

EviNav: Evidence-Driven Grounding and Validated Termination for Zero-Shot Instance Navigation

Juzhou Chen

`jc2145@sussex.ac.uk`

University of Sussex

Gaojing Zhang

University of Sussex

Phil Birch

University of Sussex

Bao Nguyen

University of Sussex

Research Article

Keywords: Zero-shot instance navigation, Instance-Image navigation, Text-goal navigation, Evidence fusion, Frontier exploration, Terminal validation

Posted Date: October 20th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11430716/v1>

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

[Read Full License](#)

Additional Declarations: No competing interests reported.

EviNav: Evidence-Driven Grounding and Validated Termination for Zero-Shot Instance Navigation

Juzhou Chen^{1*}, Gaojing Zhang¹, Phil Birch¹, Bao Kha Nguyen¹

^{1*}Engineering and Design, University of Sussex, Sussex House, Brighton, BN1 9RH, East Sussex, UK.

*Corresponding author(s). E-mail(s): jc2145@sussex.ac.uk;
Contributing authors: gz69@sussex.ac.uk; p.m.birch@sussex.ac.uk;
B.K.Nguyen@sussex.ac.uk;

Abstract

Zero-shot instance navigation requires autonomous robots to locate a specific object from a reference image or text description within a fixed action budget. Reliable navigation demands distinguishing similar objects, acquiring informative views, and deciding when to stop. We present EVI_{NAV}, an online, training-free framework that accumulates detections, feature correspondences, and vision-language model judgments over persistent three-dimensional target hypotheses while separating motion control from termination. For image goals, it recovers the reference camera pose from depth-lifted feature correspondences, aggregates pose estimates across frames, and verifies the target from an aligned viewpoint. For text goals, it grounds candidates in a scene graph and tracked detections, aggregates verification scores across distinct viewpoints, and actively acquires new views when evidence is inconclusive. Both modalities use ground-filtered occupancy mapping, exploration ranked by expected utility per action, and a budget-aware return to the strongest previously observed candidate. Evaluated on 1,000 Habitat-Matterport 3D validation episodes per modality, EVI_{NAV} achieves success rates of 84.1% for Instance-Image Navigation and 55.0% for Text-Goal Navigation, with success weighted by path length scores of 35.2% and 17.4%, respectively. Across 61 ablations, reference-view recovery and multi-view verification are the primary performance drivers, while ground-plane filtering and budget-aware fallback return improve success rates in both tasks. A physical robot study succeeds in 20 of 24 trials across two indoor environments.

Keywords: Zero-shot instance navigation, Instance-Image navigation, Text-goal navigation, Evidence fusion, Frontier exploration, Terminal validation

1 Introduction

Instance navigation requires a robot to locate a specific physical object rather than any member of its category. The target is specified by a reference photograph in Instance-Image Navigation (IIN) (Krantz et al. 2022) and by a

natural-language description in Text-Goal Navigation (TN) (Sun et al. 2024). In both settings, the agent explores an unseen environment under a fixed action budget. Locating an object of the right category is therefore insufficient: the agent must distinguish the target instance from look-alikes, reach a viewpoint from which it can be

verified, and decide when the accumulated evidence justifies stopping. We study the *zero-shot, online* setting, without task-specific training or pre-built maps.

Candidate verification is a major bottleneck in modular navigation pipelines. In a diagnostic study of 100 failed Mod-IIN episodes, re-identification false negatives and false positives accounted for 41% and 23% of failures, respectively, whereas exploration failures accounted for only 13% (Krantz et al. 2023). Recent methods address verification through repeated keypoint matching (Lei et al. 2024), scene-graph matching (Yin et al. 2025), or viewpoint-aware attribute and relation reasoning (Jang and Kim 2026). Building on these insights, we focus on how to maintain evidence over 3D object hypotheses across time, select viewpoints to inspect ambiguous candidates, and tie stopping decisions to accumulated evidence under a finite step budget.

Three practical challenges complicate this process. First, category matches, partial contextual overlap, and repeated queries from nearly identical viewpoints can all reinforce a false positive. Second, an object’s 3D centroid is rarely an ideal location for the robot to stand: matched surface points may lie on an occluded side of furniture or inside occupied space. Third, inspection and exploration compete for the same step budget: rotating to inspect a candidate consumes actions, and continuing to explore may leave too few steps to return to a strong candidate found earlier. An effective navigation framework must therefore track not only where candidate objects lie, but also how much independent evidence supports them and how many actions are required to inspect or revisit them.

We introduce EVINAV, a framework that organizes perception, exploration, and stopping around persistent 3D target hypotheses (Fig. 1). Object detections, local feature correspondences, and vision-language model (VLM) judgments are accumulated on these 3D hypotheses, and the **STOP** action is governed by an explicit verification pipeline rather than triggered automatically upon reaching a motion waypoint. Three core components drive performance:

Reference viewpoint estimation for image goals. A reference photograph specifies not only the target’s appearance, but also a valid camera pose from which the object is visible. EVINAV

lifts 2D feature correspondences to 3D using depth, recovers the reference camera pose via Perspective-n-Point (PnP), averages consistent pose estimates across frames, navigates to the recovered camera position, and rotates toward the target before querying a VLM to verify instance identity. While reference-camera localization has been used in general image-goal navigation (Wasserman et al. 2022; Guo et al. 2025), combining multi-frame PnP pose tracking with view-aligned instance verification proves critical for instance navigation: navigating to the matched object surface instead of the recovered reference viewpoint reduces IIN success rate by 19.6 percentage points (Sec. 4.4).

Multi-view verification for text goals. Because text descriptions provide no reference camera pose, target identity must be established through visual verification. Candidate objects are proposed by an online 3D surfel graph supplemented by tracked 3D detections for objects not yet merged into the graph. To evaluate a candidate, the agent projects its 3D geometry into the current camera frame and queries a VLM for separate appearance and context scores. Because repeated VLM queries from similar poses are strongly correlated, scores are averaged within nearby viewpoints and summed across distinct viewpoints, with a log-odds decision threshold inspired by Wald’s idea of sequential evidence accumulation (Wald 1945). When a candidate remains inconclusive, the agent actively moves to a new viewpoint to inspect it. Replacing active multi-view inspection and viewpoint grouping with same-pose re-querying reduces TN success rate by 15.6 percentage points.

Action-aware exploration and fallback return. Both modalities share a 2D occupancy map with ground-plane height filtering and a controller that accounts for discrete action costs. Frontier candidates are ranked by expected exploration value per action, accounting for both path distance and initial in-place turns (Dai et al. 2020). In addition, the agent records the best candidate viewpoint encountered during the episode and reserves enough steps to return to it if subsequent search fails to confirm a target before the budget expires. For image goals, pose-tagged CLIP similarity scores, weak geometric matches, and panoramic VLM queries provide auxiliary search cues to guide exploration before local feature matching succeeds.

We evaluate EVINAV in Habitat (Savva et al. 2019) on the HM3D IIN and TN validation splits (1,000 episodes each) using frozen off-the-shelf models and no learned navigation policy. EVINAV achieves 84.1% success rate (SR) and 35.2% success weighted by path length (SPL) on IIN, surpassing the strongest baselines in Table 1 by 13.9 SR points and 1.7 SPL points, and achieves 55.0% SR and 17.4% SPL on TN, outperforming prior methods by 28.8 SR points and 6.0 SPL points. Across 61 ablations, removing ground-plane height filtering or the budget-aware fallback return reduces SR by 8.3 and 2.6 points on IIN and by 7.2 and 7.8 points on TN. Under an identical VLM on a 36-episode runtime evaluation, EVINAV achieves higher success rates than UniGoal and Context-Nav while requiring less time per step, fewer VLM queries, and less GPU memory (Sec. 4.5). Substituting four alternative VLMs, including open-weight 2B and 4B models, maintains IIN SR between 81.9% and 84.1%, while TN SR ranges from 29.5% to 56.5% (App. H). Finally, a real-world robot study succeeds in 20 of 24 trials across office and laboratory environments (Sec. 4.3).

In summary, our main contributions are:

- **Reference viewpoint estimation for image goals:** a method that recovers and tracks the reference camera pose across frames via PnP and verifies the target instance after aligning with the reference viewing direction (Sec. 3.3).
- **Multi-view verification for text goals:** a method that grounds candidates in 3D scene-graph nodes and tracked detections, verifies projected candidate regions in the current view, aggregates log-odds evidence across distinct viewpoints, and actively acquires new viewpoints when uncertain (Sec. 3.4).
- **Action-aware search and fallback return:** a shared navigation controller that decouples motion control from stopping validation, ranks frontiers by utility per action on a ground-filtered occupancy map, and reserves budget to return to the strongest previously observed candidate (Secs. 3.2 and 3.5).
- **Comprehensive empirical evaluation:** experiments across 2,000 HM3D validation episodes, 61 ablations, runtime and model-substitution studies, and 24 real-robot trials (Sec. 4).

2 Related Work

Instance navigation and evaluation protocols.

Unlike category-level ObjectNav, instance navigation requires finding a specific physical object. IIN defines the target with a reference photograph (Krantz et al. 2022), whereas TN provides a free-form text description (Sun et al. 2024); related formulations consider personalized references (Barsellotti et al. 2024), collaborative dialogue (Taioli et al. 2025), and lifelong multi-target sequences (Khanna et al. 2024). End-to-end and reinforcement-learning (RL) approaches such as OVRL-v2 (Yadav et al. 2023a), FGPrompt (Sun et al. 2023), and DEBiT (Bono et al. 2024) train goal-conditioned policies directly, while IEEV combines an RL exploration policy with keypoint-based verification (Lei et al. 2024). Other recent methods, including GaussNav, GauScoreMap, and MG-Nav, build scene representations from trajectories collected prior to the evaluation episode (Lei et al. 2025; Deng et al. 2026a; Wang et al. 2025b). In contrast, EVINAV operates strictly online in unseen environments without task-specific training, learned control policies, or pre-collected maps.

Training-free navigation with foundation models.

Zero-shot modular agents typically combine pre-trained detectors, VLMs, and large language models (LLMs) with an online map, occasionally delegating local motion to a pretrained point-goal policy (Yokoyama et al. 2024; Jang and Kim 2026) or a fine-tuned segmentation network (Yu et al. 2023). Early systems pair open-vocabulary detection with nearest-frontier exploration (Gadre et al. 2023). Subsequent methods guide frontier selection using LLM priors (Zhou et al. 2023; Yu et al. 2023; Kuang et al. 2024), scene-graph relational priors (Chen et al. 2023), vision-language similarity maps (Yokoyama et al. 2024), LLM reasoning over 3D scene graphs (Yin et al. 2024), or adaptive switching between semantic and geometric frontiers (Zhang et al. 2025). Other frameworks use LLMs to synthesize multi-source value maps (Long et al. 2024), compose navigation programs (Ziliotto et al. 2025), or select actions directly from depth-derived candidates (Goetting et al. 2025). For instance-level

goals, Mod-IIN (Krantz et al. 2023) combines frontier exploration with local feature matching; GOAT (Chang et al. 2024) matches image and language goals against an instance-aware memory; UniGoal (Yin et al. 2025) unifies object, image, and text goals via graph matching; and Context-Nav (Jang and Kim 2026) guides search with caption-based value maps. EViNAV follows this modular paradigm but restricts VLM calls to initial goal parsing, targeted panoramic search queries, and geometrically gated candidate verification, avoiding per-step LLM/VLM action selection and substantially reducing query overhead (Sec. 4.5).

Scene representations.

Instance navigation requires a spatial representation that links 3D object hypotheses to navigable free space. Incremental depth segmentation (Tateno et al. 2015) forms the basis of SceneGraphFusion (Wu et al. 2021), which predicts closed-set object and relation labels with a graph neural network, while Hydra (Hughes et al. 2022) constructs hierarchical 3D scene graphs online from closed-set semantics. Open-vocabulary representations such as Concept-Graphs (Gu et al. 2024), HOV-SG (Werby et al. 2024), and VLMaps (Huang et al. 2023) attach vision-language features to 3D graphs or top-down grids, but rely on offline map construction from pre-recorded RGB-D sequences. EViNAV combines incremental surfel segmentation (Tateno et al. 2015; Wu et al. 2021) with asynchronous confidence-weighted open-vocabulary segmentation from YOLOE (Wang et al. 2025a), maintaining the 3D scene graph in a shared coordinate frame with a 2D occupancy grid that filters out elevated surfaces and drop-offs by verifying ground-plane support.

Exploration under an action budget.

Classical and learned exploration methods typically trade information gain against travel distance or continuous flight time (Yamauchi 1997; Zhou et al. 2021; Chaplot et al. 2020; Zhou et al. 2025; Yin et al. 2025). Semantic target-search methods also prioritize informative viewpoints: STEM propagates semantic priorities from observed objects to nearby 3D frontiers and plans viewpoint sequences to reduce search time for

aerial robots (Sethi et al. 2026). In continuous-time aerial exploration, Dai et al. (2020) rank candidate views by information gain divided by travel time (including yaw rotation), eliminating the need to tune a relative weight between gain and cost. We adapt this rate formulation to a discrete-action ground agent by expressing both path length and initial rotation in step units; by Dinkelbach’s lemma (Dinkelbach 1967), maximizing this ratio is equivalent to maximizing an additive objective whose action cost weight adapts automatically to the best available frontier. Furthermore, motivated by the observation in GOAT (Chang et al. 2024) that selecting the best hypothesis across an entire trajectory outperforms fixed single-step thresholds, we introduce a budget-aware fallback return that tracks the strongest candidate observed online and reserves enough actions to revisit it before the episode horizon expires.

Instance verification and stopping viewpoints.

Active perception couples sensing decisions to the information needed for a task (Bajcsy et al. 2018); for example, Patten et al. (2018) plan observation sequences under a time constraint to improve object classification while accounting for uncertainty in object class and pose. Prior works verify candidate instances through keypoint re-checks during approach (Lei et al. 2024), scene-context aggregation (Yin et al. 2024), sub-graph matching (Yin et al. 2025), attribute and spatial-relation scoring (Jang and Kim 2026), multi-frame observation (Zhang et al. 2025), or direct VLM stopping decisions (Goetting et al. 2025; Cao et al. 2025). For text goals, EViNAV grounds verification in the 2D projection of the tracked 3D candidate and accumulates log-odds scores across distinct viewpoints through a decision rule inspired by Wald’s sequential evidence accumulation (Wald 1945), actively moving to new viewpoints when a single view is ambiguous (Sec. 3.4). Choosing where to stand for verification and termination is equally important. While Mod-IIN and GOAT navigate toward localized object surfaces (Krantz et al. 2023; Chang et al. 2024) and MSGNav selects viewpoints based on target visibility (Huang et al. 2026), image-goal methods such as SLING (Wasserman et al.

2022), IGL-Nav (Guo et al. 2025), and AnyImageNav (Deng et al. 2026b) estimate the reference camera pose via PnP (Lepetit et al. 2009; Fischler and Bolles 1981) or 3D registration. EVINAV adapts reference-camera recovery to instance navigation by clustering and averaging PnP solutions across frames, navigating to the recovered camera position, and aligning with the target before running pairwise VLM verification.

3 Method

3.1 Problem Setting and Overview

Task setting.

EVINAV receives either a free-form text description g^{text} or an instance-level reference image g^{img} and navigates an unseen environment within a budget of B discrete actions. At each step t , the agent receives an RGB-D observation (I_t, D_t) and its world-frame camera pose $T_{c,t}$, and selects an action $a_t \in \{\text{STOP}, \text{MOVE_FORWARD}, \text{TURN_LEFT}, \text{TURN_RIGHT}\}$.

Candidate tracking, verification, and termination.

Each RGB-D frame updates a shared world-frame map (Sec. 3.2) together with a set of persistent 3D target hypotheses: estimated reference camera poses for image goals (Sec. 3.3) and candidate 3D objects with accumulated verification scores for text goals (Sec. 3.4). Detections, keypoint correspondences, and VLM verification outputs are associated with these 3D hypotheses in world coordinates rather than evaluated independently per frame. Navigation and stopping are decoupled: exploration, path planning, and visual servoing generate movement actions, whereas issuing **STOP** during normal operation requires both visual verification of the target instance and confirmation that the agent is within stopping range. Depending on the verification outcome, the agent may stop, move closer or to a different viewpoint for another check, or reject the candidate and resume search. Outside of verified stops, **STOP** is issued only when completing a budget-aware fallback return or when reaching the episode horizon B (Sec. 3.5.3).

Organization.

Section 3.2 describes the shared scene representation and ground-plane traversability filtering. Section 3.3 details reference camera pose estimation and aligned pairwise verification for image goals. Section 3.4 presents candidate grounding and multi-view log-odds verification for text goals. Section 3.5 introduces action-normalized frontier exploration, budget-aware fallback return, and controller arbitration. Additional implementation details and formal specifications are provided in Apps. A–E, with key hyperparameters summarized in Table B1.

3.2 Shared Scene Representation

Target grounding requires persistent 3D object representations, path planning requires accurate obstacle maps, and exploration requires tracking unobserved areas. We maintain all three in a common world frame from the same RGB-D stream so that candidate selection and trajectory planning evaluate reachability on the exact same spatial representation.

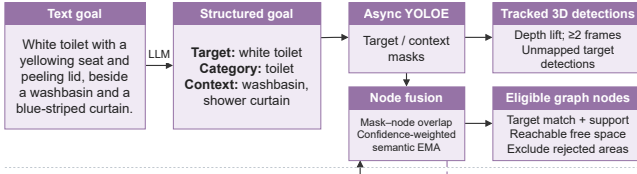
Surfel scene graph with open-vocabulary semantics.

