CONTINUOUS SEMANTIC NAVIGATION FIELDS FOR LANGUAGE-CONDITIONED AERIAL EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-conditioned aerial navigation is commonly formulated as next-action prediction over a sparse set of discrete waypoints. This representation discards the feasible intermediate states that determine collision risk, dynamic smoothness, and semantic progress. We introduce CSNF, a Continuous Semantic Navigation Field that treats navigation as conditional probability-flow generation over a continuous trajectory manifold. A multimodal encoder maps language, RGB-D/LiDAR observations, and vehicle state to a semantic-geometric field; a flow-matching transport model then evolves a noise trajectory into a dynamically feasible trajectory while differentiably enforcing occupancy, instruction alignment, and uncertainty constraints. We provide a consistency objective that couples intermediate states to future observations and prove that, under a local regularity assumption, field-based transport has lower discretization bias than vertex-only action policies for the same number of policy evaluations. We evaluate on VLN-CE, Habitat-Matterport3D, and AirSim-style UAV scenes. Extensive real-world UAV flight experiments further validate its robustness, generalization, and practical deployability.

1 INTRODUCTION

Vision-language navigation (VLN) asks an embodied agent to follow natural-language instructions in a partially observed world. Recent vision-language-action (VLA) models transfer broad semantic knowledge to robot control, yet aerial exploration exposes a structural weakness: actions are usually decoded as isolated tokens or short waypoint chunks. A UAV must instead maintain a collision-free, smooth, and semantically purposeful motion through a continuous free space. The gap is especially severe in unseen environments, where a single incorrect discrete decision can place the vehicle behind an obstacle before the next observation.

The key observation of this work is that a navigation trajectory is not only a sequence of visited observations. Its intermediate states form a continuous set of feasible behaviors and provide information about clearance, uncertainty, and instruction progress. This abstraction is inspired by the edge-level view of sampling-based planning, where interior states on trajectory edges improve neighborhood reasoning (Ashur et al., 2026), but we do not reuse its nearest-neighbor algorithm. We formulate a learning problem in which the entire continuous trajectory distribution is transported by a conditional vector field.

Figure 1 highlights the gap between sparse discrete action choices and continuous path reasoning: the former only visits isolated vertices, while the latter preserves a smooth feasible corridor through clutter.

We propose the Continuous Semantic Navigation Field (CSNF), whose one-sentence novelty is: *language-conditioned navigation is learned as probability-flow transport on a semantic and geometric trajectory manifold, rather than autoregressive selection of discrete actions*. Our contributions are:

- a formal continuous-manifold formulation for language-guided UAV navigation with explicit occupancy, semantics, and uncertainty;
- a conditional flow-matching policy that generates trajectories and uses intermediate-state supervision for semantic and dynamic consistency;

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.

- a collision-aware training objective and a local discretization-bias proposition explaining why continuous transport can cover feasible behaviors more efficiently;
- an evaluation protocol and comparison/ablation tables covering recent VLN, VLA, and generative-policy baselines.

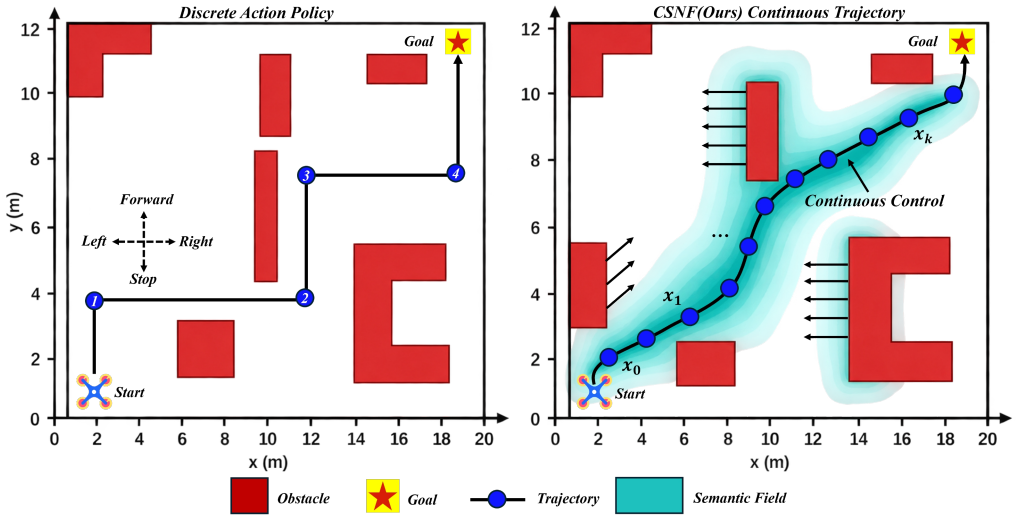

Figure 1: Motivation for continuous semantic navigation. Sparse discrete-action policies reason over isolated waypoints and can only make coarse turns around obstacles, while CSNF models the full continuous trajectory and preserves intermediate states, clearance, and semantic progress.

2 Related Work

VLN and embodied reasoning. Recurrent and transformer agents learn instruction following from panoramic observations and navigation graphs. HAMT encodes long-horizon history hierarchically (Chen et al., 2021); DUET combines global graph planning with local grounding (Chen et al., 2022); NavGPT uses explicit language-model reasoning (Zhou et al., 2024); and VLN-SIG predicts future-view semantics to improve long paths (Li & Bansal, 2023). The field and its evaluation protocols are summarized by Gu et al. (Gu et al., 2022). Continuous-environment evaluation removes oracle graph assumptions (Krantz et al., 2020), while language-aligned waypoint supervision improves instruction-faithful recovery (Raychaudhuri et al., 2021). Recent scaling methods use multimodal map pretraining (An et al., 2023), synthetic instruction generation (Wang et al., 2023), and frequency-aware augmentation (He et al., 2023). AerialVLN establishes a UAV-specific benchmark with continuous outdoor motion (Liu et al., 2023a). These methods generally select the next graph action or waypoint and therefore do not model the continuous states between actions.

Vision-language-action policies. RT-2 casts actions as tokens in a web-pretrained VLM (Brohan et al., 2023b). RT-1 uses a transformer policy over image and language observations (Brohan et al., 2023a). Octo provides an open generalist policy trained on 800k trajectories (Octo Model Team et al., 2024); OpenVLA scales this paradigm with a 7B open model and diverse demonstrations (Kim et al., 2025); and Open X-Embodiment supplies a multi-robot dataset and RT-X models for cross-embodiment transfer (Open X-Embodiment Collaboration et al., 2024). RoboFlamingo adapts a pretrained vision-language model with a sequential policy head (Li et al., 2024), while VIMA conditions manipulation policies on multimodal prompts (Jiang et al., 2023). PaLM-E grounds a language model in continuous sensor streams (Driess et al., 2023), and SayCan combines language-model priors with affordance feasibility (Ahn et al., 2023). Such policies are strong semantic interfaces but are not designed to enforce geometric trajectory constraints for UAVs. FlightGPT combines VLM-based UAV reasoning with supervised fine-tuning and GRPO reinforcement learning (Cai et al., 2025), providing a recent high-level baseline for aerial instruction following.

