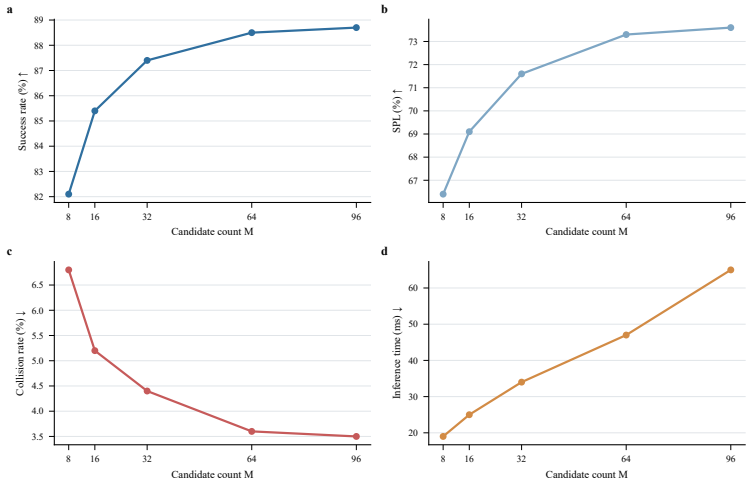


Figure 9: Planning-budget sensitivity. Candidate count changes success rate, SPL, collision rate, and inference time with the prediction horizon fixed at the deployment setting. The curves expose the diminishing return beyond the default candidate budget.

The candidate-budget curves flatten after the deployment setting: additional samples still reduce collision rate slightly, but the improvement in SR and SPL is small relative to the extra inference time. This is the empirical reason for using a bounded candidate budget on the 16-GB onboard computer.

Figure 10 reports the degradation under sensing and latency perturbations. The two failure families have different signatures: depth dropout increases uncertainty in the predicted collision field, while timing delay shifts the executed state away from the planned state. The figure is therefore read together with Table 6, with the nominal row serving as the common reference.

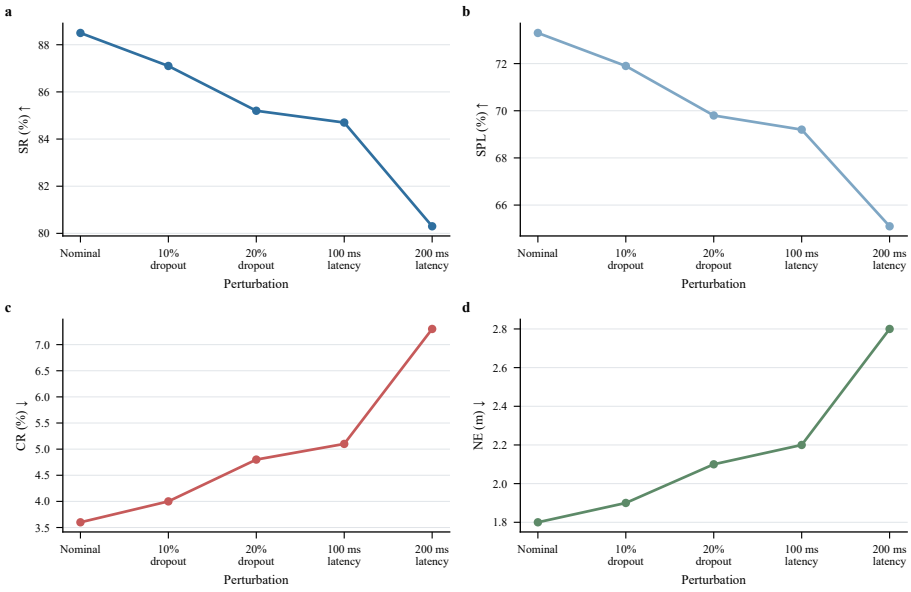


Figure 10: Robustness curves under sensing and timing perturbations. Performance is evaluated as depth dropout or sensing-to-command latency increases; the safety-aware planner degrades gradually until the perturbation exceeds the nominal onboard timing margin.

C ROS AND PX4 INTERFACE

The reference package keeps localization and low-level control outside the planner. The planner subscribes to `/fastlio/odometry` (`nav_msgs/Odometry`), `/camera/color/image_raw`, `/camera/depth/image_rect_raw`, `/lidar/points`, `/aerialgenesis/instruction` (`std_msgs/String`), and `/aerialgenesis/emergency_stop` (`std_msgs/Bool`). It publishes `/aerialgenesis/position_cmd` (`geometry_msgs/PoseStamped`) for the PX4 tracking module and `/aerialgenesis/candidate_path` (`nav_msgs/Path`) for monitoring. Topic names are parameters in `Code/config/planner.yaml`; no Fast-LIO2 or PX4 implementation is duplicated in the planner package. A stale odometry or depth timestamp causes the node to stop publishing motion commands and assert the emergency-stop topic.

D ADDITIONAL SAFETY AND REPRODUCIBILITY CHECKS

Before a flight, we verify frame conventions (ENU versus body frame), camera–LiDAR extrinsics, timestamp synchronization, command bounds, and the PX4 failsafe mode. During evaluation, we log raw observations, odometry, all sampled trajectories, selected setpoints, safety scores, controller status, and emergency-stop causes. The reported collision rate includes contact, geofence violation, emergency stop, and human intervention. These checks reproduce the safety bound empirically and distinguish planner failures from localization or low-level tracking failures.