Incoming RGB-D frames are fused into an incremental surfel map and segmented into persistent 3D object nodes following Tateno et al. (2015) and Wu et al. (2021). Each scene-graph node is represented as

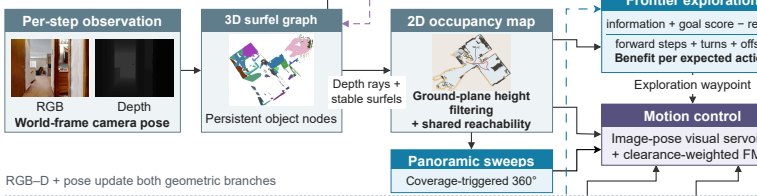
$$v = (\ell_v, c_v, n_v, \boldsymbol{\mu}_v, [\mathbf{b}_v^{\min}, \mathbf{b}_v^{\max}], \boldsymbol{\pi}_v), \quad (1)$$

where ℓ_v and c_v denote the node’s top semantic label and confidence, n_v is its surfel count, $\boldsymbol{\mu}_v$ is its 3D centroid, $[\mathbf{b}_v^{\min}, \mathbf{b}_v^{\max}]$ is its axis-aligned bounding box, and $\boldsymbol{\pi}_v$ is an empirical probability distribution over the active detector queries for text goals. Rather than using SceneGraphFusion’s closed-set graph network, we run the open-vocabulary segmenter YOLOE (Wang et al. 2025a) asynchronously in parallel with geometric fusion. Each 2D detection mask is matched to the visible node whose rendered instance mask has the largest overlap, updating $\boldsymbol{\pi}_v$ via a confidence-weighted exponential moving average (App. B).

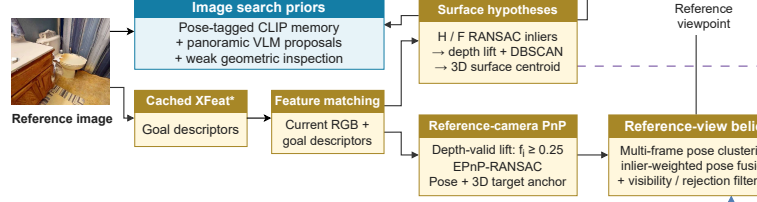
(b) Text goal grounding



(a) Shared RGB-D loop



(c) Instance image grounding



Dashed: priors / evidence updates. H / F: homography / fundamental matrix.

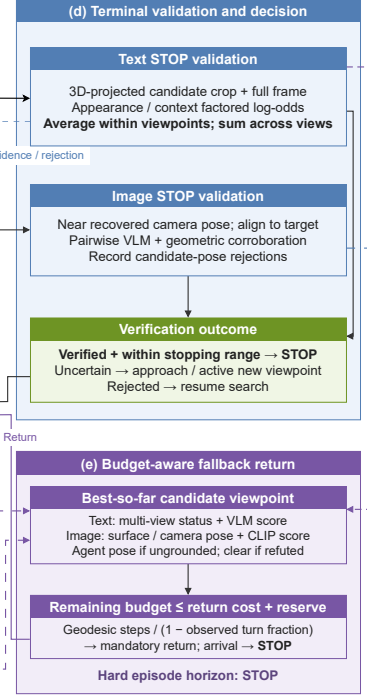


Fig. 1 EviNav overview. (a) Each RGB-D frame updates a 3D surfel scene graph and a 2D occupancy map with ground-plane height filtering, which support action-normalized frontier exploration and path planning (Secs. 3.2 and 3.5.1). (b) Text goals are parsed into target and context detector queries and matched against scene-graph nodes and tracked 3D detections (Sec. 3.4). (c) Image goals are localized through dense local-feature matching, depth lifting, and PnP recovery of the reference camera pose, aggregated across frames; pose-tagged CLIP similarity scores, panoramic VLM queries, and low-confidence feature matches also guide exploration (Secs. 3.3 and 3.5.1). (d) Modality-specific VLM verifiers evaluate candidate stops by approving termination, triggering a closer approach or a new viewpoint, or rejecting the candidate; for text goals, verification scores are accumulated across distinct viewpoints via a multi-view log-odds decision rule (Secs. 3.3 and 3.4). (e) A budget-aware fallback return tracks the highest-scoring candidate pose so the agent can revisit it before the action budget expires (Sec. 3.5.2).

2D occupancy grid.

Alongside the surfel graph, we maintain a 2D occupancy grid $\mathcal{O}_t$ at 0.05 m resolution that tracks free-space ray traversals, obstacle hits, and observation counts for every cell. Depth rays increment free-space counts along their horizontal projection and add one occupied count when terminating within the agent’s vertical height band; each stable surfel within that height band adds two occupied counts. A cell is classified as free when free counts exceed occupied counts, occupied when at least two occupied counts are present and not outweighed by free counts, and unknown before any ray has traversed it (Eq. (B3), App. B). Requiring two occupied counts prevents isolated depth noise from creating phantom obstacles while allowing persistent surfels to register immediately. Depth pixels at the maximum sensor range are

not raycast as free space, ensuring that distant unobserved areas remain unknown for frontier exploration.

Ground-plane height filtering.

For a wheeled or legged ground robot, a traversable cell must have both obstacle clearance above the floor and continuous physical support at the walking surface. Standard height-band slicing can misclassify low furniture, stair treads, or downward drop-offs (such as stairwells) as free space when camera rays pass above or beyond them. To prevent this, we track the floor elevation $\hat{F}$ online as the mode of a cumulative 1 cm height histogram of depth points near the nominal ground plane (App. B). Depth points within 0.15 m of $\hat{F}$ record confirmed floor support, as do cells directly traversed by the agent. Depth

points above $\hat{F} + 0.15$ m (up to the agent’s height) record elevated obstacle returns, and any depth ray terminating more than 0.15 m below $\hat{F}$ marks all cells beyond its ground-plane intersection as lacking floor support. Any cell that accumulates elevated returns without confirmed floor support is reclassified from free to occupied.

Shared reachability and path planning.

The resulting free-space mask of $\mathcal{O}_t$ defines the traversable region across all modules. Running Dijkstra’s algorithm (Dijkstra 1959) from the agent’s current cell yields the geodesic distance field d_{geo} . Frontier segments for exploration (Sec. 3.5.1) are extracted at the boundary between free and unknown cells. Any 3D target candidate is projected onto the nearest free cell within radius r and evaluated by its geodesic distance d_{geo} , immediately filtering out unreachable candidates without requiring a separate map-matching step. Point-to-point navigation executes on the same free-space mask using a clearance-weighted fast-marching planner (App. D.2).

3.3 Image Goals: Reference-View Recovery and Aligned Verification

An instance-image goal depicts a specific physical object from a particular camera perspective. Navigating directly toward the matched object surface may lead to an obstructed or uninformative view. Our image-goal branch therefore estimates the 3D pose of the camera that captured the reference image, aggregates these estimates across frames, and verifies instance identity after aligning the agent’s camera with the target.

Feature matching and 3D surface localization.

At each step, we match the current RGB frame against the reference image using XFeat* (Potje et al. 2024), which provides per-match fine and coarse confidence scores f_i and ς_i . Because the reference image is constant throughout the episode, its dense descriptors are computed once at the start. Matched keypoints feed two parallel geometric estimators. First, to localize the visible object surface directly, we fit a 2D homography (with a fundamental-matrix fallback) via RANSAC, lift

the current-frame inliers into 3D using the depth map D_t and camera pose $T_{c,t}$, cluster them on the ground plane, and retain the confidence-weighted centroid of the largest cluster when it satisfies minimum inlier and score thresholds (App. C). This 3D surface centroid guides closer inspection of low-confidence matches (Sec. 3.5.1) and serves as a fallback location for budget-aware return (Sec. 3.5.2), whereas primary goal-directed navigation targets the recovered reference camera pose described next.

Reference camera pose estimation.

Localizing an object’s surface indicates where the object lies, but not where the agent can stand to observe it clearly: a matched surface may lie behind other furniture or face a wall. Because the reference photograph was necessarily captured from a viewpoint with a clear line of sight to the target, we estimate the reference camera pose directly from the 3D-lifted matches. Matches with fine confidence $f_i \geq 0.25$ and valid depth provide 3D world points $\mathbf{x}_k$ paired with 2D key-points $\mathbf{u}_k^g$ in the reference image, defining the Perspective- n -Point (PnP) problem

$$(\mathbf{R}_g, \mathbf{t}_g) = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_{k \in \mathcal{I}} \|\pi(\mathbf{R}\mathbf{x}_k + \mathbf{t}) - \mathbf{u}_k^g\|^2, \quad (2)$$

where $\pi(\cdot)$ projects 3D points using the reference camera’s intrinsic matrix K_g . We solve Eq. (2) using EPnP-RANSAC and accept the estimate only if it passes thresholds on inlier count, inlier ratio, distance, and camera height (App. C). A valid solution yields the reference camera position $\mathbf{C} = -\mathbf{R}_g^\top \mathbf{t}_g$, its viewing direction $\mathbf{f} = \mathbf{R}_g^\top \mathbf{e}_z$, and an estimated 3D target point along the ray $(\mathbf{C}, \mathbf{f})$ at the median depth of the inner half of inliers closest to the reference image center. The agent navigates toward the camera standpoint $\mathbf{C}$ and rotates to face this target point for verification.

Multi-frame pose clustering and filtering.

Because single-frame PnP estimates can be noisy or temporarily drop out when the target leaves the field of view, we cluster and average valid estimates across frames. Each new PnP solution is merged into the nearest cluster within planar distance r , or initializes a new cluster if none lies within r . Within each cluster, the estimated

camera pose is computed as the inlier-weighted average of member solutions lying within distance s of the cluster’s weighted medoid; a cluster is considered *multi-view confirmed* once at least two solutions agree within s . The agent pursues the cluster with the highest cumulative inlier weight. To discard stale false positives, we apply visibility-based negative filtering: a frame counts as a miss only when the cluster’s target point projects inside the camera frustum within sensor range and the measured depth is not occluded by foreground obstacles more than r in front of the target point. A cluster is discarded after 12 unoccluded misses (the number of steps in a full 360° turn). Similarly, negative decisions from the VLM verifier below are recorded against the active cluster; once rejected, a cluster can be reactivated only by new feature matches observed from a position at least r away from previous rejection locations.

Approach and obstacle avoidance.

A valid camera viewpoint $\mathbf{C}$ is not guaranteed to be reachable along a straight line from the agent’s current position. The agent initially steers toward $\mathbf{C}$ using direct bearing control (visual servoing). If a forward step is blocked by an obstacle or 12 steps elapse without reducing the distance to $\mathbf{C}$, the agent switches to the map-based fast-marching planner, which routes to the nearest free cell around $\mathbf{C}$ before handing control back to visual servoing at close range. If no traversable path exists, the agent triggers verification from its current position provided it is already within distance r of a multi-view confirmed target point, and otherwise suspends the cluster (App. D.2).

Aligned pairwise verification.

Reaching the estimated camera position is not by itself sufficient to stop, as local feature matches can occasionally be triggered by repeated furniture or similar architectural textures. We trigger visual verification once the agent enters a verification radius around $\mathbf{C}$: $r - s = 0.75$ m for a multi-view confirmed cluster with no prior rejections, and tightened to the minimum sensor depth $\min D = 0.5$ m for single-frame clusters or previously rejected candidates (App. E.2). Before querying the verifier, the agent rotates in place until its optical axis aligns within $\delta/2$ of the estimated target point so that the candidate instance

is centered in view. Once aligned, a VLM verifier V_{pair} compares the reference and current images (I_g, I_t) and outputs a binary same-instance decision $y_t^{\text{id}} \in \{0, 1\}$, a confidence score, an evidence level $\epsilon_t \in \{\text{strong}, \text{weak}, \text{none}\}$, and an indicator $c_t^{\text{ctx}} \in \{0, 1\}$ of whether the judgment relies primarily on background context. Termination is approved when

$$\bar{y}_t^{\text{id}} = y_t^{\text{id}} \mathbb{I}[\epsilon_t = \text{strong} \vee (\epsilon_t = \text{weak} \wedge c_t^{\text{ctx}} = 0)] = 1, \quad (3)$$

which requires a positive instance match supported either by strong visual evidence or by weak evidence grounded directly in the target object rather than room layout alone. For single-frame (unconfirmed) clusters, stopping additionally requires the current frame to maintain PnP inlier consensus. If verification fails, the rejection is logged and navigation continues; two VLM rejections, a single rejection within $\min D$ of $\mathbf{C}$, or failing to re-observe an unconfirmed target from an aligned viewpoint inside the verification radius permanently discards the cluster (App. E.2).

3.4 Text Goals: Multi-View Grounding and Verification

A text goal specifies an object category, fine-grained attributes, and spatial relations to surrounding objects, but provides no reference image. Open-vocabulary detector scores alone cannot reliably distinguish fine attributes or multi-object spatial relations, and a nearby distractor of the same class may fill the camera view. We therefore separate candidate generation, single-view VLM verification grounded in the candidate’s 3D projection, and multi-view log-odds accumulation across distinct viewpoints. All verification records are indexed by 3D world coordinates rather than transient graph node IDs so that scene-graph merging and re-indexing preserve accumulated evidence.

Goal parsing and candidate generation.

At the start of each episode, an LLM parses the free-form instruction into a structured query tuple

$$\mathcal{G} = (g_{\text{can}}, Q_{\text{tgt}}, Q_{\text{ctx}}), \quad (4)$$

where g_{can} is the canonical target category, $\mathcal{Q}_{\text{tgt}}$ is a set of short detector prompts combining the target noun with its stated color, material, shape, or size modifiers, and $\mathcal{Q}_{\text{ctx}}$ contains detector prompts for surrounding landmark objects mentioned in the instruction (App. G). Feeding both $\mathcal{Q}_{\text{tgt}}$ and $\mathcal{Q}_{\text{ctx}}$ to YOLOE maps both candidate targets and their surrounding context into the scene graph. While context detections help guide frontier exploration (Eq. (12)), only target detections can propose navigation candidates; whether a candidate actually satisfies the stated contextual relations is evaluated by the VLM verifier. Because a node may strongly match a target phrase even if another query has a slightly higher score, we score each node v by its maximum probability over the target query set:

$$q_v = \max_{q \in \mathcal{Q}_{\text{tgt}}} \pi_{v,q}. \quad (5)$$

A scene-graph node becomes an admissible candidate when $q_v \geq 0.2$, $n_v \geq 64$ surfels, and its centroid projects to a geodesically reachable free cell outside previously rejected areas. Because surfel integration can lag behind 2D detection for small, thin, or partially visible objects, we also track depth-lifted 3D detections not yet merged into a qualifying graph node: any target detection observed in at least two frames forms a candidate if no scene-graph candidate lies within radius r , and is snapped to free space, filtered by reachability, and ranked alongside graph nodes (App. C).

Candidate-grounded visual verification.

When evaluating a pursued candidate, the VLM must inspect the image region corresponding to *that specific 3D candidate* rather than whichever 2D detection happens to have the highest detector score in the current frame; otherwise, a nearby distractor could trigger a false stop or falsely penalize the pursued node. We project the candidate node’s 3D surfels into the current camera frame to obtain its 2D bounding box along with two geometric visibility ratios: the visible voxel fraction $v_t \in [0, 1]$ (the fraction of the node’s 3D voxels that project inside the camera frustum) and the depth-consistent box fraction $d_t \in [0, 1]$ (the fraction of pixels within the projected 2D box whose measured depth matches the node). The

bounding box passed to the VLM is the highest-confidence 2D detection covering more than half of the projected footprint, or the projected footprint box itself whenever $d_t > 0.5$, allowing verification even when the 2D detector misses the object in the current frame (App. E.1).

The VLM evaluates the cropped candidate box alongside the full frame and returns separate probabilities for object appearance match (p_t^{app}) and surrounding context agreement (p_t^{ctx}). Separating these two channels accounts for their different visibility regimes: a close-up view clearly resolves fine material or color details but crops out neighboring furniture. Let $a_t \in [0, 1]$ denote the fraction of the image area covered by the evaluated bounding box. Clipping p_t^{app} and p_t^{ctx} to $[0.01, 0.99]$, we combine their log-odds weighted by the available surrounding view:

$$L_t = \text{logit}(p_t^{\text{app}}) + (1 - a_t) v_t \text{logit}(p_t^{\text{ctx}}). \quad (6)$$

A single-frame evaluation is positive when $L_t > 0$. When a candidate fills nearly the entire image ($a_t \rightarrow 1$), the context term vanishes so that the verifier is not penalized for surrounding landmarks outside the field of view. We define the combined frame probability as $p_t = \sigma(L_t)$, where $\sigma(\cdot)$ is the logistic sigmoid (or the fallback mapping Π in Eq. (E13), App. E.1, when factored scores are unavailable).

Multi-view evidence accumulation.

We accumulate VLM verification scores for each 3D candidate in world coordinates, merging records whose positions lie within radius r . Each verification query at step t contributes a weighted log-odds score

$$\ell_t = w_t \text{logit}(p_t), \quad (7)$$

with p_t clipped to $[0.01, 0.99]$. The association weight $w_t \in [0, 1]$ reflects how well the evaluated 2D box aligns with the 3D candidate: $w_t = 0$ when no target box is present in the frame (since occlusion or framing misses should not count as identity refutations); $w_t = 1$ when the evaluated box is directly grounded in the node’s projected footprint or comes from a directly inspected 3D detection; and otherwise w_t equals the overlap fraction of the evaluated box with the node’s 2D

projection (set to 0 if the box’s depth-lifted 3D position lies more than r from the node).