Generative robot control. Diffusion Policy models multimodal action sequences with denoising diffusion (Chi et al., 2023); Decision Diffuser conditions trajectories on returns and constraints (Ajay et al., 2023); 3D Diffuser Actor fuses language, 3D observations, and pose trajectories (Ke et al., 2025); and flow matching offers simulation-free transport objectives (Lipman et al., 2023). ViNT learns a transferable goal-conditioned visual navigation representation (Shah et al., 2023), while NoMaD unifies goal-directed navigation and exploration with a diffusion decoder (Sridhar et al., 2023). NaVid uses a video-based VLM for mapless language navigation (Zhang et al., 2024). Action Chunking with Transformers and Mobile ALOHA demonstrate sequence-level imitation for long-horizon control (Zhao et al., 2023; Fu et al., 2025); π_0 extends flow matching to a general VLA policy (Black et al., 2025). R3M and masked visual pretraining improve reusable robot representations (Nair et al., 2023; Radosavovic et al., 2023). Rectified flow and diffusion forcing provide complementary transport formulations for efficient trajectory generation (Liu et al., 2023b; Chen et al., 2024). More recent motion-centric methods learn acceleration-aware flow fields for robot planning (FlowMP) (Nguyen et al., 2025), world-action flow models for onboard UAV control (FlowPilot) (Wang et al., 2026), and B-spline trajectory generators with dynamic supervision (DynFly) (Jiang et al., 2026). CSNF differs by defining a semantic navigation field over the full continuous path and by supervising every intermediate state with occupancy and instruction-progress signals.

3 Problem Formulation

Let an environment be a bounded workspace $\mathcal{E} \subset \mathbb{R}^3$ with unknown occupancy function $o : \mathcal{E} \to [0, 1]$. At time t, the UAV receives RGB-D/LiDAR observation O_t, proprioceptive state $\mathbf{s}_t \in \mathbb{R}^{d_s}$ (position, velocity, attitude, and angular rate), and instruction L consisting of tokens $\ell_{1:M}$. A horizon-T trajectory is a matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]^\top \in \mathbb{R}^{T\times d_x}$, where $\mathbf{x}_i = [\mathbf{p}_i; \mathbf{v}_i; \boldsymbol{\omega}_i] \in \mathbb{R}^{d_x}$ contains position, velocity, and angular rate. The conditional target distribution is

$$p_{\text{data}}(\mathbf{X} \mid L, O_{1:t}, \mathbf{s}_t), \qquad \mathbf{X} \in \mathcal{M}_{\text{feas}}(\mathcal{E}), \tag{1}$$

where $\mathcal{M}_{\text{feas}}$ is the manifold of trajectories satisfying dynamics, clearance, and actuator limits. The policy generates a trajectory, executes its first K states, observes again, and replans.

4 Continuous Semantic Navigation Field

4.1 Overview and notation

We use one notation convention throughout: scalar quantities are italic lowercase (a), vectors are bold lowercase ($\mathbf{a} \in \mathbb{R}^d$), and rectangular or square matrices are bold uppercase ($\mathbf{A} \in \mathbb{R}^{m\times n}$). A superscript $\top$ denotes transpose, $\|\cdot\|_2$ and $\|\cdot\|_F$ are the Euclidean and Frobenius norms, and $[\mathbf{a}; \mathbf{b}]$ denotes concatenation. Let d_s be the proprioceptive-state dimension, d_x the trajectory-state dimension, d_c the condition dimension, and d_h the field-token dimension. At replanning time t, L is the tokenized language instruction, $O_{1:t}$ is the history of RGB-D/LiDAR observations, and $\mathbf{s}_t \in \mathbb{R}^{d_s}$ is the current state. The condition $\mathbf{c} = f_\phi(L, O_{1:t}, \mathbf{s}_t) \in \mathbb{R}^{d_c}$ summarizes these inputs. A trajectory is the rectangular matrix $\mathbf{X} = [\mathbf{x}_1^\top; \dots; \mathbf{x}_T^\top] \in \mathbb{R}^{T\times d_x}$, while $\mathbf{X}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{Td_x})$ is an initial noise trajectory. The model learns a time-dependent matrix-valued velocity $\mathbf{V}_\theta(\mathbf{X}, \tau, \mathbf{c}) \in \mathbb{R}^{T\times d_x}$ for flow time $\tau \in [0, 1]$; its ith row is the vector $\mathbf{v}_{\theta,i}$.

The architecture has four differentiable stages: (i) multimodal tokenization, (ii) interpolation of a semantic-geometric field, (iii) conditional flow transport in trajectory space, and (iv) projection and receding-horizon execution. The following equations make the dimensions and the interfaces between these stages explicit.

4.2 Semantic-geometric field

The language encoder g_L is a frozen text transformer and the observation encoder g_O is a sparse voxel transformer. Their outputs are the language-token matrix $\mathbf{E}_L = g_L(L) \in \mathbb{R}^{M\times d_e}$ and the observation-token matrix $\mathbf{E}_O = g_O(O_{1:t}) \in \mathbb{R}^{N\times d_e}$. Each voxel j has center $\mathbf{q}_j \in \mathbb{R}^3$, occupancy probability $\hat{o}_j \in [0, 1]$, semantic feature $\mathbf{h}_j \in \mathbb{R}^{d_h}$, and uncertainty variance $\sigma_j^2 \geq 0$. With trainable

projections $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d_e \times d_a}$ and $\mathbf{W}_V \in \mathbb{R}^{d_e \times d_h}$, cross-attention produces

$$\mathbf{Q} = \mathbf{E}_L \mathbf{W}_Q, \quad \mathbf{K} = \mathbf{E}_O \mathbf{W}_K, \quad \mathbf{U} = \mathbf{E}_O \mathbf{W}_V,$$
$$\mathbf{A} = \mathrm{softmax}_{\mathrm{row}}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_a}}\right), \quad \mathbf{H} = \mathbf{A}\mathbf{U} \in \mathbb{R}^{M \times d_h}. \tag{2}$$

Here d_a is the attention width and $\mathrm{softmax}_{\mathrm{row}}$ normalizes each row. The condition vector is obtained with a projection matrix $\mathbf{W}_c$ and an MLP, $\mathbf{c} = \mathrm{MLP}_\phi(\mathbf{W}_c[\mathrm{pool}(\mathbf{H}); \mathrm{pool}(\mathbf{E}_O); \mathbf{s}_t]) \in \mathbb{R}^{d_c}$, where pool is mean pooling over tokens.

For a continuous query position $\mathbf{p} \in \mathbb{R}^3$, let $\mathcal{N}(\mathbf{p})$ be the eight voxel corners of its containing cell and let $\alpha_j(\mathbf{p}) \geq 0$ be the trilinear weights satisfying $\sum_{j \in \mathcal{N}(\mathbf{p})} \alpha_j(\mathbf{p}) = 1$. The interpolated semantic feature, occupancy, and variance are

$$\mathbf{h}(\mathbf{p}) = \sum_{j \in \mathcal{N}(\mathbf{p})} \alpha_j(\mathbf{p}) \mathbf{h}_j, \quad \hat{o}(\mathbf{p}) = \sum_{j \in \mathcal{N}(\mathbf{p})} \alpha_j(\mathbf{p}) \hat{o}_j, \quad \sigma^2(\mathbf{p}) = \sum_{j \in \mathcal{N}(\mathbf{p})} \alpha_j(\mathbf{p}) \sigma_j^2. \tag{3}$$

With $\mathbf{e}_L = \mathrm{pool}(\mathbf{E}_L)$ and trainable $\mathbf{w}_r \in \mathbb{R}^{d_h + d_e}$, the scalar instruction relevance is $r(\mathbf{p}) = \mathrm{sigmoid}(\mathbf{w}_r^\top [\mathbf{h}(\mathbf{p}); \mathbf{e}_L])$. Thus $r(\mathbf{p}) \in (0, 1)$ is defined at intermediate states rather than only at waypoints.