Because consecutive frames taken from nearly the same camera pose yield highly correlated VLM judgments, we group observations into pose-distinct *viewpoint clusters* $\{\mathcal{V}_k\}$. Two observations are considered pose-distinct if their camera positions differ by at least $\min D \tan(\phi/2) = 0.41$ m or, when both frames contain a detection, their camera headings differ by at least $\phi/2 = 39.5^\circ$. Each new observation is assigned to the first viewpoint cluster whose initial pose is not pose-distinct from the current pose, or starts a new cluster $\mathcal{V}_k$ otherwise. Within cluster $\mathcal{V}_k$, let W_k denote the total weight, $\bar{\ell}_k$ the weighted mean log-odds, and R_k the maximum association weight:

$$\begin{aligned} W_k &= \sum_{t \in \mathcal{V}_k} w_t, & R_k &= \max_{t \in \mathcal{V}_k} w_t, \\ \bar{\ell}_k &= \frac{\sum_{t \in \mathcal{V}_k} \ell_t}{W_k} \quad (W_k > 0), & E_k &= R_k \bar{\ell}_k, \\ \Lambda &= \sum_{k: W_k > 0} E_k, & \eta &= \log \frac{1 - \epsilon_{\text{seq}}}{\epsilon_{\text{seq}}}. \end{aligned} \quad (8)$$

Averaging within each cluster limits the contribution of repeated VLM queries from the same standpoint, while multiplying by R_k prevents a single weakly overlapping view from receiving full weight. Inspired by Wald’s idea of sequential evidence accumulation (Wald 1945), we use a log-odds decision rule with $\epsilon_{\text{seq}} = 0.02$, giving a decision threshold of $\eta = \log 49 \approx 3.89$.

To evaluate whether to stop at the current step t , we compute the cumulative score S_t obtained by including the current observation in the viewpoint partition:

$$S_t = \sum_{k: W_k^{(+t)} > 0} E_k^{(+t)}, \quad S_t \geq \eta, \quad (9)$$

where $(+t)$ denotes the cluster statistics after incorporating step t . For example, two full-weight views with $p_t = 0.9$ from the same standpoint yield $S_t = \log 9 \approx 2.20 < \eta$, whereas observing $p_t = 0.9$ from two distinct viewpoints yields $S_t = 2 \log 9 \approx 4.39 > \eta$. A single view can cross η on its own only when $p_t \geq 0.98$; moderately confident views require confirmation from a second viewpoint.

Stopping, candidate rejection, and ranking.

Issuing **STOP** directly at step t requires both positive verification and close proximity to the candidate:

$$L_t > 0, \quad S_t \geq \eta, \quad d_{\text{node}} \leq r, \quad d_{\text{ev}} \leq r, \quad (10)$$

where d_{node} is the geodesic distance from the agent to the candidate’s free-space standpoint and d_{ev} is the planar distance to the depth-lifted standpoint of the evaluated 2D bounding box (the condition $d_{\text{ev}} \leq r$ is omitted when no separate box standpoint is active; App. E.1). When a candidate passes verification while the agent is still in the extended approach range $(r, r_{\text{far}}]$ with $r_{\text{far}} = r + s = 1.25$ m, the agent executes a short close-range approach maneuver, centering the target and stepping forward to enter the stopping radius r before terminating (App. E.1).

Within r_{far} , a candidate is also accepted if its accumulated evidence reaches $\Lambda \geq \eta$ via multi-view agreement (Algorithm 5, App. E.1), provided the current view does not exhibit an explicit mismatch ($p_t^{\text{app}} < 0.5$, or $p_t^{\text{ctx}} < 0.5$ when surrounding context is visible). Conversely, a candidate and its radius- r neighborhood are rejected and excluded from future selection once $\Lambda \leq -\eta$, or when repeated observations fail to resolve ambiguity within the allowed viewpoint budget (App. E.1). In addition, if the agent pursues a candidate beyond $r + s$ for 60 steps without reducing its geodesic distance by at least one effective step ($s \cos(\delta/2) = 0.24$ m), the unreachable candidate and its radius- r neighborhood are retired (App. D.2).

For candidates that have neither been accepted nor rejected, accumulated evidence modulates target selection. Visiting candidates strictly in order of distance would waste steps on nearby objects that the verifier has already judged unlikely to match the goal. The agent therefore selects

$$v^* = \arg \min_v \frac{d_{\text{geo}}(v)}{\sigma(\Lambda_v)}, \quad (11)$$

where Λ_v is candidate v ’s accumulated log-odds score. Uninspected candidates ($\Lambda_v = 0$, $\sigma(\Lambda_v) = 0.5$) are ranked by $2d_{\text{geo}}(v)$, whereas positive evidence prioritizes a candidate and negative evidence defers it in favor of more promising alternatives.

Active viewpoint selection.

When a verification query leaves a candidate inconclusive ($-\eta < \Lambda < \eta$), re-querying from the same location adds no new viewpoint cluster. Instead, if the current view is negative or ambiguous, the agent navigates to a new reachable standpoint on the circle of radius d_{node} around the candidate that lies at least 0.41 m from both the current pose and all previous verification poses. At most two such viewpoint changes are attempted per candidate before retiring it as inconclusive. Conversely, if the current view is positive ($L_t > 0$) but cumulative evidence remains below η , the agent steps closer to the candidate to obtain a higher-resolution view rather than re-querying in place (App. E.1).

Handling unmapped detections, distant candidates, and final approach.

Three complementary checks handle common edge cases during text-goal navigation (App. E.1):

1. *Unmapped nearby detections:* When a high-confidence 2D target detection appears at close range while the currently pursued graph node is distant or out of view, the agent temporarily evaluates the depth-lifted 3D detection directly and merges the resulting log-odds evidence into the world-coordinate record at that location.
2. *Distant candidate verification:* Whenever a candidate node is already clearly visible from afar, the agent runs VLM verification from its current pose before travelling to it; a strong refutation prunes the candidate early, while a positive score boosts its ranking in Eq. (11).
3. *Post-approach verification:* When a verified candidate triggers a short centering or forward approach maneuver to enter the stopping radius r , the agent confirms the target after completing the maneuver and before issuing **STOP**. This check requires that the target remain in view with sufficient context or high close-range detector confidence, or that consecutive close-range views agree with a full-frame scene context check.

3.5 Budget-Aware Search and Termination

Operating under a finite horizon of B discrete actions requires balancing exploration, candidate

inspection, and timely termination. Building on the shared traversability map of Sec. 3.2, we rank frontiers by expected value per action, reserve sufficient actions to return to the best previously seen candidate before the budget runs out, and enforce a strict priority order over motion and stopping.

3.5.1 Action-Normalized Frontier Exploration

Before a target is localized, efficient exploration determines whether the goal is encountered within the step budget. Standard frontier utilities take the additive form $U = w_I I + w_G G - w_C C - w_R R$, combining area gain I (in grid cells) with path distance C (in meters). Because I can be on the order of 10^3 cells on a 0.05 m grid while C is on the order of meters, such objectives are sensitive to manual scale tuning and ignore the discrete actions spent rotating in place. Instead, we maximize the net exploration benefit per expected discrete action required to reach and face each frontier.

Frontier value terms.

Connected frontier segments on $\mathcal{O}_t$ are represented by reachable frontier waypoints $\{\mathbf{g}_i\}$ (App. D). For waypoint i , the geometric information gain I_i counts the number of unknown cells inside the camera frustum wedge (half-angle $\phi/2 = 39.5^\circ$, range $r_{\text{max}} = 5.0$ m) oriented along the arrival direction $\mathbf{g}_i - \mathbf{x}_t$. For text goals, the semantic goal score G_i sums the query match scores of mapped scene-graph nodes falling inside the same frustum wedge:

$$G_i = \sum_{v: \mu_v \in \text{wedge}(i)} \max_{q \in \mathcal{Q}_{\text{tgt}} \cup \mathcal{Q}_{\text{ctx}}} \pi_{v,q}, \quad (12)$$

allowing frontiers near candidate targets or relevant landmark objects to outrank larger purely geometric frontiers ($w_G = 150$). For image goals, the open-vocabulary detector is disabled ($G_i \equiv 0$) and visual similarity scores guide search as described below. A revisiting penalty R_i down-weights recently traversed regions and previously rejected locations (App. D).

Action-normalized rate objective.

In a discrete action space, both forward translation and in-place rotation consume steps: a frontier directly ahead requires no initial turn, whereas

one behind the agent requires up to six 30° turn actions before any forward motion occurs. We express the total cost of reaching frontier i in action steps:

$$n_i = \underbrace{\frac{C_i}{s}}_{\text{translation}} + \kappa \underbrace{\frac{|\text{wrap}(\beta_i - \theta_t)|}{\delta}}_{\text{rotation}}, \quad (13)$$

where C_i is the geodesic distance to $\mathbf{g}_i$ on $\mathcal{O}_t$, β_i is the initial bearing to $\mathbf{g}_i$, θ_t is the agent’s current heading, $\text{wrap}(\cdot) \in [-\pi, \pi)$ wraps the heading error, and $\kappa = 1$ weights each turn increment δ as one action. We then score each frontier by its net benefit per action:

$$U_i = \frac{w_I I_i + w_G G_i - w_R R_i}{n_i + c_0/s}, \quad (14)$$

$c_0 = r_{\max}, \quad w_I = 1, \quad w_R = 25,$

selecting $\arg \max_i U_i$. The baseline offset $c_0/s = 20$ steps regularizes the denominator at very short distances, and no separate $-w_R R_i$ term is needed because travel cost already normalizes the objective (App. D).

Connection to adaptive additive weighting.

Writing the numerator as $N_i = w_I I_i + w_G G_i - w_R R_i$, Dinkelbach’s theorem for fractional programming (Dinkelbach 1967) implies that maximizing U_i in Eq. (14) is equivalent to maximizing $N_i - \lambda^*(n_i + c_0/s)$ at the optimal rate $\lambda^* = \max_j U_j$. Because c_0/s is constant across frontiers,

$$\arg \max_i U_i = \arg \max_i N_i - \lambda^* n_i. \quad (15)$$

Whereas a conventional additive utility $N_i - \lambda n_i$ prices every action at a fixed weight λ , the ratio formulation automatically adapts the cost of an action λ^* to the best frontier currently available: steps are priced high when nearby frontiers offer rich information and priced low when only distant frontiers remain. In Sec. 4.4, we compare this parameter-free rate directly against additive baselines whose fixed weights λ are tuned with oracle access to the full validation set.

Panoramic sweeps and visual search priors.

To broaden visual coverage during exploration, the agent executes a 360° panoramic sweep ($\lceil 360^\circ/\delta \rceil = 12$ turns) whenever its planar displacement since the last sweep exceeds $\zeta \max(r_{\max}, L_{t-1})$ and sufficient nearby cells remain unknown (Eq. (D10)). Here, L_{t-1} is the cumulative path length. We set $\zeta = 1$ for text goals and $\zeta = 0.2$ for image goals, since local feature matching requires frequent omnidirectional views to catch the target within sensor range.

For image goals, three visual priors supplement geometric frontier exploration (App. D.1): (1) a spatial buffer of pose-tagged CLIP image-similarity scores biases frontier selection toward directions that visually resemble the reference scene, arbitrated with geometric coverage via an epoch schedule; (2) geometrically consistent XFeat* matches that fall below the full PnP acceptance threshold still trigger a closer inspection visit without authorizing termination; and (3) after a panoramic sweep, the four cardinal views are tiled into a 2×2 grid for a VLM query that selects the most promising direction and localizes a candidate bounding box for closer inspection. All three mechanisms only propose waypoints for closer observation; normal image-goal stopping requires passing the aligned verification check of Sec. 3.3, except for budget-aware fallback arrival and the hard step limit $t = B - 1$ (Sec. 3.5.3).

3.5.2 Budget-Aware Fallback Return

Under a finite budget of B steps, an agent that observes a plausible candidate early in the episode but continues exploring in search of stronger confirmation risks running out of steps far from the target. We therefore maintain a fallback candidate $(\mathbf{x}^*, p^*, t^*)$ recording the best traversable candidate standpoint encountered so far, ranked by multi-view verification status and VLM score p^* for text goals and by geometric grounding and CLIP similarity for image goals (App. E.3). If the corresponding hypothesis is later refuted, the fallback record is cleared.

At each step t , we estimate the number of actions needed to return from the current position $\mathbf{x}_t$ to $\mathbf{x}^*$. Letting f_{turn} denote the empirical fraction of turn actions executed so far in the episode (App. E.3), we inflate the geodesic translation step

count to account for turning along the path:

$$\hat{n}_{\text{ret}} = \frac{d_{\text{geo}}(\mathbf{x}_t, \mathbf{x}^*)}{s} \cdot \frac{1}{1 - f_{\text{turn}}}. \quad (16)$$

The factor $(1 - f_{\text{turn}})^{-1}$ adapts automatically to the clutter and turning overhead of the current environment. The agent initiates a mandatory fallback return as soon as the remaining step budget reaches

$$B - 1 - t \leq \hat{n}_{\text{ret}} + n_{\text{over}}, \quad (17)$$

$$n_{\text{over}} = \left\lceil \frac{360^\circ}{\delta} \right\rceil + \left\lceil \frac{\min D}{s} \right\rceil + 1 = 15,$$

where the arrival buffer n_{over} reserves 12 steps for a full 360° turn to face the target, 2 forward steps to close the minimum depth standoff $\min D$, and 1 step to issue **STOP**. In addition, the return can be triggered up to $n_{\text{marg}} = 40$ steps earlier if 60 steps have passed without new positive evidence and either $\hat{n}_{\text{ret}} \leq n_{\text{marg}}$ or no active candidate is improving toward p^* . Once triggered, the shared fast-marching planner guides the agent back to $\mathbf{x}^*$ and issues **STOP** within $\min D = 0.5$ m of $\mathbf{x}^*$.

3.5.3 Action Selection and Termination Control

At each step, multiple modules may propose an action. Rather than blending continuous action scores, the controller selects a single active module according to a fixed priority hierarchy ordered by evidence specificity: (1) verified stopping or close-range verification maneuvers, (2) visual pose servoing, (3) candidate-directed path planning, (4) budget-aware fallback return, (5) panoramic sweeps, and (6) frontier exploration. Once a budget-aware fallback return is underway, it can be preempted only by a newly verified immediate stop, preventing unverified detections from diverting the agent when steps are nearly exhausted. Two invariants govern execution: first, only the VLM-verified stopping pipelines (Secs. 3.3 and 3.4), the fallback return upon reaching $\mathbf{x}^*$, and the hard step limit $t = B - 1$ are permitted to emit **STOP** (any waypoint-arrival stop from the path planner or explorer is converted into an in-place turn); second, only the module selected at step t updates its internal step counters and state transitions (Algorithm 1, App. A).

4 Experiments

We evaluate EVINAV in both simulation and real-world environments. Our simulation experiments compare EVINAV with prior learned and zero-shot methods on Instance-Image Navigation (IIN) and Text-Goal Navigation (TN) using the HM3D validation split, and include comprehensive component ablations. Our hardware experiments evaluate end-to-end physical execution across 24 trials in an office and a laboratory under real stereo depth.

4.1 Experimental Setup

Datasets and evaluation metrics.

Simulation experiments are conducted in Habitat (Savva et al. 2019) on the HM3D (Ramakrishnan et al. 2021; Yadav et al. 2023b) validation splits for Instance-Image Navigation (IIN) (Krantz et al. 2022) and Text-Goal Navigation (TN) (Sun et al. 2024), comprising 1,000 episodes per modality across 36 scenes and 795 unique goal instances. An episode succeeds when the agent issues **STOP** within $r = 1.0$ m geodesic distance of any ground-truth goal viewpoint within a budget of $B = 1000$ actions. The agent translates in $s = 0.25$ m steps, rotates in $\delta = 30^\circ$ increments, has a cylindrical base radius of 0.18 m, and captures 640×480 RGB-D frames with a $\phi = 79^\circ$ horizontal field of view (HFOV) at a camera height of 0.88 m; valid depth is clipped to $[\min D, r_{\text{max}}] = [0.5, 5.0]$ m. For IIN, the reference camera intrinsic matrix K_g used in PnP pose estimation is constructed from the reference image’s HFOV. We report Success Rate (SR) and Success weighted by Path Length, $\text{SPL} = \frac{1}{N} \sum_{i=1}^N S_i l_i / \max(p_i, l_i)$, where $S_i \in \{0, 1\}$ indicates episode success, l_i is the shortest-path geodesic distance from the start position to the goal, and p_i is the executed path length; both metrics are reported as percentages.

Models.

All visual and language models are used off the shelf without task-specific training or fine-tuning. Open-vocabulary detection and segmentation use YOLOE (Wang et al. 2025a) (yolo-e-26s-seg.pt) at 640-pixel resolution with a 0.05 confidence threshold; the detector runs only for TN and is disabled for IIN. Local feature matching uses

XFeat* (Potje et al. 2024) with up to 8000 keypoints, and visual similarity scoring uses CLIP (Radford et al. 2021) with the RN50 image encoder. LLM goal parsing and single-view verification for TN (Sec. 3.4), pairwise instance verification for IIN (Sec. 3.3), and the panoramic grid and bounding-box localization queries for IIN search (Sec. 3.5.1) all use `gemin-3.1-flash-lite`. Reasoning settings and comparisons across alternative VLMs are reported in App. H.

Robot platform.