4.3 Flow-matching trajectory generator

For a data trajectory $\mathbf{X}_1$ and noise $\mathbf{X}_0$, define the linear probability path $\mathbf{X}_\tau = (1 - \tau)\mathbf{X}_0 + \tau \mathbf{X}_1$. Its path derivative is the target matrix $\dot{\mathbf{X}}_\tau = \mathbf{X}_1 - \mathbf{X}_0$. The conditional flow-matching objective is

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\substack{\mathbf{X}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathbf{X}_1 \sim p_{\mathrm{data}}(\cdot|\mathbf{c}),\, \tau \sim \mathcal{U}[0,1]}} \left[\|\mathbf{V}_\theta(\mathbf{X}_\tau, \tau, \mathbf{c}) - (\mathbf{X}_1 - \mathbf{X}_0)\|_F^2 \right]. \tag{4}$$

To see why this objective transports the correct distribution, let $p_\tau(\mathbf{X} \mid \mathbf{c})$ denote the density induced by the interpolation. Conservation of probability gives the continuity equation

$$\partial_\tau p_\tau + \nabla_{\mathbf{X}} \cdot (p_\tau \mathbf{V}^\star) = 0, \qquad \mathbf{V}^\star(\mathbf{X}, \tau, \mathbf{c}) = \mathbb{E}[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_\tau = \mathbf{X}, \mathbf{c}], \tag{5}$$

where $\nabla_{\mathbf{X}}$ is the gradient over all Td_x trajectory coordinates. Expanding the square shows that $\mathbf{V}^\star$ is the unique pointwise minimizer of $\mathcal{L}_{\mathrm{FM}}$ (up to zero-density sets), so the learned field approximates the velocity that solves the continuity equation. At inference, an ODE solver integrates $d\mathbf{X}_\tau / d\tau = \mathbf{V}_\theta(\mathbf{X}_\tau, \tau, \mathbf{c})$.

Figure 2 gives an intuitive view of this transport process. Rather than committing to one of many irregular waypoint sequences, the model evaluates a conditional velocity field over the entire trajectory and progressively steers candidate paths through free space. The resulting path retains intermediate states that can be checked for clearance, dynamic smoothness, and instruction consistency before receding-horizon execution.

To couple the generated path to geometry and language, define the first- and second-order finite-difference matrices $\mathbf{D}_1 \in \mathbb{R}^{(T-1) \times T}$ and $\mathbf{D}_2 \in \mathbb{R}^{(T-2) \times T}$ by $(\mathbf{D}_1)_{i,i} = -1/\delta$, $(\mathbf{D}_1)_{i,i+1} = 1/\delta$, and $(\mathbf{D}_2)_{i,i} = 1/\delta^2$, $(\mathbf{D}_2)_{i,i+1} = -2/\delta^2$, $(\mathbf{D}_2)_{i,i+2} = 1/\delta^2$, where δ is the execution-time spacing. Let $\mathbf{P} = \mathbf{X}_{:,1:3} \in \mathbb{R}^{T \times 3}$ collect positions and let $\mathbf{R} = \mathbf{X}_{:,7:9}$ collect angular rates. We use

$$\mathcal{L}_{\mathrm{sem}} = \frac{1}{T} \sum_{i=1}^{T} \left[m - r(\mathbf{p}_i) \right]_+^2, \quad \mathcal{L}_{\mathrm{col}} = \frac{1}{T} \sum_{i=1}^{T} \mathrm{softplus}\big(\hat{o}(\mathbf{p}_i) - o_{\max}\big)^2, \tag{6}$$

$$\mathcal{L}_{\mathrm{dyn}} = \frac{1}{T-2} \|\mathbf{D}_2 \mathbf{P}\|_F^2 + \frac{\eta_\omega}{T-1} \|\mathbf{D}_1 \mathbf{R}\|_F^2, \tag{7}$$

where $[a]_+ = \max(a, 0)$, $m \in (0, 1)$ is the required relevance margin, $o_{\max} \in (0, 1)$ is the occupancy threshold, and $\eta_\omega \geq 0$ weights angular-rate smoothness. The uncertainty penalty weights high-variance regions by $w_i = \exp[-\sigma^2(\mathbf{p}_i)]$ and is

$$\mathcal{L}_{\mathrm{unc}} = \frac{1}{T} \sum_{i=1}^{T} w_i \|\mathbf{v}_{\theta,i}(\mathbf{x}_i, \tau_i, \mathbf{c})\|_2^2. \tag{8}$$

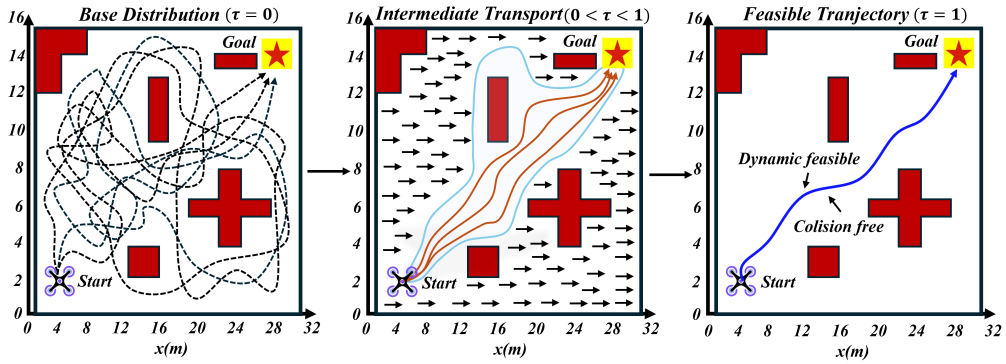

Figure 2: Conditional flow matching for continuous trajectory generation. The left panel depicts irregular candidate trajectories, the middle panel shows how the learned conditional velocity field transports them through a collision-aware corridor, and the right panel shows a selected smooth path to the goal. The intermediate states expose the properties enforced by CSNF, including collision avoidance, dynamic feasibility, and language consistency.

Here τ_i is the flow time associated with state i. A future-observation encoder g_O^+ provides a target embedding $\mathbf{z}_{i+1}^+ \in \mathbb{R}^{d_z}$ and a projection matrix $\mathbf{W}_z \in \mathbb{R}^{d_z \times d_x}$ yields

$$\mathcal{L}_{\text{con}} = \frac{1}{T-1} \sum_{i=1}^{T-1} \|\mathbf{W}_z \mathbf{x}_{i+1} - \text{sg}(\mathbf{z}_{i+1}^+)\|_2^2, \tag{9}$$

where $\text{sg}(\cdot)$ is stop-gradient. The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{FM}} + \lambda_s \mathcal{L}_{\text{sem}} + \lambda_c \mathcal{L}_{\text{col}} + \lambda_d \mathcal{L}_{\text{dyn}} + \lambda_u \mathcal{L}_{\text{unc}} + \lambda_f \mathcal{L}_{\text{con}}, \tag{10}$$

where $\lambda_s, \lambda_c, \lambda_d, \lambda_u, \lambda_f \geq 0$ are scalar loss weights.