Figure 2 shows our physical robot platform: a Kobuki mobile base equipped with a ZED 2i stereo camera (2.1 mm lens, mounted 0.5 m above the floor, $\approx 101^\circ$ HFOV) and an onboard laptop. We process the native 16:9 stream at 1280×720 resolution to avoid aspect-ratio distortion and calibrate camera-to-base extrinsics prior to each session. The robot executes the same nominal discrete actions as in Habitat (0.25 m forward steps and 30° in-place turns) using closed-loop ZED visual-inertial odometry, pausing 0.15 s after each motion before capturing the next RGB-D frame. Because stereo depth can exhibit missing returns on featureless or very close surfaces, a low-level depth-clearance check monitors forward obstacle clearance and halts motion if an unsafe advance is detected. All mapping, grounding, planning, and control modules run onboard the laptop, while `gemin-3.1-flash-lite` is queried via API. To account for higher-resolution stereo processing and onboard battery limits, hardware trials use an action budget of $B = 500$ steps.

4.2 Quantitative Comparison with Prior Methods

Table 1 compares EViNAV with prior learned and zero-shot methods on the HM3D validation splits. All compared methods operate online in unseen environments without pre-built maps.

Instance-image navigation.

EViNAV achieves 84.1% SR and 35.2% SPL on IIN, outperforming all compared online methods on both metrics. In particular, it surpasses the highest-SR baseline (IEVE InternImage) by 13.9 SR points and the highest-SPL baseline (DEBiT-L + adapters) by 1.7 SPL points, while improving over the training-free UniGoal baseline by 23.9

SR points and 11.5 SPL points. Thus, EViNAV locates substantially more target instances while following shorter paths, without requiring any task-specific policy training. As shown in the ablations (Sec. 4.4), the largest contributor to this gain is recovering and navigating to the reference camera pose rather than approaching the matched object surface directly.

Text-goal navigation.

On TN, EViNAV reaches 55.0% SR and 17.4% SPL, exceeding Context-Nav by 28.8 SR points and UniGoal by 6.0 SPL points. The substantial improvement in SR reflects the difficulty of reliably distinguishing fine-grained text descriptions from distractors: in our ablations (Sec. 4.4), disabling VLM verification and its associated stopping logic drops SR to 11.6%, while removing the budget-aware fallback return reduces SR by 7.8 points.

4.3 Qualitative Results and Comparison

Simulation qualitative comparison.

Figure 3 visualizes 8 episodes across 7 HM3D scenes evaluated under both goal modalities, comparing EViNAV against IEVE Mask R-CNN (Lei et al. 2024) and UniGoal (Yin et al. 2025) on IIN, and against Context-Nav (Jang and Kim 2026) and UniGoal on TN. All methods that query an LLM or VLM use the same `gemin-3.1-flash-lite` backbone (IEVE InternImage and DEBiT are omitted from trajectory visualizations because their public repositories do not include complete runnable IIN configurations; App. F). EViNAV succeeds in all 16 displayed runs, whereas all displayed baselines fail in episodes 250, 847, and 969. A supplementary video showing qualitative results of EViNAV on HM3D validation episodes is provided in Online Resource 1.

Two failure modes dominate the baseline trajectories. The first is exhausting the 1,000-step budget: IEVE runs out of steps in episodes 301, 847, and 855 (in episode 301, its trajectory appears short only because the agent repeatedly attempts to step forward into an obstacle from step 70 onward). The second is premature termination at a same-category or visually similar distractor, or at a location that satisfies only part of



Fig. 2 Hardware platform and experimental environments: (a) a Kobuki mobile base carrying a ZED 2i stereo camera and a laptop; (b) the office area; (c) and (d) the laboratory area.

Table 1 HM3D validation results for Instance-Image Navigation (IIN) and Text-Goal Navigation (TN). All tabulated methods operate online without a pre-built scene map. *Zero-shot* means no training on the goal specification of the corresponding task; *Learned policy* marks any reinforcement- or imitation-trained policy component. For Context-Nav, this is the depth-only PointGoal local controller, pretrained with reinforcement learning (Variable Experience Rollout) on HM3D to execute planner-selected waypoints (Jang and Kim 2026, Sec. 3.3). Values are percentages; higher is better. Bold and underline indicate the first- and second-ranked reported values in each column, respectively; “–” denotes an unreported result. Recent preprints AnyImageNav and V LingNav also report IIN results on this split; we exclude them because their results have not been peer-reviewed and no public implementation is available for verification, and note them here for completeness (Deng et al. 2026b; Wang et al. 2026). GaussNav (Lei et al. 2025), GauScoreMap (Deng et al. 2026a) and MG-Nav (Wang et al. 2025b) report high path efficiency, but the traversal used to create the map is not charged to the evaluated navigation episode. We therefore treat prior-map methods as a different protocol from the online setting studied here.

Method	Zero-shot	Learned policy	IIN		TN	
			SR $\uparrow$	SPL $\uparrow$	SR $\uparrow$	SPL $\uparrow$
<i>Requires task-specific training or fine-tuning for IIN/TN</i>						
RL Baseline (Krantz) (Krantz et al. 2022)	$\times$	$\checkmark$	8.3	3.5	–	–
OVRL-v2, fine-tuned for IIN (Yadav et al. 2023a)	$\times$	$\checkmark$	24.8	11.8	–	–
DEBiT-L + adapters (Bono et al. 2024)	$\times$	$\checkmark$	61.1	<u>33.5</u>	–	–
IEVE Mask R-CNN (Lei et al. 2024)	$\times$	$\checkmark$	68.4	24.1	–	–
IEVE InternImage (Lei et al. 2024)	$\times$	$\checkmark$	<u>70.2</u>	25.2	–	–
<i>No task-specific training or fine-tuning for IIN/TN</i>						
GOAT (Chang et al. 2024)	$\checkmark$	$\times$	37.4	16.1	17.0	8.8
Mod-IIN (Krantz et al. 2023)	$\checkmark$	$\times$	56.1	23.3	–	–
UniGoal (Yin et al. 2025)	$\checkmark$	$\times$	60.2	23.7	20.2	<u>11.4</u>
Context-Nav (Jang and Kim 2026)	$\checkmark$	$\checkmark$	–	–	<u>26.2</u>	9.1
Ours	$\checkmark$	$\times$	84.1	35.2	55.0	17.4

Sources. Baseline values are literature-reported results, with the following exceptions to reporting in the method’s own paper: RL Baseline and OVRL-v2-IIN are from Mod-IIN (Krantz et al. 2023, Table 1); GOAT’s IIN and TN values are from UniGoal (Yin et al. 2025, Table 1). DEBiT values are the reported maxima over the final five checkpoints (Bono et al. 2024, Table 5).

the text context. This occurs for IEVE in episodes 250, 490, and 969; Context-Nav in episodes 250, 855, and 969; UniGoal (IIN) in episode 969; and UniGoal (TN) in episodes 301, 490, and 969. Even when the baselines succeed (e.g., IEVE in episode 188; Context-Nav and UniGoal IIN in episode

301), their trajectories exhibit more backtracking and loop intersections than EViNAV’s.

Figure 4 illustrates how EViNAV revises early hypotheses and confirms targets across views (detailed step-by-step in App. F). In IIN episode 490 (Fig. 4b), a distant view of dining chairs at step 145 initially triggers a positive cue at

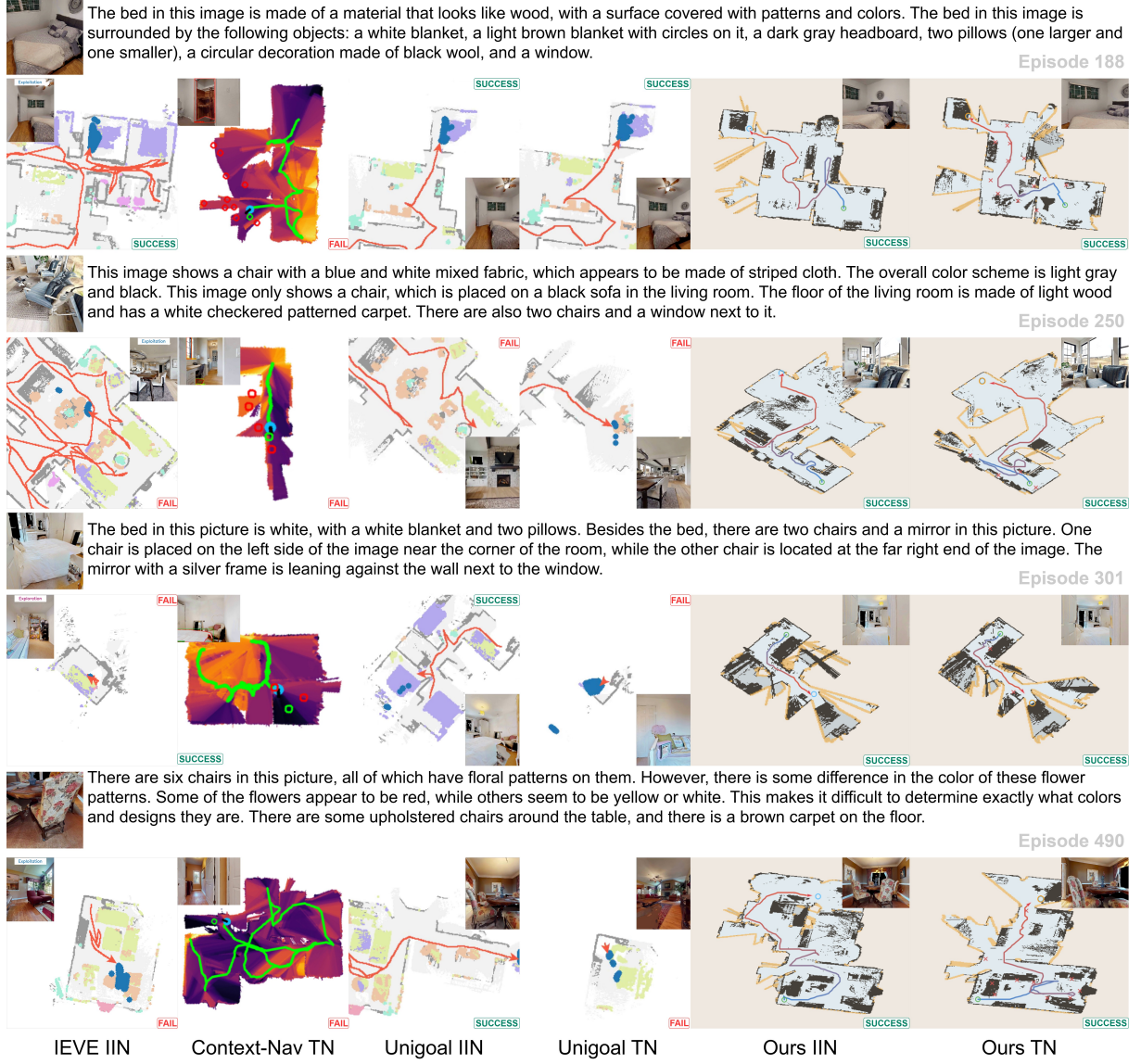
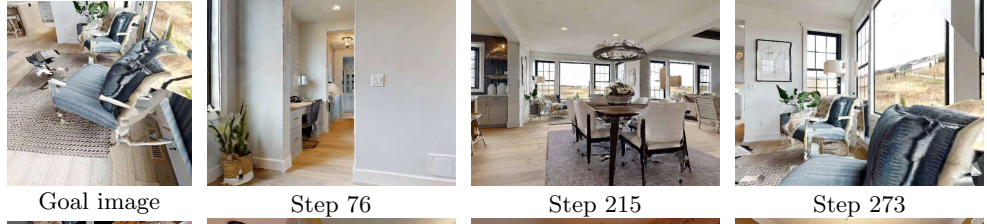


Fig. 3 Simulation qualitative comparison: episodes 188, 250, 301, and 490. Each row gives the reference image for IIN and the description for TN, followed by trajectories, first-person observations, and outcome labels. Columns from left to right are IEVE Mask R-CNN (IIN) (Lei et al. 2024), Context-Nav (TN) (Jang and Kim 2026), UniGoal (IIN and TN) (Yin et al. 2025), and EViNAV (IIN and TN). EViNAV succeeds in both modalities for all four episodes; all displayed baselines fail in episode 250.

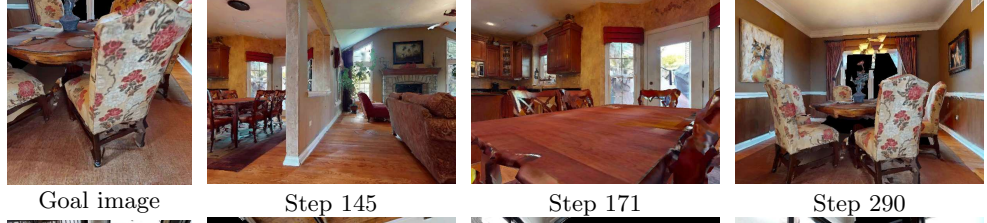


Fig. 3 (continued) Episodes 557, 847, 855, and 969, with the same column order and presentation format as the preceding page. Together, the two pages cover eight episodes across seven scenes; episodes 847 and 855 share a scene but have different targets. EViNAV succeeds in both modalities for all four episodes shown here, whereas all displayed baselines fail in episodes 847 and 969.

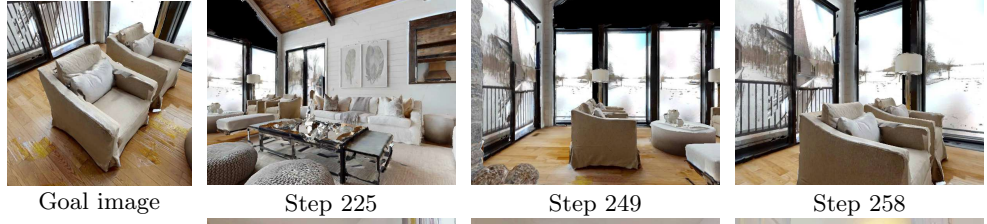
(a) IIN episode 250
Background cue.



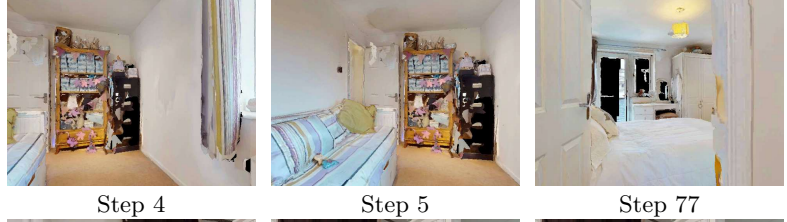
(b) IIN episode 490
Chair disambiguation.



(c) IIN episode 557
Distant target approach.



(d) TN episode 301
White-bed disambiguation.



(e) TN episode 855
Bunk-bed verification.



(f) TN episode 969
Early evidence and heading correction.



(g) TN episode 557
Final approach.



Fig. 4 Original RGB keyframes of EViNAV, with IIN goal images in (a–c). Steps are zero-based pre-action indices; the corresponding actions and verification decisions are described in Sec. 4.3 and App. F.

step 150, but inspecting them closely at step 171 reveals lattice backs rather than the floral upholstery in the goal image; pairwise verification rejects the candidate at step 182, and the agent continues exploring until finding and verifying the floral chairs at step 290. In TN episode 969 (Fig. 4f), the dark shelving seen at step 25 is rejected, while a distant view at step 149 provides initial positive evidence for the target sofa. Upon reaching the sofa at step 177 with the object partially off-center, a single right turn at step 178 centers the target, bringing the multi-view log-odds score above η and confirming the post-maneuver bounding-box check to trigger **STOP**. Similar multi-view behaviors appear in the remaining examples: early false positives in IIN episode 250 and TN episode 301 are rejected upon closer inspection, TN episodes 301 and 855 accumulate evidence across three and two distinct viewpoints before stopping, and in episode 557 both modalities successfully approach and verify a distant corner chair that UniGoal (TN) observes from afar but fails to reach.

Real-world deployment.

Figure 5 summarizes all 24 physical robot trials across three office targets (a tall beige trash bin with a red lid, a brown wooden bookshelf, and a cylindrical black trash can) and three laboratory targets (a black bucket, a wall-mounted fire extinguisher, and a rectangular table). Each target is tested from two starting positions under both TN and IIN. For IIN, goal images are casual smartphone photos taken from arbitrary standing heights and angles; because the phone’s exact intrinsics were not calibrated, we approximate K_g using the ZED 2i HFOV. Overall, EViNAV succeeds in 20 of 24 trials (83.3%): 11/12 in the office and 9/12 in the laboratory, with 10/12 successes in each modality. A supplementary video of real-robot trials is provided in Online Resource 2.

In successful trials, the verification pipeline filters out distractors despite real-world sensor noise and viewpoint mismatch. For instance, in an office TN trial, the robot rejects nearby cardboard boxes and binders before confirming the beige bin via its red lid and surrounding hallway context. In a laboratory IIN trial, XFeat* and PnP recover a valid reference camera pose from the high-angle smartphone photo of the black bucket despite the

robot’s low 0.5 m camera mount, guiding the robot into position for pairwise VLM verification. In an office bookshelf IIN trial, weak initial PnP consensus delays termination even after visual similarity is high; a 30° turn recovers geometric inlier support and allows **STOP**.

The four failure cases (detailed in App. F) highlight remaining hardware and perception challenges: (1) in one office bin IIN trial, a steep oblique passing view fails to yield a PnP pose estimate; (2) in one laboratory IIN trial, a missing stereo-depth patch on a table leg bypasses the forward-clearance check, causing the robot’s bumper to contact the table corner; and (3) in the two laboratory table TN trials, the budget expires. One of these TN failures occurs because YOLOE fails to segment the table even though the VLM recognizes it (triggering a budget-aware fallback return to an earlier view); the other occurs because the agent spends its step budget inspecting competing table candidates.

4.4 Ablation Study

Protocol.