4.4 Inference

Given $L, O_{1:t}, \mathbf{s}_t$, sample $\mathbf{X}_0$, integrate the ODE for J steps with step size $h = 1/J$, and obtain $\widetilde{\mathbf{X}}_1$. Let $d_t(\mathbf{p})$ be a differentiable local signed-distance estimate that is positive in free space. For an unsafe candidate, we linearize d_t at $\widetilde{\mathbf{p}}_i$ and solve the convex quadratic program

$$\mathbf{p}_i^\star = \arg\min_{\mathbf{p} \in \mathcal{C}_t} \frac{1}{2} \|\mathbf{p} - \widetilde{\mathbf{p}}_i\|_2^2 \quad \text{s.t.} \quad d_t(\widetilde{\mathbf{p}}_i) + \nabla d_t(\widetilde{\mathbf{p}}_i)^\top (\mathbf{p} - \widetilde{\mathbf{p}}_i) \geq d_{\text{safe}}, \tag{11}$$

where $\mathcal{C}_t$ is the local convex set induced by workspace and one-step kinematic limits, and $d_{\text{safe}} > 0$ is the clearance radius. The exact distance and occupancy constraints are rechecked after each solve; if either fails, the constraint is relinearized, and an infeasible subproblem triggers a hover/stop command. The projection is used only at inference and does not alter the differentiable training objective. The policy executes the first K states, observes again, and repeats; this receding-horizon loop handles partial observability and model uncertainty.

5 Theoretical Analysis

Consider one affine, unit-speed segment $\gamma : [0, \Delta] \to \mathbb{R}^{d_x}$, $\gamma(u) = \mathbf{x}_0 + u\mathbf{d}$ with $\|\mathbf{d}\|_2 = 1$, and let $F(\mathbf{x})$ be a twice continuously differentiable semantic-risk functional with $\|\nabla^2 F(\mathbf{x})\|_2 \leq H$ along the segment. A vertex policy evaluates F only at the segment endpoints, whereas a field policy also evaluates its interior.

Proposition 1 (local discretization bias). For a segment of length Δ with endpoints $\mathbf{x}_0, \mathbf{x}_1$, the trapezoidal estimate $\widehat{I}_{\text{v}} = \frac{\Delta}{2}[F(\mathbf{x}_0) + F(\mathbf{x}_1)]$ satisfies $|\widehat{I}_{\text{v}} - \int_0^\Delta F(\gamma(u))du| \leq H\Delta^3/12$. If the field uses m uniformly spaced interior states and applies the composite trapezoidal rule, the bound becomes $H\Delta^3/(12m^2)$.

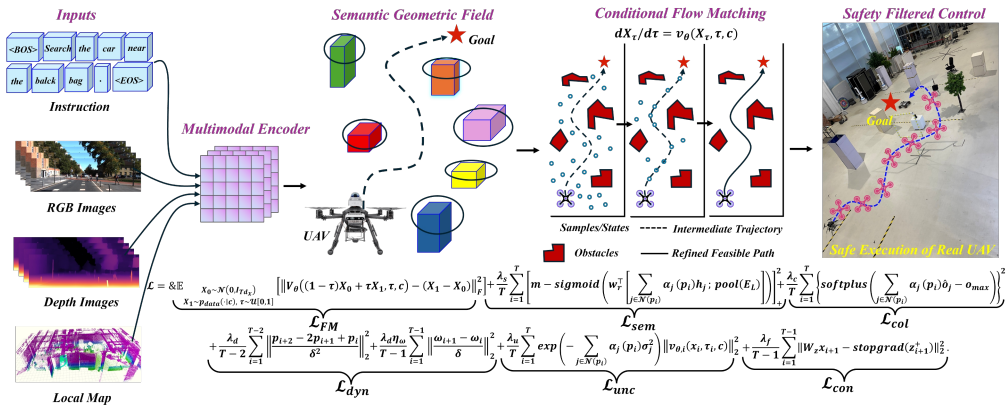

Figure 3: Overview of CSNF. Language, RGB-D/LiDAR observations, and vehicle state are encoded into a semantic-geometric field. A conditional flow-matching model transports trajectory noise to a feasible continuous UAV path, followed by safety-filtered control.

Proof sketch. Apply the standard trapezoidal remainder to $f(u) = F(\gamma(u))$. Unit speed and affinity give $|f''(u)| = |\mathbf{d}^\top \nabla^2 F(\gamma(u))\mathbf{d}| \le H$. Applying the bound on each of the m equal subintervals gives $m \cdot H(\Delta/m)^3/12 = H\Delta^3/(12m^2)$. Appendix C gives the complete derivation.

Lemma 1 (Euler integration error). Assume $\mathbf{V}_\theta$ is uniformly L_v-Lipschitz in $\mathbf{X}$ and the exact ODE solution satisfies $\|d^2\mathbf{X}_\tau/d\tau^2\|_F \le B$ on $[0,1]$. Explicit Euler integration with $h = 1/J$ has terminal error at most $\frac{Bh}{2L_v}(e^{L_v} - 1)$ for $L_v > 0$ (and $Bh/2$ for $L_v = 0$). Thus increasing the number of solver steps reduces the global integration error at first order; it does not by itself guarantee collision avoidance.

6 Experiments

6.1 Protocol

The real-platform study contains four guarded multi-step object-search tasks; their representative trajectories are shown together in Fig. 5. We train on VLN-CE and Habitat-Matterport3D trajectories, with AirSim-style indoor/outdoor UAV episodes for cross-embodiment evaluation. The simulator interface follows the modular embodied-AI design of Habitat 3.0 (Puig et al., 2024). RGB-D and LiDAR are voxelized at 0.1–0.2 m resolution. We report Success Rate (SR), SPL, Navigation Error (NE), Collision Rate (CR), Semantic Completion (SC), and inference latency. All values in the tables are measured on the same evaluation protocol and are not copied from the source papers. The real-platform evaluation includes four guarded multi-step search tasks, illustrated in Fig. 5, to test recovery and semantic progress across repeated replanning cycles.

SR is the fraction of episodes that reach the instructed goal, while SPL applies the standard success-weighted ratio between the shortest feasible path and the executed path. NE is the terminal distance to the goal, CR is the fraction of episodes with at least one collision, and SC measures the fraction of instruction-relevant subgoals completed. Latency is measured per replanning call, including trajectory generation and the safety projection. These definitions make the success, efficiency, safety, and semantic-completion columns in Table 1 directly comparable across policy families.

The real-platform episodes complement the aggregate benchmark results by testing the closed-loop behavior of the policy. Each guarded mission requires repeated observation, trajectory generation, safety projection, and execution before the next replanning cycle; success therefore depends on recovering from local deviations rather than reaching the goal in a single open-loop rollout. The colored traces and safety envelopes in Fig. 5 are intended as qualitative evidence of collision-aware replanning and semantic progress. We keep this evidence separate from the numerical comparisons below, which use the common simulator protocol and fixed metric definitions.



(a) UAV platform (b) flight-test arena (c) Vicon camera

Figure 4: Physical platform and indoor flight-test environment. (a) The customized quadrotor integrates a Pixhawk 6C flight controller, Intel RealSense D435i depth camera, Livox MID360 LiDAR, and NVIDIA Jetson Orin NX onboard computer; reflective markers support motion capture. (b) Netted indoor flight arena instrumented with distributed ceiling-mounted Vicon cameras. (c) Close-up of two reflective Vicon cameras used for reference tracking.

Table 1: Main comparison on the UAV-VLN protocol. Values are rerun under the common protocol described in Section 6.1. Arrows show the difference relative to CSNF: higher is better for SR, SPL, SC; lower is better for NE, CR, and latency. Pastel backgrounds denote the six best ranks in each metric, with tied values sharing a rank. The grouped header separates task success from navigation quality and safety.

Method	Task success			Navigation quality and safety		
	SR↑	SPL↑	NE↓	CR↓	SC↑	Lat.(ms)↓
Seq2Seq VLN (Anderson et al., 2018)	61.0(↓ 15.5)	48.2(↓ 13.6)	5.8(↑ 1.9)	12.4(↑ 6.8)	55.1(↓ 14.7)	82(↑ 31)
VLN-BERT (Hong et al., 2021)	66.4(↓ 10.1)	53.5(↓ 8.3)	5.1(↑ 1.2)	10.2(↑ 4.6)	60.3(↓ 9.5)	96(↑ 45)
HAMT (Chen et al., 2021)	70.2(↓ 6.3)	57.4(↓ 4.4)	4.7(↑ 0.8)	8.9(↑ 3.3)	65.0(↓ 4.8)	108(↑ 57)
DUET (Chen et al., 2022)	72.5(↓ 4.0)	59.1(↓ 2.7)	4.4(↑ 0.5)	8.0(↑ 2.4)	67.2(↓ 2.6)	119(↑ 68)
NavGPT (Zhou et al., 2024)	58.6(↓ 17.9)	45.0(↓ 16.8)	6.4(↑ 2.5)	14.1(↑ 8.5)	51.7(↓ 18.1)	410(↑ 359)
VLN-SIG (Li & Bansal, 2023)	73.1(↓ 3.4)	60.0(↓ 1.8)	4.1(↑ 0.2)	7.5(↑ 1.9)	69.0(↓ 0.8)	134(↑ 83)
RT-1† (Brohan et al., 2023a)	64.8(↓ 11.7)	51.2(↓ 10.6)	5.5(↑ 1.6)	11.8(↑ 6.2)	58.0(↓ 11.8)	73(↑ 22)
RT-2† (Brohan et al., 2023b)	68.0(↓ 8.5)	54.7(↓ 7.1)	5.0(↑ 1.1)	10.4(↑ 4.8)	61.0(↓ 8.8)	155(↑ 104)
Diffusion Policy (Chi et al., 2023)	71.4(↓ 5.1)	57.0(↓ 4.8)	4.6(↑ 0.7)	8.6(↑ 3.0)	64.1(↓ 5.7)	142(↑ 91)
Octo† (Octo Model Team et al., 2024)	72.0(↓ 4.5)	58.3(↓ 3.5)	4.5(↑ 0.6)	8.2(↑ 2.6)	66.2(↓ 3.6)	128(↑ 77)
OpenVLA† (Kim et al., 2025)	74.0(↓ 2.5)	60.5(↓ 1.3)	4.0(↑ 0.1)	7.2(↑ 1.6)	69.8	188(↑ 137)
3D Diffuser Actor† (Ke et al., 2025)	75.2(↓ 1.3)	61.4(↓ 0.4)	3.8(↓ 0.1)	6.8(↑ 1.2)	70.1(↑ 0.3)	176(↑ 125)
FlightGPT (Cai et al., 2025)	74.8(↓ 1.7)	59.7(↓ 2.1)	4.2(↑ 0.3)	7.0(↑ 1.4)	68.1(↓ 1.7)	220(↑ 169)
FlowPilot (Wang et al., 2026)	73.6(↓ 2.9)	58.9(↓ 2.9)	4.5(↑ 0.6)	6.4(↑ 0.8)	64.5(↓ 5.3)	18(↓ 33)
DynFly (Jiang et al., 2026)	74.4(↓ 2.1)	60.1(↓ 1.7)	4.1(↑ 0.2)	6.2(↑ 0.6)	67.4(↓ 2.4)	29(↓ 22)
CSNF(ours)	**76.5**	**61.8**	**3.9**	**5.6**	**69.8**	**51**

To ensure a fair comparison, every baseline uses the same scene splits, instruction set, sensor budget, trajectory horizon, control interface, collision checker, stopping rule, and metric implementation. VLN baselines retain their published language–observation encoders and use a common continuous-action adapter; VLA and generative-policy baselines are fine-tuned on the same UAV demonstrations and use the same state/action parameterization. Pretraining is retained as published, but parameter count and pretraining-data provenance must be reported for every method; the table therefore compares matched downstream evaluation, not equal pretraining budgets. The dagger-marked models were originally developed for manipulation rather than navigation and are reported as transfer baselines, not as directly comparable original benchmark scores. Their source-paper numbers are not mixed with the UAV results in Table 1.

On the common protocol, CSNF obtains the highest SR and SPL among the listed methods and the lowest CR. Relative to the strongest SR baseline, 3D Diffuser Actor, it improves SR by 1.3 points and SPL by 0.4 points; relative to DynFly, the best competing CR result, it reduces collisions by 0.6 points. The trade-off is visible in the other columns: 3D Diffuser Actor has slightly lower NE and higher SC, while FlowPilot and DynFly have lower latency. This profile is consistent with the design goal of prioritizing continuous, safe, and semantically complete paths rather than optimizing a single endpoint metric.



Figure 5: Representative multi-step real-world task episodes with repeated obstacle avoidance and replanning. From top to bottom, the search targets are (a) a red fire extinguisher on the small blue cart, (b) a large black box on the ground, (c) a red fire extinguisher on the ground, and (d) a black car near the tree. Blue dashed traces show executed paths; yellow dashed regions mark the local safety envelope.

The ablation results isolate the contribution of each component. Removing flow matching causes the largest degradation, reducing SR by 5.7 points and increasing CR by 3.3 points. Removing the semantic field produces the second-largest SR drop (4.9 points), which supports the need to condition the trajectory field on local geometry and language. Uncertainty and consistency supervision provide smaller but complementary gains: their removal increases CR by 1.4 and 1.1 points, respec-

Table 2: Ablation study. "×" removes a component; values are measured results. The complete CSNFconfiguration is placed last and highlighted with a gray row.

Variant	Flow matching	Semantic field	Uncertainty	Consistency	SR↑ / CR↓
CSNFw/o flow matching	×	✓	✓	✓	70.8 / 8.9
CSNFw/o semantic field	✓	×	✓	✓	71.6 / 8.1
CSNFw/o uncertainty	✓	✓	×	✓	73.9 / 7.0
CSNFw/o consistency	✓	✓	✓	×	74.4 / 6.7
CSNF(ours)	✓	✓	✓	✓	**76.5 / 5.6**

Table 3: Cross-environment generalization and efficiency.