We evaluate 34 text-goal and 27 instance-image ablations under the protocol of Sec. 4.1. Each variant runs on the same 1,000 episodes per modality with identical random seeds so that performance differences can be paired by episode. Every ablation is *capability-preserving*: rather than deleting a module outright (such as removing frontier exploration or target pursuit entirely), each variant replaces the target mechanism with a functional baseline so that the agent can still explore, pursue candidates, and issue **STOP**. Tables 2–4 report paired differences relative to the full system along with 95% percentile-bootstrap confidence intervals (10,000 resamples); boldface indicates intervals that exclude zero. All deltas are computed from unrounded values.

Image goals: reference pose estimation, pursuit, and verification.

IIN performance depends above all on recovering the reference camera pose. Navigating to the 3D centroid of matched surface points instead of the PnP camera pose reduces SR from 84.1 to 64.5 and SPL from 35.2 to 13.5, representing the largest drop in the IIN study (Table 2). Even

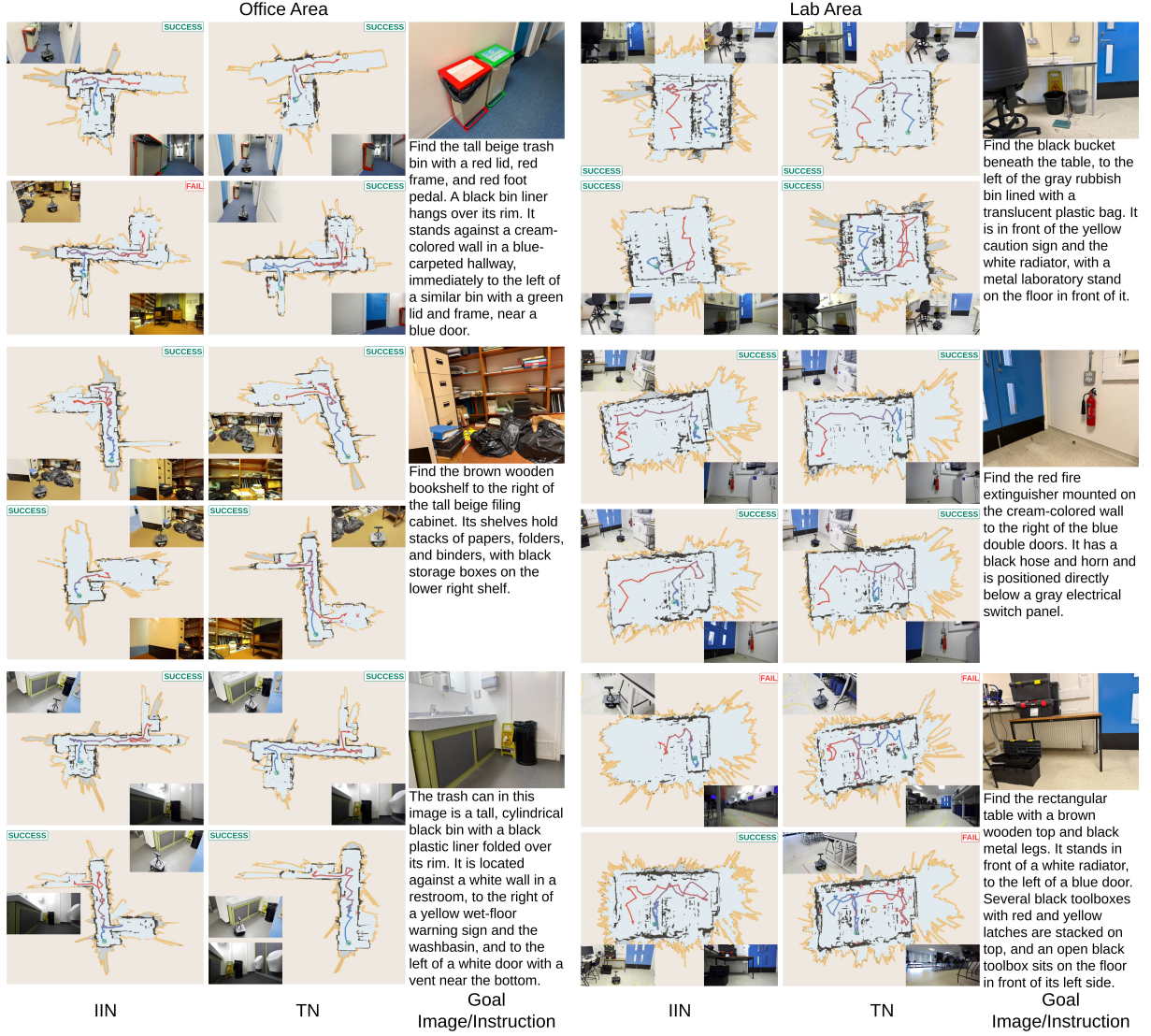


Fig. 5 Real-world evaluation of EviNAV in the office (left) and laboratory (right). Each area contains three selected targets, each evaluated from two starting locations for both IIN and TN, giving 24 trials. For each target, the IIN and TN columns show trajectories with first-person and third-person views; the goal column shows the goal photograph and text instruction. Success and failure labels indicate the observed trial outcomes: 20 successes and four failures.

when the PnP camera pose is estimated, navigating toward the 3D target point rather than the camera position loses 8.5 SR and 14.1 SPL, because the reference camera position guarantees a clear line of sight whereas a surface point on the object may lie against a wall or behind furniture. During pursuit, visual servoing requires a path-planning fallback: relying on visual servoing alone loses 8.2 SR $[-10.4, -6.0]$, whereas using the path planner from the first step reduces SPL by only

1.4 points, showing that visual servoing primarily shortens paths in open space. At termination, centering the camera on the target before verification (-2.9 SR, -6.3 SPL), requiring two-frame PnP confirmation (-2.2 SR), and performing pairwise VLM verification (-2.4 SR) each contribute 2–3 SR points. Multi-frame pose averaging primarily improves path efficiency over single-frame PnP (-2.0 SPL), while triggering verification via the minimum-depth threshold ($\min D \leq 0.5$ m) performs comparably to the $r-s$ radius threshold.

Table 2 Instance-image ablations on HM3D (1,000 validation episodes). The full system obtains 84.1 SR and 35.2 SPL. Δ SR and Δ SPL are paired changes in percentage points with 95% bootstrap intervals; bold: interval excludes zero. [†]Modifies multiple components simultaneously.

Configuration	SR $\uparrow$	Δ SR	SPL $\uparrow$	Δ SPL
Full framework	84.1	–	35.2	–
<i>Occupancy map</i>				
w/o ground-plane height filtering	75.8	– 8.3 [–11.1, –5.6]	29.2	– 6.0 [–7.6, –4.5]
Map from depth rays only (no surfel occupancy)	84.3	+0.2 [–2.1, +2.5]	34.8	–0.4 [–1.6, +0.8]
<i>Goal localization and pursuit</i>				
Matched-surface centroid instead of PnP camera pose	64.5	– 19.6 [–22.7, –16.6]	13.5	– 21.8 [–23.4, –20.2]
3D target point instead of PnP camera pose	75.6	– 8.5 [–10.8, –6.1]	21.1	– 14.1 [–15.6, –12.6]
Visual servoing only (no path-planner fallback)	75.9	– 8.2 [–10.4, –6.0]	32.8	– 2.5 [–3.5, –1.3]
Path planner from first step (no visual servoing)	83.3	–0.8 [–2.5, +0.9]	33.9	– 1.4 [–2.3, –0.4]
<i>Terminal verification and stopping</i>				
Single-frame PnP instead of multi-frame pose averaging	82.9	–1.2 [–3.1, +0.6]	33.2	– 2.0 [–3.0, –1.0]
Unaligned verification view (no camera centering)	81.2	– 2.9 [–4.7, –1.1]	28.9	– 6.3 [–7.4, –5.2]
Single PnP solution instead of two-frame confirmation	81.9	– 2.2 [–4.0, –0.4]	35.5	+0.3 [–0.6, +1.2]
Minimum-depth threshold (min $D \leq 0.5$ m) instead of $r - s$	84.1	0.0 [–1.5, +1.5]	34.9	–0.3 [–1.1, +0.4]
w/o pairwise VLM verification	81.7	– 2.4 [–3.9, –0.9]	34.8	–0.5 [–1.2, +0.3]
w/o all VLM queries [†]	80.7	– 3.4 [–5.8, –1.0]	32.2	– 3.1 [–4.5, –1.6]
w/o budget-aware fallback return	81.5	– 2.6 [–4.3, –0.9]	34.5	–0.7 [–1.5, 0.0]
<i>Search and observation</i>				
w/o panoramic sweeps	79.7	– 4.4 [–6.9, –2.0]	23.9	– 11.3 [–13.1, –9.6]
Every-frontier sweeps ($\zeta = 1$) and no panoramic VLM query [†]	81.1	– 3.0 [–5.5, –0.6]	27.3	– 8.0 [–9.5, –6.4]
Every-frontier sweeps ($\zeta = 1$)	81.6	–2.5 [–4.9, 0.0]	27.1	– 8.1 [–9.6, –6.7]
w/o panoramic VLM grid query	82.3	–1.8 [–4.1, +0.5]	32.1	– 3.1 [–4.4, –1.7]
w/o VLM bounding-box localization	82.5	–1.6 [–3.6, +0.4]	33.4	– 1.8 [–3.0, –0.7]
w/o CLIP similarity buffer	83.0	–1.1 [–3.4, +1.2]	33.1	– 2.2 [–3.6, –0.7]

Disabling all VLM queries removes both pairwise stop verification and panoramic VLM search queries while retaining XFeat/PnP pose estimation, multi-frame pose averaging, geometric pursuit and stopping, and panoramic sweeps. This purely geometric variant still achieves 80.7 SR and 32.2 SPL, with paired drops of –3.4 SR [–5.8, –1.0] and –3.1 SPL [–4.5, –1.6] points. Thus, VLM verification and search queries provide a statistically significant boost in both success and efficiency, while the geometric PnP pipeline accounts for the bulk of IIN performance.

Text goals: multi-view verification and stopping.

In text-goal navigation, VLM verification is essential. Disabling the entire VLM verification and stopping pipeline forces the agent to stop on open-vocabulary detection geometry alone, causing SR to plummet from 55.0 to 11.6 (Table 3). To disentangle the individual mechanisms within this pipeline, we ablate the multi-view decision rule, viewpoint acquisition strategy, and verification crop one at a time while keeping the rest of the system intact.

Comparing *decision rules* reveals a clear trade-off between premature stopping and excessive

delay. Stopping on a single positive VLM query preserves path efficiency (–0.3 SPL) but loses 4.8 SR [–8.0, –1.6] due to false positives on look-alike distractors. Conversely, requiring a fixed vote count of two positive views maintains SR (+1.0, confidence interval includes zero) but loses 3.7 SPL [–5.1, –2.2], because even an unambiguous close-up match must wait for a second viewpoint. Confidence-weighted log-odds accumulation (Eq. 10) resolves this trade-off: a high-confidence, well-framed observation can reach the threshold η on its own, whereas ambiguous observations require corroboration from additional views.

Comparing *viewpoint acquisition strategies* shows that additional evidence must come from genuinely new camera viewpoints. Re-querying the VLM from the same pose and summing the resulting log-odds is the single most damaging modification outside removing the verifier entirely, reducing SR by 15.6 points [–18.9, –12.3]: because repeated queries from the same viewpoint see identical visual evidence, their errors are strongly correlated, allowing a single misinterpretation to cross η and trigger a false stop. Passive viewpoint accumulation enforces spatial separation between views but waits for normal navigation to provide new angles rather than actively stepping around the candidate. It avoids this failure (–1.7 SR,

Table 3 Text-goal ablations on HM3D (1,000 validation episodes). The full system obtains 55.0 SR and 17.4 SPL. Δ SR and Δ SPL are paired changes in percentage points with 95% bootstrap intervals; bold: interval excludes zero. The *Map from depth rays only* and *Highest-scoring detection box instead of candidate projection* rows have identical displayed values but different unrounded SPL statistics; this is not a transcription error. [†]Modifies multiple components simultaneously.

Configuration	SR $\uparrow$	Δ SR	SPL $\uparrow$	Δ SPL
Full framework	55.0	–	17.4	–
<i>Occupancy map and grounding</i>				
w/o ground-plane height filtering	47.8	–7.2 [–10.4, –4.0]	14.3	–3.1 [–4.5, –1.7]
Map from depth rays only (no surfel occupancy)	52.6	–2.4 [–5.5, +0.7]	16.7	–0.7 [–2.0, +0.6]
Geodesic reachability filter with Euclidean ranking	53.8	–1.2 [–3.9, +1.5]	16.8	–0.6 [–1.7, +0.5]
No reachability filter, Euclidean ranking	53.3	–1.7 [–4.8, +1.4]	17.2	–0.3 [–1.6, +1.1]
Depth-only map + reachability filter, Euclidean ranking [†]	53.4	–1.6 [–4.7, +1.5]	16.6	–0.8 [–2.1, +0.4]
Depth-only map + no reachability filter, Euclidean ranking [†]	54.6	–0.4 [–3.5, +2.7]	16.3	–1.1 [–2.4, +0.3]
w/o tracked 3D detections	49.0	–6.0 [–9.4, –2.6]	12.8	–4.6 [–6.1, –3.1]
Top-1 class label instead of query-conditional retrieval	54.2	–0.8 [–3.4, +1.9]	17.3	–0.1 [–1.1, +0.8]
First-sentence query (no LLM goal parser) [†]	51.5	–3.5 [–6.7, –0.4]	15.4	–2.0 [–3.5, –0.6]
Nearest candidate instead of log-odds-weighted ranking	53.7	–1.3 [–4.5, +1.9]	16.5	–0.9 [–2.3, +0.4]
<i>Search</i>				
w/o panoramic sweeps	53.3	–1.7 [–5.0, +1.6]	15.2	–2.2 [–3.8, –0.6]
w/o unreachable-candidate timeout	52.1	–2.9 [–6.0, +0.2]	16.6	–0.9 [–2.1, +0.4]
<i>Verification and multi-view evidence accumulation</i>				
w/o VLM verification and stopping pipeline [†]	11.6	–43.4 [–46.7, –40.2]	7.2	–10.2 [–11.9, –8.7]
Single VLM query instead of multi-view log-odds rule	50.2	–4.8 [–8.0, –1.6]	17.1	–0.3 [–1.7, +1.2]
Vote counting instead of multi-view log-odds rule	56.0	+1.0 [–2.3, +4.3]	13.7	–3.7 [–5.1, –2.2]
Same-pose re-querying instead of active viewpoint changes	39.4	–15.6 [–18.9, –12.3]	15.5	–1.9 [–3.3, –0.5]
Passive instead of active viewpoint changes	53.3	–1.7 [–4.9, +1.5]	15.3	–2.1 [–3.5, –0.7]
Highest-scoring detection box instead of candidate projection	52.6	–2.4 [–5.5, +0.7]	16.7	–0.7 [–2.0, +0.6]
Full image instead of candidate projection	51.1	–3.9 [–7.2, –0.5]	16.0	–1.4 [–2.8, 0.0]
w/o unmapped-detection evaluation	49.4	–5.6 [–8.8, –2.4]	15.5	–2.0 [–3.3, –0.7]
w/o distant candidate verification	53.8	–1.2 [–4.4, +2.0]	15.4	–2.1 [–3.4, –0.8]
<i>Simplified verification baseline, same models[†]</i>				
Simplified baseline, multi-view log-odds rule	51.1	–3.9 [–7.1, –0.7]	15.6	–1.8 [–3.2, –0.5]
Simplified baseline, vote counting	56.5	+1.5 [–1.9, +4.8]	10.3	–7.1 [–8.5, –5.8]
Simplified baseline, single VLM query	43.9	–11.1 [–14.5, –7.6]	16.0	–1.4 [–2.9, +0.1]
<i>Terminal approach and fallback return</i>				
w/o single-step terminal centering	54.4	–0.6 [–3.8, +2.7]	15.2	–2.2 [–3.5, –1.0]
w/o close-range approach maneuver	52.9	–2.1 [–5.2, +0.9]	16.6	–0.8 [–2.2, +0.5]
w/o budget-aware fallback return	47.2	–7.8 [–10.8, –4.7]	16.2	–1.2 [–2.5, +0.1]

Table 4 Exploration-objective ablations on HM3D. The first row gives the full system’s absolute SR and SPL; other rows give paired changes in percentage points with 95% bootstrap intervals. N is the numerator and n the action cost of Eq. (14); λ_{orcl} is oracle-calibrated on the full evaluation set for the additive ablations only and is not used by the full system (App. D). Bold: interval excludes zero. [†]Replaces the frontier objective and removes the CLIP similarity buffer.