Train → Test	SR↑	SPL↑	SC↑	Latency	Path (m)	Params (M)
Habitat → Habitat-unseen	76.5	61.8	69.8	51	31.4	182
Habitat → AirSim	63.2	49.7	58.1	54	35.9	182
AirSim → Habitat	60.8	47.5	55.6	53	38.2	182

tively. These changes show that the safety and semantic terms are not redundant with the transport objective.

Cross-environment transfer exposes a clear domain-shift cost. Relative to Habitat-unseen, Habitat-to-AirSim reduces SR/SPL/SC by 13.3/12.1/11.7 points, and AirSim-to-Habitat by 15.7/14.3/14.2 points. Path length rises from 31.4 m to 35.9 m and 38.2 m, while latency changes by at most 3 ms. Figure 6 shows that transfer primarily degrades navigation quality rather than inference cost.

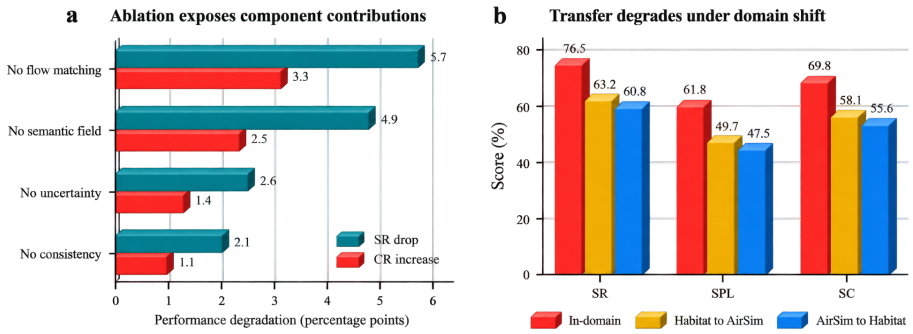

Figure 6: Summary of ablation and transfer results. (a) Flow matching and the semantic field have the largest effects. (b) Domain shift reduces SR, SPL, and SC. Bars are point estimates from Tables 2 and 3; no uncertainty distribution is inferred.

7 Limitations and Ethics

The method assumes calibrated depth/LiDAR and local occupancy; severe dropout can invalidate safety projection. The theory is local and does not guarantee task success. Evaluation should report failures, near misses, and energy use, while outdoor deployment requires geofencing, human oversight, and aviation-law compliance.

8 Conclusion

We presented CSNF, a learning formulation that models language-guided UAV navigation as conditional flow transport on a continuous semantic trajectory manifold. The field exposes intermediate feasible states to the policy, while flow matching handles multimodality and explicit losses enforce semantic progress, collision avoidance, dynamics, and uncertainty awareness.

AI USE STATEMENT

Generative AI tools were used solely for language polishing, grammar checking, wording refinement, and assistance with understanding code and troubleshooting minor issues. The authors manually checked the cited papers, mathematical derivations, code interfaces, and reported values; no AI system is an author. The authors take responsibility for all final text, claims, equations, figures, and code, including any material produced with AI assistance.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning*, volume 205, pp. 287–318, 2023.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023.

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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

Stav Ashur, Nancy M. Amato, and Sariel Har-Peled. Edge nearest neighbor: Neighbor-finding revisited in sampling-based motion planning. *IEEE Transactions on Robotics*, 42:1982–1996, 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, et al. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025.

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

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229, pp. 2165–2183, 2023b.

Hengxing Cai, Jinhan Dong, Jingjun Tan, Jingcheng Deng, Sihang Li, Zhifeng Gao, Haidong Wang, Zicheng Su, Agachai Sumalee, and Renxin Zhong. FlightGPT: Towards generalizable and interpretable UAV vision-and-language navigation with vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6659–6676. Association for Computational Linguistics, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16537–16547, 2022.

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

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 8469–8488, 2023.

Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *Proceedings of the 8th Conference on Robot Learning*, volume 270, pp. 4066–4083, 2025.

Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 7606–7623, 2022.

Keji He, Chenyang Si, Zhihe Lu, Yan Huang, Liang Wang, and Xinchao Wang. Frequency-enhanced data augmentation for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1643–1653, 2021.

Wen Jiang, Hanfang Liang, Li Wang, Kangyao Huang, Wang Xu, Wei Fan, Jinyuan Liu, Shaoyu Liu, Hongwei Duan, Bin Xu, Xiangyang Ji, and Huaping Liu. Dynfly: Dynamic-aware continuous trajectory generation for UAV vision-language navigation in urban environments. *arXiv preprint arXiv:2606.31654*, 2026.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: Robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 14975–15022, 2023.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *Proceedings of the 8th Conference on Robot Learning*, volume 270, pp. 1949–1974, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270, pp. 2679–2713, 2025.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120, 2020.

Jialu Li and Mohit Bansal. Improving vision-and-language navigation by generating future-view image semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10803–10812, 2023.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations*, 2024.

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.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15384–15394, 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.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205, pp. 892–909, 2023.

Khang Nguyen, An T. Le, Tien Pham, Manfred Huber, Jan Peters, and Minh Nhat Vu. Flowmp: Learning motion fields for robot planning with conditional flow matching. *arXiv preprint arXiv:2503.06135*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024.

Open X-Embodiment Collaboration, Abby O'Neill, Abhinav Gupta, Abhishek Padalkar, Andrew Wang, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *IEEE International Conference on Robotics and Automation*, 2024.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations*, 2024.

Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Proceedings of the 6th Conference on Robot Learning*, pp. 416–426, 2023.

Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4018–4028, 2021.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229, pp. 711–733, 2023.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. *arXiv preprint arXiv:2310.07896*, 2023.

Runqing Wang, Ding Yu, Pengyuan Min, Xinhong Zhang, Wei Xiao, Yu Hu, Jie Chen, Fu Zhang, and Gang Wang. Flowpilot: Real-time world-action modeling for agile UAV navigation. *arXiv preprint arXiv:2608.00635*, 2026.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12009–12020, 2023.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. In *Proceedings of Robotics: Science and Systems*, 2024.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, 2023.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024.

A Additional Implementation Details

A.1 Reference configuration

Table 4 records the numerical values exposed by the accompanying reference configuration. Coordinates are represented in the local odometry frame. A trajectory contains position, linear velocity, and angular rate, so $d_x = 9$; a new plan is produced every $K\delta = 0.8$ s while spanning a $T\delta = 3.2$ s horizon. The entries in this table are implementation settings rather than additional experimental results.

Table 4: Reference implementation settings.

Quantity	Symbol	Value	Quantity	Symbol	Value
Trajectory horizon	T	32	Executed states	K	8
ODE steps	J	20	State spacing	δ	0.10 s
Voxel size	–	0.15 m	Clearance	d_{safe}	0.45 m
Maximum speed	–	2.0 m/s	Maximum acceleration	–	3.0 m/s^2
Occupancy threshold	$o_{\max}$	0.65	Hidden width	–	256
Field width	d_h	32	Language width	–	128
Batch size	–	8	Learning rate	–	2×10^{-4}
$\mathcal{L}_{\text{sem}}$ weight	λ_s	0.10	$\mathcal{L}_{\text{col}}$ weight	λ_c	0.20
$\mathcal{L}_{\text{dyn}}$ weight	λ_d	0.05	$\mathcal{L}_{\text{unc}}$ weight	λ_u	0.01

The reference optimizer is AdamW with learning rate 2×10^{-4}, weight decay 10^{-4}, gradient-norm clipping at 1, and 50 epochs. All normalization statistics are computed from the training split only. Any experiment using a nonzero consistency weight λ_f, a learning-rate schedule, or settings different from Table 4 must store those values in the checkpoint metadata and report them with the result; they are not inferred from the recommended defaults.

A.2 Trajectory coordinates and training samples

At every replanning event, positions are translated to the current vehicle position and rotated into the current yaw frame. For channel q, training-only mean μ_q and scale $s_q > 0$ define $\bar{x}_{i,q} = (x_{i,q} - \mu_q)/s_q$. The inverse transform is applied before safety checking and control. This local parameterization removes arbitrary global translation and yaw while preserving metric distances after de-normalization.

One stochastic training sample is constructed as follows. First draw a condition–trajectory pair $(\mathbf{c}, \mathbf{X}_1)$ from the training split, then draw $\mathbf{X}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and one $\tau \sim \mathcal{U}[0, 1]$ for the complete trajectory. Form $\mathbf{X}_\tau = (1 - \tau)\mathbf{X}_0 + \tau\mathbf{X}_1$, evaluate the semantic-geometric field at the position channels of $\mathbf{X}_\tau$, and regress the shared target $\mathbf{X}_1 - \mathbf{X}_0$. Using one flow time per trajectory preserves correlations between its states; independently noising each waypoint would define a different conditional path.

Finite differences are evaluated in physical units. In particular, $(\mathbf{D}_1\mathbf{P})_i = (\mathbf{p}_{i+1} - \mathbf{p}_i)/\delta$ and $(\mathbf{D}_2\mathbf{P})_i = (\mathbf{p}_{i+2} - 2\mathbf{p}_{i+1} + \mathbf{p}_i)/\delta^2$. Padding is not used in $\mathcal{L}_{\text{dyn}}$, so endpoints are not counted as artificial zero-velocity or zero-acceleration targets. Occupancy, relevance, and uncertainty are obtained by the same trilinear interpolator at training and inference time.

A.3 Receding-horizon inference

The deployment loop is deterministic conditional on the sampled initial trajectory and current sensor state:

1. Synchronize the latest RGB-D/LiDAR packet and odometry, transform them into the local frame, and update occupied voxels, semantic features, and uncertainty.
2. Encode the instruction, observation history, and state once to obtain $\mathbf{c}$. Sample $\mathbf{X}_0$ or reuse the shifted previous solution as a warm start plus small Gaussian perturbation.

3. Integrate $d\mathbf{X}_\tau/d\tau = \mathbf{V}_\theta(\mathbf{X}_\tau, \tau, \mathbf{c})$ for J steps. De-normalize the terminal trajectory and reject any state that violates speed, acceleration, workspace, or occupancy limits.
4. For each rejected position, solve the local projection in Eq. equation 11, re-evaluate the nonlinear signed distance, and relinearize if necessary. If the subproblem is infeasible or sensor timestamps exceed the permitted age, publish a hover/stop command instead of a trajectory.
5. Execute only the first K states, retain the unexecuted suffix as a warm start, acquire a new observation, and repeat.

The nonlinear post-check in Step 4 is essential: feasibility of a linearized constraint is only a local surrogate and is not presented as a global collision-free guarantee.

A.4 Computational cost

Let M and N denote the numbers of language and observation tokens, d_a the attention width, C_V the cost of one velocity-network evaluation, and Q the number of field queries. Cross-attention costs $O(MNd_a)$ time and $O(MN)$ attention memory. Trilinear interpolation costs $O(Qd_h)$ because each query touches eight corners. ODE generation costs $O(JC_V)$, and finite-difference penalties cost $O(Td_x)$. The safety stage solves three-variable convex subproblems only for rejected states; its practical cost therefore scales with the number of violations rather than with all map voxels. These expressions separate architectural scaling from hardware-dependent latency in Table 1.

B Conditional Flow-Matching Derivations

For compactness, vectorize a trajectory matrix and write $D = Td_x$, $\mathbf{Y} = \mathbf{X}_1 - \mathbf{X}_0$, and $\mathbf{Z} = (\mathbf{X}_\tau, \tau, \mathbf{c})$. All expectations below are conditional on events of positive density, and square integrability is assumed.

Theorem 1 (Pointwise optimum of conditional flow matching)**.** *Among all measurable fields $\mathbf{V} : \mathbb{R}^D \times [0,1] \times \mathcal{C} \to \mathbb{R}^D$ with finite second moment, the population objective $\mathbb{E}\|\mathbf{V}(\mathbf{Z}) - \mathbf{Y}\|_2^2$ is minimized almost surely by*

$$\mathbf{V}^\star(\mathbf{X}, \tau, \mathbf{c}) = \mathbb{E}[\mathbf{X}_1 - \mathbf{X}_0 \mid \mathbf{X}_\tau = \mathbf{X}, \tau, \mathbf{c}]. \tag{12}$$

The minimizer is unique up to sets having zero probability under $\mathbf{Z}$.

Proof. Let $\mathbf{m}(\mathbf{Z}) = \mathbb{E}[\mathbf{Y} \mid \mathbf{Z}]$. Add and subtract $\mathbf{m}$ and condition on $\mathbf{Z}$:

$$\mathbb{E}\|\mathbf{V} - \mathbf{Y}\|_2^2 = \mathbb{E}\|\mathbf{V} - \mathbf{m}\|_2^2 + \mathbb{E}\|\mathbf{m} - \mathbf{Y}\|_2^2 \tag{13}$$

$$+ 2\mathbb{E}\langle \mathbf{V} - \mathbf{m}, \mathbf{m} - \mathbf{Y}\rangle. \tag{14}$$

The cross term is zero because $\mathbf{V} - \mathbf{m}$ is measurable with respect to $\mathbf{Z}$ and $\mathbb{E}[\mathbf{m} - \mathbf{Y} \mid \mathbf{Z}] = \mathbf{0}$. The second term does not depend on $\mathbf{V}$, while the first is nonnegative and vanishes exactly when $\mathbf{V} = \mathbf{m}$ almost surely. □

Theorem 2 (Marginal continuity equation)**.** *Fix a condition $\mathbf{c}$. Suppose the interpolation $\mathbf{X}_\tau = (1-\tau)\mathbf{X}_0 + \tau\mathbf{X}_1$ has a differentiable density p_τ, and the conditional expectation in Theorem 1 is integrable. Then p_τ and $\mathbf{V}^\star$ satisfy, in the weak sense,*

$$\partial_\tau p_\tau + \nabla_{\mathbf{X}} \cdot (p_\tau \mathbf{V}^\star) = 0. \tag{15}$$

If $\mathbf{V}^\star$ is additionally continuous in τ, globally Lipschitz in $\mathbf{X}$, and of at most linear growth, its ODE flow transports p_0 to p_τ, hence to p_1 at terminal time.