Frontier objective	Text-Goal Navigation		Instance-Image Navigation	
	Δ SR	Δ SPL	Δ SR	Δ SPL
Action rate, $N/(n + c_0/s)$ (full)	55.0	17.4	84.1	35.2
Difference, $N - \lambda_{\text{orcl}} n$	–2.7 [–5.8, +0.4]	–1.6 [–2.9, –0.3]	+0.4 [–1.4, +2.2]	+0.1 [–0.8, +0.9]
Difference, $N - 0.5 \lambda_{\text{orcl}} n$	–0.7 [–3.7, +2.4]	–1.3 [–2.6, +0.1]	–0.8 [–2.7, +1.0]	–1.2 [–2.1, –0.3]
Difference, $N - 2 \lambda_{\text{orcl}} n$	–2.1 [–5.2, +1.0]	–0.4 [–1.7, +1.0]	–0.9 [–2.9, +1.1]	+0.3 [–0.7, +1.3]
Gain only, $N (\lambda \rightarrow 0)$	–3.0 [–6.2, +0.2]	–2.3 [–3.6, –0.9]	–2.6 [–4.7, –0.5]	–1.4 [–2.6, –0.3]
Nearest frontier, $-n (\lambda \rightarrow \infty)$	–5.1 [–8.2, –1.8]	–1.4 [–2.9, –0.1]	–0.8 [–2.9, +1.4]	–0.4 [–1.6, +0.9]
Rate, translation-only cost ($\kappa = 0$)	–1.3 [–4.3, +1.7]	–1.3 [–2.6, 0.0]	–0.3 [–2.2, +1.5]	–0.3 [–1.3, +0.7]
Difference, translation-only cost ($\kappa = 0$)	–2.3 [–5.4, +0.8]	–1.1 [–2.5, +0.2]	–0.9 [–3.0, +1.1]	–0.1 [–1.1, +0.9]
Nearest frontier w/o CLIP similarity buffer [†]	–	–	–2.7 [–5.1, –0.3]	–4.3 [–6.0, –2.7]

interval includes zero) but reduces SPL by 2.1 points [–3.5, –0.7]. For the *verification region*, evaluating the full image without a bounding-box crop loses 3.9 SR [–7.2, –0.5], while cropping around the highest-scoring raw 2D detection

instead of the projected 3D candidate loses 2.4 SR (interval includes zero).

We also evaluate a *simplified verification baseline* that removes distant candidate verification, log-odds candidate ranking, single-step terminal

Table 5 Runtime comparison on the same 36 HM3D episodes per modality. Navigation time and steps are summed over all episodes; mean time per step is their ratio. VRAM ranges include local models and the Habitat simulator. LLM/VLM request counts include initialization; “—” denotes no LLM/VLM use. IEVE uses the Mask R-CNN variant, as in the qualitative comparison.

Method	Navigation time (s)	Total steps	Mean time (s/step)	VRAM (MiB)	LLM/VLM requests	Successes
<i>Instance-Image Navigation (IIN)</i>						
IEVE Mask R-CNN	1,109.10	9,776	0.113	6,240–7,348	—	23/36
UniGoal	18,959.49	10,121	1.873	7,999–8,482	8,962	21/36
Ours	2,590.81	12,240	0.212	1,274–1,756	418	34/36
<i>Text-Goal Navigation (TN)</i>						
UniGoal	7,915.26	4,004	1.977	8,027–8,487	4,775	12/36
Context-Nav	14,655.48	12,709	1.153	12,006–13,694	3,994	12/36
Ours	3,063.23	15,031	0.204	1,455–1,863	809	28/36

centering, and the close-range approach maneuver while using passive viewpoint accumulation and raw 2D detection crops. Paired with our multi-view log-odds rule, this simplified baseline loses 3.9 SR and 1.8 SPL; paired with a single-query rule, it loses 11.1 SR; and paired with vote counting, it maintains SR (+1.5) only at the cost of a severe drop in efficiency (10.3 SPL, −7.1). Ablating the auxiliary terminal refinements individually shows that they primarily improve path efficiency: removing distant candidate verification or single-step terminal centering reduces SPL by 2.1 and 2.2 points, respectively, while removing the unreachable-candidate timeout (−2.9 SR) or close-range approach maneuver (−2.1 SR) produces smaller effects whose confidence intervals include zero.

Text goals: open-vocabulary grounding.

Among the grounding mechanisms on TN, maintaining tracked 3D detections before scene-graph consolidation has the largest impact, contributing 6.0 SR [−9.4, −2.6] and 4.6 SPL [−6.1, −3.1]. The SPL effect is the largest among single-component text ablations. Without tracked 3D detections, objects spotted by the detector cannot be pursued until multi-frame depth fusion creates a scene-graph node, wasting actions. Similarly, evaluating unmapped nearby detections directly when no scene-graph candidate is available contributes 5.6 SR [−8.8, −2.4] and 2.0 SPL. Bypassing the LLM goal parser and using only the first sentence of the description as the detector query reduces SR by 3.5 points [−6.7, −0.4] and SPL by 2.0 points [−3.5, −0.6] (even though the VLM verifier still receives the full text instruction). Query-conditional retrieval (−0.8 SR, −0.1 SPL) and

log-odds-weighted candidate ranking (−1.3 SR, −0.9 SPL) have smaller individual effects.

Action-rate frontier exploration.

As shown by Eq. (15), normalizing frontier utility by action cost dynamically sets the exchange rate λ^* between information gain N_i and action expenditure n_i . In Table 4, we compare this parameter-free rate objective against the linear combination $N_i - \lambda n_i$ evaluated at an oracle-calibrated weight λ_{orcl} (the median λ^* observed across all full-system evaluation episodes for each modality), as well as $0.5\lambda_{\text{orcl}}$, $2\lambda_{\text{orcl}}$, pure information gain ($\lambda \rightarrow 0$), and nearest-frontier selection ($\lambda \rightarrow \infty$). Note that λ_{orcl} uses offline oracle statistics from the evaluation set, giving the linear baseline an advantage that our full system does not require.

On TN, the action-rate objective achieves higher point estimates in both SR and SPL than every alternative (Table 4). The two extremes perform poorly: nearest-frontier exploration loses 5.1 SR [−8.2, −1.8], while gain-only exploration loses 2.3 SPL [−3.6, −0.9]. Even with oracle tuning, the linear objective at λ_{orcl} loses 1.6 SPL [−2.9, −0.3], demonstrating that self-normalizing the action rate matches or outperforms a tuned fixed weight without task-specific calibration. Ignoring rotation cost in the action model ($\kappa = 0$) reduces both SR and SPL on TN by 1.3 points. On IIN, varying the frontier coverage objective has a smaller impact because two out of every three contested exploration epochs are driven by the CLIP similarity buffer or VLM-localized regions rather than frontier coverage; however, replacing the entire image search strategy with nearest-frontier exploration and no CLIP buffer reduces

SR by 2.7 points $[-5.1, -0.3]$ and SPL by 4.3 points $[-6.0, -2.7]$.

Occupancy mapping and budget-aware fallback.

Coupling the 3D surfel graph with the 2D occupancy grid provides a unified traversability representation for grounding, frontier extraction, and path planning. While removing surfel occupancy votes or geodesic reachability filtering produces modest individual changes (Tables 2 and 3), ground-plane height filtering is critical in both modalities: disabling it reduces SR by 7.2 points and SPL by 3.1 points on TN, and reduces SR by 8.3 points and SPL by 6.0 points on IIN. Without height filtering, elevated surfaces such as tabletops or beds without visible floor support are misclassified as traversable free space, corrupting both path planning and viewpoint selection.

Similarly, the budget-aware fallback return significantly improves success rate in both modalities, contributing 7.8 SR points on TN $[-10.8, -4.7]$ and 2.6 SR points on IIN $[-4.3, -0.9]$, with negligible change in SPL. Because the fallback triggers only when the episode budget is nearly exhausted, the episodes it rescues have long paths that contribute modestly to SPL, but they reliably convert timeout failures into successful stops at the best previously observed candidate.

Panoramic sweeps and visual search cues.

In IIN, disabling panoramic camera sweeps reduces SR by 4.4 points and SPL by 11.3 points (Table 2). This is the third-largest SPL drop, after localizing the goal by the matched-surface centroid (-21.8 SPL) and navigating to the 3D target point (-14.1 SPL). Therefore, periodic panoramic scanning enables the system to detect the target instance much earlier along the trajectory. Restricting sweeps to frontier arrivals only ($\zeta = 1$, as in TN) loses 8.1 SPL, and disabling the panoramic VLM grid query loses 3.1 SPL. VLM bounding-box localization and the CLIP similarity buffer each contribute 1.8–2.2 SPL points. On TN, panoramic sweeps similarly improve path efficiency (-2.2 SPL when disabled).

Summary across modalities.

Among the 61 ablations, removing the entire VLM verification and stopping pipeline causes the largest TN SR drop (-43.4 SR). Excluding this combined ablation, the largest SR drops in each modality result from replacing reference camera pose estimation with matched-surface centroid localization in IIN (-19.6 SR) and replacing VLM evidence accumulation across distinct camera viewpoints with same-pose re-querying in TN (-15.6 SR). Shared across both modalities, ground-plane height filtering (-8.3 SR in IIN, -7.2 SR in TN) and budget-aware fallback return (-2.6 SR in IIN, -7.8 SR in TN) provide substantial gains in task success, while panoramic sweeps and action-rate exploration consistently improve path efficiency (SPL).

4.5 Runtime Analysis

We profile the wall-clock runtime of EViNAV in Habitat on 36 HM3D validation episodes per modality (selecting the final episode of each of the 36 validation scenes in dataset order). All episodes are run sequentially in a single process on an NVIDIA GeForce RTX 3080 Laptop GPU using the configuration of Sec. 4.1, with `gemini-3.1-flash-lite` serving all LLM and VLM requests. Including both successful and unsuccessful trials, the runtime evaluation covers 12,240 actions on IIN and 15,031 actions on TN. The complete control loop averages 211.67 ms per action on IIN and 203.79 ms per action on TN (71.97 s and 85.09 s per episode, respectively). Remote VLM API calls account for the largest share of wall-clock time (34.8% on IIN and 46.4% on TN); a detailed per-module breakdown is provided in App. F (Table F2).

Comparison with baselines.

Table 5 compares EViNAV against IEVE Mask R-CNN, UniGoal, and Context-Nav on the same 36 episodes per modality on the same RTX 3080 laptop, with UniGoal, Context-Nav, and EViNAV all using `gemini-3.1-flash-lite` for LLM/VLM inference. Timing covers active navigation and excludes simulator resets, model loading, and initial one-time goal parsing. Among the methods that use foundation-model inference, EViNAV achieves the lowest total and per-step runtime,

issues $5\times-21\times$ fewer remote API requests, consumes $4\times-7\times$ less local GPU memory (1.3–1.9 GiB VRAM), and achieves the highest success count in both modalities. The purely local IEVE Mask R-CNN baseline runs faster per step on IIN (0.113 s vs. 0.212 s) but requires $4\times$ more VRAM and succeeds on 23/36 episodes compared to 34/36 for EViNAV.

5 Conclusion

We presented EViNAV, a training-free framework for instance-level navigation that couples a 3D surfel scene graph with a ground-plane-filtered 2D occupancy map to support open-vocabulary grounding, frontier exploration, path planning, and explicit stop verification. Rather than terminating as a side effect of reaching a planned waypoint, EViNAV treats stopping as a dedicated verification problem under a finite step budget: image-goal navigation recovers the 6-DoF reference camera pose via local feature matching and PnP and verifies the target from an aligned viewpoint, while text-goal navigation accumulates candidate-grounded VLM log-odds across spatially distinct viewpoints and actively repositions the robot when initial observations are ambiguous. Both modalities share an action-rate frontier explorer and a budget-aware fallback mechanism that returns to the best previously observed candidate before the step budget expires.

Evaluated on 1,000 HM3D validation episodes per modality, EViNAV achieves 84.1% SR and 35.2% SPL on Instance-Image Navigation and 55.0% SR and 17.4% SPL on Text-Goal Navigation, outperforming recent zero-shot baselines while using under 1.9 GiB of local VRAM and far fewer VLM queries. Our 61 capability-preserving ablations confirm that the primary performance drivers are 6-DoF reference viewpoint recovery for image goals, multi-viewpoint VLM evidence accumulation for text goals, ground-plane height filtering, and budget-aware fallback return. Finally, deployment on a wheeled robot across 24 real-world office and laboratory trials demonstrates that the exact same pipeline transfers directly to noisy stereo depth and casual smartphone reference photos without fine-tuning.

Supplementary information.

Online Resource 1 (ESM.1.mp4): A supplementary video of qualitative results on HM3D validation episodes.

Online Resource 2 (ESM.2.mp4): A supplementary video of real-robot trials.

Declarations

Funding

This research received no external funding.

Conflict of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, literature review, methodology, experimental design, evaluation, results analysis, code development, and manuscript preparation, J.C.; provision of some of the hardware used in the real-robot trials, P.B. All authors reviewed the manuscript and provided suggestions for revisions.

Code availability

The code is available at <https://github.com/sjdjfcu/SGF>.

Appendix A Action Arbitration and Control Loop

This appendix details the per-step control loop summarized in Sec. 3.5.3 (Algorithm 1).

Restricted termination.

Only three mechanisms are permitted to terminate an episode by issuing STOP: (i) the VLM-verified stopping pipelines of Secs. 3.3 and 3.4, (ii) the budget-aware fallback return (Sec. 3.5.2) upon reaching $\mathbf{x}^*$, and (iii) the episode step limit at $t = B - 1$. Any waypoint-arrival STOP produced by the path planner or frontier explorer is converted into an in-place turn so that reaching a planned waypoint never terminates the episode without verification.

Module state updates.

At each step, only the module whose action is actually executed updates its internal step, stall, or viewpoint counters; preempted modules retain their prior state. Similarly, the fast-marching path planner (App. D.2) advances its internal state only when its proposed action is executed.

Priority hierarchy.

Candidate actions are evaluated in decreasing order of goal specificity: (1) verified stopping or close-range verification maneuvers (a^{stop}), (2) visual pose servoing (a^{servo}), (3) path planning toward a grounded candidate (a^{plan}), (4) budget-aware fallback return (a^{ret}), (5) panoramic camera sweeps (a^{pano}), and (6) frontier exploration (a^{exp}). Because a budget-aware fallback return is initiated when the remaining step budget $B - 1 - t$ barely covers the return trip to $\mathbf{x}^*$, an active return trajectory cannot be preempted by newly detected visual servoing or path-planning targets ($a^{\text{servo}}, a^{\text{plan}}$); only an immediate VLM-verified stop (a^{stop}) can override an active fallback return.

Verification and execution.

When a candidate triggers a stop proposal (a^{stop}), it passes through the modality-specific terminal verification pipeline (App. E), which either approves STOP, initiates a close-range approach or viewpoint-change maneuver, or rejects the proposal and falls back to the highest-priority motion

action. Completing a single-step terminal centering turn (App. E.1) or reaching the step limit $t = B - 1$ also issues STOP.

Terminal verification stages.

For text goals, candidate verification proceeds through six stages (detailed in App. E.1):

1. *Triggering verification.* A pursued scene-graph candidate triggers verification upon entering the stopping radius r (or within the extended range $r < d \leq r_{\text{far}}$ once tracked for at least 12 steps). An unmapped nearby 2D detection can also trigger verification directly at its depth-lifted 3D position.
2. *Single-view VLM scoring.* Candidate-grounded VLM verification produces the frame probability p_t and association weight w_t from the evaluated bounding box and projected 3D candidate, yielding the updated multi-view score S_t via Eq. (9).
3. *Immediate stop.* If the stopping criteria of Eq. (10) are satisfied within radius r , STOP is approved (after a single centering turn if the target lies near the edge of the field of view).
4. *Close-range approach.* If verification succeeds in the extended range $r < d \leq r_{\text{far}}$, the positive score is recorded and the agent executes a short close-range approach maneuver (up to 16 steps) to enter radius r , followed by a post-approach confirmation check. Positive verifications obtained beyond r_{far} (e.g., during distant candidate verification) update the candidate’s log-odds score and ranking in Eq. (11) while normal navigation continues.
5. *Multi-view log-odds update.* Incorporating the observation into the candidate’s viewpoint clusters (Algorithm 5) either approves stopping, excludes the candidate and its radius- r neighborhood, or marks the candidate as inconclusive.
6. *Active viewpoint change.* When a candidate remains inconclusive within radius r , the agent navigates to a new viewpoint on a circle of radius d_{node} around the candidate (Sec. 3.4); after two viewpoint changes without reaching a decision, the candidate is retired.

For image goals, entering the verification radius of the active reference-pose cluster triggers an in-place alignment turn toward the 3D target point

Algorithm 1 Single-step control loop. Modules are queried in priority order and the first available action is selected. $\emptyset$ denotes no action proposed, and $\text{nonstop}(\cdot)$ converts any planner waypoint-arrival **STOP** into an in-place turn.

Require: observation $(I_t, D_t, T_{c,t})$, budget B , step t

- 1: **Perceive:** fuse (I_t, D_t) into the surfel scene graph and occupancy grid $\mathcal{O}_t$; update node semantics (for text goals)
 - 2: **Ground:** select a target candidate (text) or match keypoints and estimate the reference camera pose (image); update the CLIP similarity buffer
 - 3: **Propose:** evaluate, in order, a^{stop} , a^{servo} , a^{plan} , a^{ret} , a^{pano} ; run frontier exploration to obtain a^{exp} only if all higher-priority modules return $\emptyset$
 - 4: **if** $a^{\text{stop}} \neq \emptyset$ **then** $a_t \leftarrow \text{STOP}$
 - 5: **else if** $a^{\text{servo}} \neq \emptyset$ and no fallback return is active **then** $a_t \leftarrow a^{\text{servo}}$
 - 6: **else if** $a^{\text{plan}} \neq \emptyset$ and no fallback return is active **then** $a_t \leftarrow \text{nonstop}(a^{\text{plan}})$
 - 7: **else if** $a^{\text{ret}} \neq \emptyset$ **then** $a_t \leftarrow a^{\text{ret}}$
 - 8: **else if** $a^{\text{pano}} \neq \emptyset$ **then** $a_t \leftarrow a^{\text{pano}}$
 - 9: **else if** $a^{\text{exp}} \neq \emptyset$ **then** $a_t \leftarrow \text{nonstop}(a^{\text{exp}})$
 - 10: **end if**
 - 11: **Verify:** if $a_t = \text{STOP}$ and it was not issued by a^{ret} , run modality-specific terminal verification; if approved, keep $a_t = \text{STOP}$; if a closer view or new viewpoint is needed, replace a_t with the corresponding maneuver; if rejected, restore the highest-priority motion action and retire the candidate if exclusion criteria are met
 - 12: **Safety check** (hardware only): if $a_t = \text{MOVE_FORWARD}$ and stereo depth indicates insufficient forward clearance, replace a_t with an in-place turn
 - 13: **Finalize:** if a single-step terminal centering turn just completed or $t = B - 1$, set $a_t \leftarrow \text{STOP}$
 - 14: **Execute** a_t and update the active module’s internal counters
-

followed by pairwise VLM verification (Eq. (3)). Approval issues **STOP**, whereas rejection resumes motion and increments the cluster’s rejection count (App. E.2). Budget-aware fallback return bypasses both verification pipelines and issues **STOP** upon reaching $\mathbf{x}^*$.