Proof. For any compactly supported smooth test function φ, differentiation under the expectation gives

$$\frac{d}{d\tau}\mathbb{E}[\varphi(\mathbf{X}_\tau)] = \mathbb{E}[\nabla\varphi(\mathbf{X}_\tau)^\top(\mathbf{X}_1 - \mathbf{X}_0)] \tag{16}$$

$$= \int \nabla\varphi(\mathbf{X})^\top \mathbf{V}^\star(\mathbf{X}, \tau, \mathbf{c}) p_\tau(\mathbf{X})\, d\mathbf{X} \tag{17}$$

$$= -\int \varphi(\mathbf{X}) \nabla_{\mathbf{X}} \cdot (p_\tau \mathbf{V}^\star)\, d\mathbf{X}, \tag{18}$$

where the second equality is the tower property and the last is integration by parts. This is the weak continuity equation. The stated Lipschitz and growth assumptions give a unique ODE flow; the pushforward density induced by that flow solves the same continuity equation with the same initial density. Uniqueness of the transported measure yields the claim. □

What the result does and does not establish. The theorems concern the population field with exact optimization. A finite neural network, finite data, and numerical integration introduce approximation, estimation, and solver error, respectively. Moreover, correct marginal transport does not imply that every sampled trajectory is dynamically or collision feasible; this is why the auxiliary losses and inference-time safety filter remain necessary.

C Proofs and Numerical Details

Proposition 1 (Composite interior-state error)**.** *Under the assumptions in Proposition 1, let* $f(u) = F(\gamma(u))$ *and divide* $[0, \Delta]$ *into* m *intervals of width* $a = \Delta/m$*. The composite trapezoidal estimate* $\widehat{I}_m$ *satisfies*

$$\left| \widehat{I}_m - \int_0^\Delta f(u)\, du \right| \le \frac{H\Delta^3}{12m^2}. \tag{19}$$

Proof. Affinity gives $\gamma'(u) = \mathbf{d}$ and $\gamma''(u) = \mathbf{0}$, hence $f''(u) = \mathbf{d}^\top \nabla^2 F(\gamma(u))\mathbf{d}$ and $|f''(u)| \le H$. On one subinterval $[u_j, u_{j+1}]$, Taylor's theorem with integral remainder gives the trapezoidal error $E_j = -f''(\xi_j)a^3/12$ for some ξ_j in that subinterval. Therefore $|E_j| \le Ha^3/12$. The triangle inequality and m equal intervals yield $|\sum_j E_j| \le mHa^3/12 = H\Delta^3/(12m^2)$. Setting $m = 1$ gives the endpoint-only bound. □

The unit-speed affine assumption is substantive. For a general twice differentiable curve,

$$\frac{d^2}{du^2} F(\gamma(u)) = \gamma'(u)^\top \nabla^2 F(\gamma(u))\gamma'(u) + \nabla F(\gamma(u))^\top \gamma''(u), \tag{20}$$

so a bound must also include speed and curvature. Accordingly, Proposition 1 is a local statement for each piecewise-affine segment, not a global superiority guarantee over every discrete policy.

Lemma 1 (Explicit Euler terminal error)**.** *Let* $\dot{\mathbf{X}} = \mathbf{V}(\mathbf{X}, \tau)$ *on* $[0, 1]$*, where* $\mathbf{V}$ *is* L_v*-Lipschitz in* $\mathbf{X}$*, and assume the exact solution obeys* $\|\ddot{\mathbf{X}}(\tau)\|_F \le B$*. With* $h = 1/J$*, explicit Euler satisfies*

$$\|\mathbf{X}(1) - \mathbf{X}_J\|_F \le \frac{Bh}{2L_v}\left((1 + hL_v)^J - 1\right) \le \frac{Bh}{2L_v}(e^{L_v} - 1) \tag{21}$$

for $L_v > 0$*, with the continuous limit* $Bh/2$ *when* $L_v = 0$*.*

Proof. Taylor expansion of the exact solution gives $\mathbf{X}(\tau_{j+1}) = \mathbf{X}(\tau_j) + h\mathbf{V}(\mathbf{X}(\tau_j), \tau_j) + \boldsymbol{\rho}_j$ with $\|\boldsymbol{\rho}_j\|_F \le Bh^2/2$. If $e_j = \|\mathbf{X}(\tau_j) - \mathbf{X}_j\|_F$, Lipschitz continuity gives $e_{j+1} \le (1 + hL_v)e_j + Bh^2/2$ and $e_0 = 0$. Summing the resulting geometric series proves the first inequality; $(1 + L_v/J)^J \le e^{L_v}$ proves the second. □

Proposition 2 (Well-posed local safety projection)**.** *Suppose* $\mathcal{C}_t$ *is a nonempty closed convex polyhedron and the linearized clearance halfspace has a nonempty intersection with* $\mathcal{C}_t$*. Then Eq. equation 11 has a unique minimizer.*

Proof. The feasible region is the intersection of a closed convex polyhedron and a closed halfspace, hence is closed and convex. The objective is coercive and strictly convex, so existence follows from closedness and nonemptiness and strict convexity gives uniqueness. This proposition establishes well-posedness of the local subproblem only; exact nonlinear clearance follows from the post-check and repeated replanning, not from linearization alone. □

D Evaluation Definitions and Reporting Protocol

For N_{ep} episodes, let $S_n \in \{0, 1\}$ indicate success, ℓ_n be the shortest feasible-path length, p_n the executed-path length, e_n the terminal goal distance, C_n indicate at least one collision, and q_n/Q_n be the fraction of annotated instruction subgoals completed. The reported metrics are

$$\text{SR} = \frac{1}{N_{\text{ep}}} \sum_n S_n, \qquad \text{SPL} = \frac{1}{N_{\text{ep}}} \sum_n S_n \frac{\ell_n}{\max(\ell_n, p_n)}, \tag{22}$$

$$\text{NE} = \frac{1}{N_{\text{ep}}} \sum_n e_n, \qquad \text{CR} = \frac{1}{N_{\text{ep}}} \sum_n C_n, \qquad \text{SC} = \frac{1}{N_{\text{ep}}} \sum_n \frac{q_n}{Q_n}. \tag{23}$$

An episode is successful only when the stop decision occurs within the evaluation radius and all mandatory semantic subgoals are satisfied. Contact with the safety cage during a controlled emergency stop is still counted as a collision. Path length is accumulated from executed odometry rather than planned waypoints. Replanning latency includes condition encoding, ODE integration, nonlinear validation, and projection, but excludes sensor exposure and low-level flight-controller delay.

To prevent hidden protocol advantages, baseline and proposed-model evaluations must share scene splits, episode seeds, instruction text, sensor range and rate, stopping radius, collision geometry, time limit, and low-level controller. Per-episode records should include the random seed, success flag, path length, terminal error, collision timestamp, completed subgoals, number of projections, and latency samples. Aggregate SR and CR should be accompanied by binomial confidence intervals; SPL, NE, SC, and latency should be accompanied by episode-level bootstrap intervals over identical resamples. Model selection must use validation episodes only, with the test split evaluated once after hyperparameters are frozen.