Appendix B Scene Graph and Occupancy Grid Details

Confidence-weighted semantic fusion.

YOLOE (Wang et al. 2025a) performs open-vocabulary detection and instance segmentation asynchronously in parallel with geometric surfel fusion. Each 2D detection mask is matched to the visible scene-graph node whose rendered instance mask has the highest intersection-over-union (IoU). Matches with $\text{IoU} < \tau_{\text{IoU}} = 0.15$ are discarded, and detections are processed in descending order of IoU so that at most one detection updates each node per frame. Given an

accepted detection of class c with detector confidence $s \in [0, 1]$, the exponential moving average weight is scaled linearly between $\alpha_{\min} = 0.05$ and $\alpha_{\max} = 0.60$:

$$\begin{aligned} \alpha &= \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) s, \\ \alpha_{\min} &= 0.05, \quad \alpha_{\max} = 0.60. \end{aligned} \tag{B1}$$

The first semantic observation initializes the node’s class distribution $\boldsymbol{\pi}_v = \mathbf{e}_c$ directly, where $\mathbf{e}_c$ is the one-hot vector for query c ; subsequent observations update $\boldsymbol{\pi}_v$ via

$$\begin{aligned} \boldsymbol{\pi}_v &\leftarrow (1 - \alpha)\boldsymbol{\pi}_v + \alpha\mathbf{e}_c, \\ \ell_v &= \arg \max_c \pi_{v,c}, \quad c_v = \max_c \pi_{v,c}. \end{aligned} \tag{B2}$$

Mask association and node merging.

For text goals, rendered node identity masks are evaluated at 160×120 resolution (quarter resolution of the 640×480 input), and YOLOE segmentation masks are downsampled via nearest-neighbor interpolation to compute mask IoU (Algorithm 2). To prevent a single noisy frame

Table B1 Implementation hyperparameters. Task and sensor parameters are given in Sec. 4.1. Slash-separated values correspond to the order listed in the description.

Symbol	Quantity	Value
<i>Shared representation and path planning</i>		
ρ	Occupancy-grid cell size	0.05 m
—	Maximum search radius for snapping to free space	2.0 m
m	Planner obstacle clearance margin	0.05 m
λ_γ	Obstacle clearance penalty weight in Eq. (D11)	9.0
λ_c	Blocked-step penalty weight in Eq. (D11)	12.0
<i>Frontier exploration and visual search priors</i>		
κ	Rotation cost weight, Eq. (13)	1.0
w_G	Semantic goal-score weight, text / image	150 / 0
s_k	Repeatedly failed frontier penalty weight, Eq. (D6)	4.0
—	Minimum CLIP similarity for buffer admission	0.70
R_m	CLIP similarity buffer spatial radius	6.0 m
ϕ_m	CLIP similarity buffer angular cone	110°
—	Geometric-coverage to CLIP-prior epoch ratio	1:2
ζ	Panoramic-sweep distance scale, text / image, Eq. (D10)	1.0 / 0.2
<i>Text semantics and grounding</i>		
τ_{IoU}	Detection-to-node mask IoU threshold	0.15
n_v	Minimum surfel count for an admissible target node	≥ 64
q_v	Minimum query probability for an admissible target node, Eq. (5)	≥ 0.2
—	Minimum confidence / observations for tracked 3D detections	0.2 / 2
—	Unreachable-candidate timeout	60 actions
<i>Instance-image grounding</i>		
f_i	Minimum fine-matcher confidence for an admitted correspondence	≥ 0.25
—	Reference-image foreground crop (lower vertical fraction)	0.70
—	Minimum inliers for 3D surface localization	≥ 14
$S^*, \bar{f}^*$	Minimum fine-confidence sum and mean for 3D surface localization	12.0 / 0.35
—	Minimum confident matches to attempt PnP	≥ 24
—	PnP RANSAC reprojection threshold	10 px
—	Minimum PnP inliers / inlier ratio for an accepted camera pose	≥ 14 / ≥ 0.25
—	Accepted PnP camera planar distance range for visual servoing	[0.25, 4.0] m
—	Maximum PnP camera height difference from current camera	≤ 1.0 m
<i>Terminal verification and fallback return</i>		
—	Minimum interval between text-verification queries	8 actions
$c_{\text{probe}}^{\text{det}}$	Minimum confidence for unmapped-detection evaluation	≥ 0.2
—	Close-range terminal centering distance threshold	0.75 r
ϵ_{seq}	Log-odds decision threshold parameter, Eq. (8)	0.02
—	Viewpoint stagnation limit / max viewpoint clusters per candidate	8 / 8
—	Maximum active viewpoint changes per candidate	2
—	Maximum close-range approach steps per episode	16
n_{marg}	Fallback return early-activation margin	40 actions
—	Fallback return stagnation window / rival score margin	60 steps / 0.02

from merging distinct physical objects, geometric node merges require two nodes to co-occur within the same 2D instance mask in more than three frames.

Occupancy grid classification.

Each cell of the 2D grid $\mathcal{O}_t$ maintains cumulative counts of free-space ray traversals (ν_f), obstacle hits (ν_o), and total observations (ν_{obs}). Cells are classified into three primary states:

$$\begin{aligned}
 \text{free} &\iff \nu_f \geq 1 \wedge \nu_o > \nu_o, \\
 \text{occupied} &\iff \nu_o \geq 2 \wedge \nu_o \geq \nu_f, \\
 \text{unknown} &\iff \nu_{\text{obs}} = 0.
 \end{aligned} \tag{B3}$$

Because occupied classification requires at least two obstacle votes whereas free space requires only one ray traversal, isolated single-pixel depth noise (which adds $\nu_o = 1, \nu_f = 0$) leaves a cell

observed ($\nu_{\text{obs}} > 0$) but neither free nor occupied; such cells are treated as untraversable by the path planner without generating false frontier boundaries. Conversely, each stable 3D surfel within the obstacle height band contributes two occupied votes, registering persistent surfaces as occupied immediately.

Ground-plane estimation and free-space snapping.

The floor elevation $\hat{F}$ (Sec. 3.2) is initialized at the start of the episode from the nominal camera height above the base and refined online as the mode of a 1 cm vertical height histogram of depth points near the ground plane. At each frame, depth pixels subsampled on a 4×4 stride are raycast to update $(\nu_f, \nu_o, \nu_{\text{obs}})$, followed by projecting all stable surfels within the height band $[h_{\min}, h_{\max}]$ (Algorithm 2). To project any 3D

candidate onto the traversable map, an eight-connected breadth-first search locates the nearest free cell within 2.0 m (40 grid cells) and verifies that its geodesic distance d_{geo} from the agent’s current cell is finite.

Algorithm 2 Per-frame semantic fusion and occupancy grid update.

Require: depth D_t , detections $\mathcal{D}_t$, visible node masks $\{M_v\}$, surfels $\mathcal{S}_t$

- 1: **if** text-goal modality **then**
- 2: Downsample masks in $\mathcal{D}_t$ to 160×120 and compute $\text{IoU}(d, v)$ against each visible node mask M_v
- 3: Sort pairs (d, v) with $\text{IoU}(d, v) > \tau_{\text{IoU}}$ in descending order of IoU
- 4: **for** each detection d in sorted order **do**
- 5: Let v^* be the highest-overlap node for d
- 6: **if** v^* has not yet been updated in frame t **then**
- 7: Update π_{v^*} with (c_d, s_d) via Eqs. (B1)–(B2)
- 8: **end if**
- 9: **end for**
- 10: **for** each node pair (u, v) with mask co-occurrence count > 3 **do**
- 11: Merge u into v and combine their semantic distributions π_v
- 12: **end for**
- 13: **end if**
- 14: **for** each pixel $\mathbf{u}$ on a 4×4 stride with valid depth $D_t(\mathbf{u})$ **do**
- 15: Increment ν_f and ν_{obs} along the 2D ray from the camera to endpoint $\mathbf{p}$
- 16: **if** endpoint height $p_y \in [h_{\min}, h_{\max}]$ **then**
- 17: $\nu_o(\mathbf{c_p}) \leftarrow \nu_o(\mathbf{c_p}) + 1$; $\nu_{\text{obs}}(\mathbf{c_p}) \leftarrow \nu_{\text{obs}}(\mathbf{c_p}) + 1$
- 18: **else**
- 19: $\nu_{\text{obs}}(\mathbf{c_p}) \leftarrow \nu_{\text{obs}}(\mathbf{c_p}) + 1$ $\triangleright$ floor or ceiling hit: mark observed only
- 20: **end if**
- 21: **end for**
- 22: **for** each stable surfel $s \in \mathcal{S}_t$ with height in $[h_{\min}, h_{\max}]$ **do**
- 23: $\nu_o(\mathbf{c_s}) \leftarrow \nu_o(\mathbf{c_s}) + 2$; $\nu_{\text{obs}}(\mathbf{c_s}) \leftarrow \nu_{\text{obs}}(\mathbf{c_s}) + 1$
- 24: **end for**

Appendix C Goal-Grounding Details

Text-query normalization and candidate selection.

The language instruction is parsed once at the start of the episode into target and context queries (App. G). Before matching detector labels against parsed queries, strings are normalized by lowercasing, stripping punctuation, collapsing whitespace, deduplicating words, and singularizing nouns. Admissible scene-graph nodes and tracked 3D detections are projected onto the nearest free cell on $\mathcal{O}_t$, filtered by geodesic reachability, and ranked jointly according to the evidence-weighted geodesic cost of Eq. (11).

Tracked 3D detections.

To pursue targets before they accumulate enough surfels to form a scene-graph node ($n_v \geq 64$), any 2D detection matching a target query in $\mathcal{Q}_{\text{tgt}}$ with confidence ≥ 0.2 is lifted to 3D using the depth map and camera pose. To reject background depth outliers near mask boundaries, we compute the component-wise median of the lifted mask points and average the points lying within radius r of that median. The resulting 3D centroid is merged into the nearest tracked detection within planar distance r , or initializes a new track. Once observed in at least two frames, a tracked detection becomes an active navigation candidate unless a scene-graph candidate already lies within distance r .

3D surface localization for image goals.

To localize the target surface from XFeat* matches, correspondences below the minimum confidence threshold are discarded and filtered geometrically using 2D homography RANSAC (with a fundamental-matrix RANSAC fallback when fewer than 4 homography inliers are found). Because reference photographs typically frame the target object in the center and lower portion of the image while the upper region often shows background walls or ceilings, we run RANSAC on both the lower 70% of the reference image $\mathbf{I}_g$ and the full image, preferring the lower-region fit whenever it yields at least 14 inliers.

Each current-frame inlier $\mathbf{u}_i^t = (u_i, v_i)^\top$ with valid depth $D_t(\mathbf{u}_i^t)$ is back-projected into the

world frame:

$$\begin{bmatrix} \mathbf{x}_i^e \\ 1 \end{bmatrix} = \hat{T}_{c,t}^e \begin{bmatrix} D_t(\mathbf{u}_i^t) K^{-1} \tilde{\mathbf{u}}_i^t \\ 1 \end{bmatrix}, \quad (\text{C4})$$

where $\tilde{\mathbf{u}}_i^t = (u_i, v_i, 1)^\top$ denotes homogeneous pixel coordinates, K is the camera intrinsic matrix, and $\hat{T}_{c,t}^e$ is the camera-to-world pose. The resulting 3D points are clustered on the horizontal (x, z) plane using DBSCAN, and the cluster C_t^* with the highest total confidence weight yields the surface centroid

$$\mathbf{c}_t = \frac{\sum_{i \in C_t^*} s_i \mathbf{x}_i^e}{\sum_{i \in C_t^*} s_i}, \quad (\text{C5})$$

where s_i is the match confidence weight. A cluster is accepted as a valid surface localization when $|C_t^*| \geq 14$, the sum of fine match scores is at least $S^* = 12.0$, and the mean fine score is at least $\bar{f}^* = 0.35$. When PnP camera pose recovery does not yet have sufficient inliers, $\mathbf{c}_t$ guides closer inspection (App. D.1) and provides a fallback candidate for budget-aware return (Sec. 3.5.2).

Reference camera pose filtering.

To recover the 6-DoF reference camera pose via Eq. (2), we run EPnP-RANSAC (100 iterations, 0.99 confidence, 10-pixel reprojection threshold) whenever at least 24 matches have fine confidence $f_i \geq 0.25$ and valid depth. The focal length of the reference camera is set to $f_x = f_y = W_g / (2 \tan(\phi_g/2))$ from its horizontal field of view ϕ_g . A PnP solution is accepted if it has at least 14 inliers, an inlier ratio of at least 0.25, a camera center $\mathbf{C} = -\mathbf{R}_g^\top \mathbf{t}_g$ within 4.0 m horizontally of the agent, and a camera elevation within 1.0 m of the agent’s camera height. If the horizontal distance to $\mathbf{C}$ is at least 0.25 m, the pose is used as a visual servoing target; if it is closer than 0.25 m, it is still fused into the multi-frame pose cluster without triggering translational servoing.

Descriptor matching and pose clustering.

In XFeat* (Potje et al. 2024), coarse match similarity ς_i is the zero-truncated inner product of normalized patch descriptors, while the fine matcher predicts sub-pixel offsets on an 8×8 grid whose peak softmax probability defines f_i . Sub-pixel refinement is applied to the reference-image keypoint $\mathbf{u}_i^g$ while keeping the current-frame keypoint $\mathbf{u}_i^t$ at integer pixel coordinates for direct depth

lookup in Eq. (C4). Algorithm 3 summarizes the two-branch geometric grounding procedure. Within each multi-frame pose cluster, member camera poses are weighted by their PnP inlier counts $|I|$, and the cluster medoid is the member minimizing the inlier-weighted horizontal distance to all other members in the cluster.

Appendix D Exploration and Planning Details

Frontier extraction, revisit penalty, and exploration epochs.

Frontier candidates are extracted directly from the 2D occupancy grid $\mathcal{O}_t$. Free cells adjacent to unknown cells are grouped into eight-connected frontier components, and each component is represented by a frontier waypoint $\mathbf{g}_i$ obtained by snapping its centroid to the nearest free cell. A frontier component is retained if it contains at least 20 cells, lies at least 1.2 m horizontally from the agent, has a finite geodesic path distance C_i , and is at least 1.0 m away from any location where navigation has previously failed three or more times. Letting $\mathcal{W}_i$ denote the local grid window around waypoint $\mathbf{g}_i$, $\mathcal{C}_t^{\text{vis}}$ the set of grid cells visited along the trajectory so far, and d_k the horizontal distance from $\mathbf{g}_i$ to the k -th previously unreachable goal, the revisit penalty is

$$R_i = \frac{|\mathcal{C}_t^{\text{vis}} \cap \mathcal{W}_i|}{|\mathcal{W}_i|} + \sum_{k: d_k < 1.0 \text{ m}} s_k \left(2 - \frac{d_k}{1.0 \text{ m}} \right), \quad (\text{D6})$$

$$s_k \in \{1, 4\},$$

where the penalty weight s_k increases from 1 to 4 once three navigation failures have occurred near location k . If no frontier component satisfies the admission criteria, exploration falls back to the nearest unvisited free cell. Each selected frontier waypoint is pursued as an exploration epoch until the agent reaches the waypoint, stalls due to an obstacle or lack of forward progress, switches to pursuing a localized target candidate, or observes a drop in visual similarity below the admission threshold during visually guided exploration. When an epoch ends due to a stall

Algorithm 3 Per-frame instance-image grounding and reference-pose clustering.

Require: correspondences $\mathcal{M} = \{(\mathbf{u}_i^g, \mathbf{u}_i^t, \varsigma_i, f_i)\}$, depth D_t , pose $\widehat{T}_{c,t}^e$

- 1: Back-project all matches with valid depth $D_t(\mathbf{u}_i^t)$ to 3D world points $\mathbf{x}_i^e$ via Eq. (C4)
- 2: **3D surface localization branch:**
- 3: **for** candidate set $\mathcal{M}_{\text{cand}}$ in {lower 70% of $\mathbf{I}_g$, full $\mathbf{I}_g$ } **do**
- 4: Fit 2D homography H via RANSAC (or fundamental matrix F if < 4 homography inliers)
- 5: **if** lower-70% fit has ≥ 14 inliers **or** full-frame fallback is reached **then**
- 6: Cluster inlier 3D points on the horizontal (x, z) plane via DBSCAN
- 7: Select highest-weight cluster C_t^* and compute centroid $\mathbf{c}_t$ via Eq. (C5)
- 8: **if** $|C_t^*| \geq 14$ **and** $\sum_{i \in C_t^*} f_i \geq S^*$ **and** $\bar{f}_{C_t^*} \geq \bar{f}^*$ **then**
- 9: Accept localized surface centroid $\mathbf{c}_t$
- 10: **end if**
- 11: **break**
- 12: **end if**
- 13: **end for**
- 14: **Reference camera pose branch:**
- 15: $\mathcal{M}_{\text{pnp}} \leftarrow \{i \in \mathcal{M} : f_i \geq 0.25 \wedge D_t(\mathbf{u}_i^t) \text{ valid}\}$
- 16: **if** $|\mathcal{M}_{\text{pnp}}| \geq 24$ **then**
- 17: Solve EPnP-RANSAC (Eq. (2)) on $\mathcal{M}_{\text{pnp}}$ to obtain inliers $\mathcal{I}$ and camera center $\mathbf{C} = -\mathbf{R}_g^\top \mathbf{t}_g$
- 18: Let $d_{xz} = \sqrt{(C_x - x_{t,x})^2 + (C_z - x_{t,z})^2}$ and $h_{c,t}$ be the current camera height
- 19: **if** $|\mathcal{I}| \geq 14$ **and** $|\mathcal{I}|/|\mathcal{M}_{\text{pnp}}| \geq 0.25$ **and** $d_{xz} \leq 4.0 \text{ m}$ **and** $|C_y - h_{c,t}| \leq 1.0 \text{ m}$ **then**
- 20: Merge $(\mathbf{C}, \mathbf{f}, |\mathcal{I}|)$ into the nearest pose cluster within radius r
- 21: **if** $d_{xz} \geq 0.25 \text{ m}$ **then**
- 22: Set cluster pose as active visual servoing target
- 23: **end if**
- 24: **end if**
- 25: **end if**

or blocked path, the waypoint is added to the unreachable-goal set.

Action-cost normalization and oracle calibration.

In the action-normalized frontier utility of Eqs. (13)–(14), the translation term C_i/s and rotation term $|\text{wrap}(\beta_i - \theta_t)|/\delta$ express path distance and initial heading alignment in continuous action-step units. The denominator offset c_0/s prevents near-zero travel costs from inflating the ratio; with $c_0 = r_{\text{max}} = 5.0 \text{ m}$ and $s = 0.25 \text{ m}$, this baseline offset equals 20 steps and is held fixed across all ablations (including $\kappa = 0$). When agent heading is unavailable, omitting the rotation term reduces Eq. (14) to a distance-normalized rate $U_i = (w_I I_i + w_G G_i - w_R R_i)/(C_i + c_0)$, which differs only by the constant positive factor s and therefore preserves the frontier ranking. No separate additive distance penalty $-w_C C_i$ is needed because travel cost already enters the denominator.

For the additive frontier-utility ablations in Sec. 4.4, we calibrate the fixed action-cost weight λ_{orcl} using the optimal Dinkelbach ratio $\lambda^* = \max_j U_j$ of Eq. (15) recorded across all frontier decisions of the full system over all 1,000 validation episodes per modality. Setting λ_{orcl} to the empirical median yields 155.8 for TN and 157.1 for IIN. This oracle calibration is used solely by the additive baseline variants in Table 4; the full system evaluates the ratio objective of Eq. (14) directly online without any pre-calibrated exchange rate.

D.1 Image-Guided Search and Panoramic Observation

Pose-tagged CLIP similarity buffer and frontier arbitration.

During instance-image navigation, the cosine similarity between CLIP RN50 embeddings of the reference image and the current view can provide a useful directional cue well before local feature matching can establish 3D correspondences. Such cues may arise, for example, when the agent looks down a hallway toward the target room. To retain these cues without letting noisy single-frame scores dominate exploration, we maintain a spatial buffer of at most 80 pose-tagged directional observations. A frame with CLIP cosine similarity $\text{sim} > 0.70$ is assigned a normalized quality $q = \min(1, \max(0, (\text{sim} - 0.70)/0.30))$, which decays

over time as $2^{-\Delta t/H}$ with half-life

$$H = \max(80, \lceil 2R_m/s \rceil + n_{\text{turn}}) = 80 \text{ steps},$$

matching the step count required for a round trip across the $R_m = 6.0$ m support radius plus one full rotation ($n_{\text{turn}} = \lceil 360^\circ/\delta \rceil = 12$). A new observation within $\max(2\rho, s/2) = 0.125$ m in position and $\delta/2$ in heading of an existing buffer entry merges with that entry, updating its quality to the maximum of q and the decayed quality (and updating the stored pose only when q exceeds the decayed value) so that repeated views from the same location do not accumulate multiple entries.

Each buffer entry j with current decayed quality q_j contributes directional support to frontier waypoint i according to

$$w_{ij} = q_j \left(0.7 \frac{\cos \theta_{ij} - \cos \frac{\phi_m}{2}}{1 - \cos \frac{\phi_m}{2}} + 0.3 \left(1 - \frac{d_{ij}}{R_m} \right) \right), \quad (\text{D7})$$

clipped to $[0, 1]$, where θ_{ij} is the angle between entry j 's recorded viewing direction and the vector from its observation position to $\mathbf{g}_i$, d_{ij} is the horizontal distance to $\mathbf{g}_i$, and ϕ_m and R_m define the angular cone and range. The weights 0.7 and 0.3 place greater emphasis on directional consistency while retaining a preference for spatial proximity. Combining all buffer entries via probabilistic summation gives the total visual support $\bar{w}_i = 1 - \prod_j (1 - w_{ij}) \in [0, 1]$ for frontier i .

To prevent a single uncorroborated similarity spike from overriding geometric exploration, we arbitrate between the coverage utility U_i (Eq. (14)) and a visual frontier score that combines $\bar{w}_i$, frontier size, travel cost, and revisit penalties. Let U_{best} denote the highest coverage utility among current frontiers and $\mathbf{c}_i$ the vector of weighted utility components of candidate i . The visually preferred frontier i is allowed to directly replace the coverage winner only when

$$U_{\text{best}} - U_i \leq \tau_i \|\mathbf{c}_{\text{best}} - \mathbf{c}_i\|_1, \quad (\text{D8})$$

where the trust weight $\tau_i \in [0, 1]$ is the product of the candidate's pooled visual support $\bar{w}_i$, its multi-view corroboration factor $1 - 1/n_{\text{eff}}$ with

effective sample size $n_{\text{eff}} = (\sum_j w_{ij})^2 / \sum_j w_{ij}^2$, and the relative margin $(\bar{w}_{(1)} - \bar{w}_{(2)}) / \bar{w}_{(1)}$ between the top two visually supported frontiers (with $\tau_i = 0$ for all frontiers other than the unique top scorer). Consequently, a single observation ($n_{\text{eff}} = 1$) or tied support yields $\tau_i = 0$ and cannot override the coverage winner directly. When the two rankings disagree and Eq. (D8) does not authorize a direct override, exploration epochs are scheduled between the coverage-selected frontier and the visually selected frontier in a 1:2 ratio (shifting to 1:3 in favor of visual search when panoramic VLM queries confirm the target direction from at least two viewpoints separated by s).

Inspecting low-confidence feature matches.

At long viewing distances, the target object subtends only a small image region, so XFeat* correspondences that pass 2D RANSAC and exceed the minimum inlier count may still fall short of the cumulative confidence threshold S^* required for full 3D surface localization (App. C). Rather than discarding such matches, we measure their relative confidence completeness:

$$c = \min\left(1, \frac{\sum_i f_i}{S^*}\right) \cdot \min\left(1, \frac{\bar{f}}{\bar{f}^*}\right), \quad (\text{D9})$$

$c \equiv 0$ below the inlier floor,

where $S^* = 12.0$ and $\bar{f}^* = 0.35$ are the acceptance thresholds from App. C, so $c = 1$ precisely when full surface localization succeeds. A sub-threshold match ($0 < c < 1$) is held as a tentative 3D point at the confidence-weighted centroid of its dominant depth cluster. If a second frame within one full rotation (12 steps) produces a corroborating match within 1.0 m horizontally, the higher-scoring centroid is snapped to the nearest free cell and pursued as an inspection waypoint. Upon arrival, the tentative waypoint is cleared and the standard geometric localization and PnP gates are re-evaluated from the closer viewpoint.

Panoramic sweeps and VLM-guided localization.

Because a $\phi = 79^\circ$ forward-facing camera can walk past open doorways or side rooms without observing them, the agent periodically performs a 360° panoramic sweep ($\lceil 360^\circ/\delta \rceil = 12$ turns). To keep the cumulative turning overhead bounded as the

explored area grows, the required planar displacement P_t since the previous sweep scales with the cumulative trajectory length L_{t-1} :

$$P_t \geq d_{\text{pano}}(t) = \zeta \max(r_{\text{max}}, L_{t-1}), \quad (\text{D10})$$

and a sweep is triggered only when at least 20 grid cells within range r_{max} remain unknown (an initial sweep is also executed at the start of the episode). We set $\zeta = 1$ (5.0 m minimum interval) for text goals, where open-vocabulary detection operates across full rooms, and $\zeta = 0.2$ (1.0 m minimum interval) for image goals, where local feature matching requires closer views.

For image goals, four cardinal views from each completed sweep are arranged into a numbered 2×2 grid and passed to the VLM alongside the reference image to identify which panel (if any) contains the target instance, returning 0 when no single panel is credible or when the resemblance is purely contextual (App. G). When a panel is selected, its viewing direction is inserted into the CLIP similarity buffer with unit quality ($q = 1$). In addition, the VLM is queried a second time on that panel’s full-resolution image to predict a 2D bounding box around the target instance. The median depth over the central 50% of the predicted box is back-projected to 3D and snapped to the nearest free cell to form an inspection waypoint (provided it lies at least 1.0 m from any previously failed goal). Upon reaching this waypoint, the agent performs one panoramic sweep so that local feature matching and PnP pose estimation can operate at close range. Neither low-confidence feature matches nor VLM sweep queries can trigger a STOP; both serve strictly to bring the agent closer to promising candidates.

D.2 Path Planning and Local Control

The shared navigation planner guides the agent to frontier waypoints, text-goal candidates, localized 3D surfaces, or recovered reference camera poses while maintaining obstacle clearance and recovering from motion stalls.

Clearance-weighted fast marching.

At each planning step, a Euclidean distance transform on the free-space mask of $\mathcal{O}_t$ gives the obstacle clearance $\gamma(\mathbf{x})$ of every traversable cell.

Writing $\gamma^* = r_{\text{agent}} + m = 0.23$ m for the desired clearance ($r_{\text{agent}} = 0.18$ m, margin $m = 0.05$ m), we define the cell traversal cost

$$\Gamma(\mathbf{x}) = 1 + \lambda_\gamma \left[\max(0, \min(1, \frac{\gamma^* - \gamma(\mathbf{x})}{\gamma^*})) \right]^2 + \lambda_c \chi(\mathbf{x}), \quad (\text{D11})$$

where $\lambda_\gamma = 9.0$ penalizes cells closer than γ^* to obstacles and $\lambda_c = 12.0$ weights the local collision count $\chi(\mathbf{x})$ accumulated when a forward step achieves less than one cell of displacement ($\Delta_{\text{block}} = \rho = 0.05$ m). This soft clearance cost steers paths toward corridor centers while still permitting traversal of narrow passages when necessary. We clip $\Gamma(\mathbf{x})$ to $[1, 64]$ and set $\Gamma = 1$ on the 3×3 cell patch immediately under the agent so that a robot close to a wall can smoothly plan away from it. Solving the Eikonal equation $\|\nabla \mathcal{T}(\mathbf{x})\| = \Gamma(\mathbf{x})$ with boundary condition $\mathcal{T}(\mathbf{g}) = 0$ at the goal cell $\mathbf{g}$ yields the cost-to-go potential field $\mathcal{T}(\mathbf{x})$.

Local waypoint selection and stall recovery.

At grid resolution $\rho = 0.05$ m, one nominal forward step spans $n_s = \text{round}(s/\rho) = 5$ cells. Let $\Delta_i = \mathcal{T}(\mathbf{x}_t) - \mathcal{T}(\mathbf{x}_i)$ denote the potential reduction at cell i and o_i its Euclidean grid offset (in cells) from $\mathbf{x}_t$. From the candidate set $\mathcal{P}_t$ of traversable cells within roughly three step radii ($1 \leq o_i \leq 3n_s + 0.25$) that have finite $\mathcal{T}(\mathbf{x}_i)$, positive progress $\Delta_i > 0$, and an unobstructed line of sight on the traversability mask, the planner selects the local waypoint

$$\begin{aligned} \arg \max_{i \in \mathcal{P}_t} & \Delta_i + 0.25 \frac{\Delta_i}{\max(1, o_i)} \\ & - \frac{|\text{wrap}(\beta_i - \theta_t)|}{\pi} \max\left(1, \frac{n_s}{4}\right) \\ & - 0.03 |o_i - n_s|. \end{aligned} \quad (\text{D12})$$

The local waypoint is converted into a discrete action by turning in place whenever the heading error $|\text{wrap}(\beta_{i^*} - \theta_t)|$ exceeds $\delta/2$, and stepping forward otherwise. Closed-loop collision marking, cost relaxation, and stall recovery follow Algorithm 4.

Visual servoing with path-planner fallback.

For image goals, the primary navigation target is the recovered reference camera center $\mathbf{C}$. When

Algorithm 4 Local waypoint selection and motion-stall recovery.

Require: agent pose $(\mathbf{x}_t, \theta_t)$, goal cell $\mathbf{g}$, traversability mask, collision counts χ

- 1: **if** previous action was `MOVE_FORWARD` and displacement $< \Delta_{\text{block}}$ **then**
- 2: Record θ_t in local blocked-heading set; increment $\chi(\mathbf{c})$ for cells immediately ahead
- 3: Mark cells immediately ahead as obstacles in the local traversability mask
- 4: **if** consecutive stalls ≥ 3 **then**
- 5: **return** 90° escape rotation
- 6: **end if**
- 7: **end if**
- 8: Solve Eikonal potential $\mathcal{T}$ to $\mathbf{g}$ on traversability mask with cost Γ (Eq. (D11))
- 9: **if** $\mathcal{T}(\mathbf{x}_t) = \infty$ **or** candidate set $\mathcal{P}_t = \emptyset$ **then**
- 10: Re-solve $\mathcal{T}$ once with local obstacle marks relaxed (retaining soft penalty χ in Γ)
- 11: **end if**
- 12: **if** $\mathcal{P}_t = \emptyset$ **then**
- 13: Increment planning-failure counter; if ≥ 8 , declare $\mathbf{g}$ unreachable
- 14: **return** $\emptyset$
- 15: **end if**
- 16: Select waypoint $i^* \in \mathcal{P}_t$ maximizing Eq. (D12)
- 17: Convert bearing $\beta_{i^*} - \theta_t$ into a turn or `MOVE_FORWARD`
- 18: **if** selected action is `MOVE_FORWARD` along a blocked heading **then**
- 19: **return** lateral bypass maneuver
- 20: **end if**
- 21: **return** selected action

the path ahead is clear, a direct bearing controller rotates the agent until its heading error to $\mathbf{C}$ is within $\max(\delta/2, \arcsin(\min(1, r/\|\mathbf{C} - \mathbf{x}_t\|)))$ and then steps forward. Because line-of-sight visibility of the target object does not guarantee a straight collision-free path to $\mathbf{C}$, the controller switches to the fast-marching path planner whenever a forward servo step is blocked or 12 steps pass without reducing the distance to $\mathbf{C}$ by at least one effective step. The path planner routes the agent around obstacles to the free cell nearest $\mathbf{C}$ and hands control back to visual servoing within $\min D + s = 0.75$ m of the target. If motion stalls a second time, makes no progress for another 12 steps, or has no valid path, the approach ends: the agent runs terminal verification from its current

pose if it is already within radius r of a multi-view confirmed target point, or otherwise suspends the candidate and records its direction in the CLIP similarity buffer.

Retiring unreachable candidates.

In partially observed scenes, a 3D candidate that appears reachable on the 2D grid may still be physically blocked by unmapped obstacles such as glass partitions, low barriers, or stairs. To prevent the agent from spending its action budget on an inaccessible target, we track the minimum distance achieved toward each selected candidate and the step index of the last distance reduction of at least $s \cos(\delta/2) = 0.24$ m (the forward progress of one step at maximum heading misalignment $\delta/2$, which prevents in-place turns from resetting the timer). If 60 steps elapse without progress while the candidate remains beyond $r + s$, the candidate and its radius- r neighborhood are retired from future selection. Within $r + s$, termination and exclusion are governed directly by multi-view verification rather than the progress timer.

Appendix E Terminal-Decision Details

E.1 Text Goals

Verification scheduling and unmapped detections.

During text-goal navigation, VLM verification can be triggered at three distance regimes relative to the stopping radius $r = 1.0$ m and extended approach threshold $r_{\text{far}} = r + s = 1.25$ m:

1. *Close range* ($\leq r$): Any tracked scene-graph candidate within the stopping radius triggers immediate evaluation under Eq. (10).
2. *Extended approach range* ($(r, r_{\text{far}}]$): To prevent transient single-frame detections from triggering premature stops, a scene-graph candidate in $(r, r_{\text{far}}]$ must have been tracked for at least $n_{\text{turn}} = 12$ steps before verification. A negative verdict suppresses re-querying that candidate for 8 steps, whereas a positive verdict initiates a short close-range approach maneuver to bring the agent within radius r before termination. During this maneuver, the agent centers

the target bounding box when off-axis and steps forward while its median depth exceeds $\min D = 0.5$ m.

3. *Distant visible candidates* ($> r_{\text{far}}$): Whenever a candidate node beyond r_{far} is already clearly visible in the current frame ($v_t > 0.5$ and $d_t > 0.5$), the agent evaluates it from its current pose at most once every 8 steps. Because the resulting log-odds score enters the multi-view evidence record directly, a distant refutation can prune a distractor without walking to it, while a positive score boosts its priority in Eq. (11).

In addition, when the currently pursued scene-graph node lies beyond r_{far} , the agent inspects 2D detector outputs (confidence ≥ 0.2) every 4 steps (with an 8-step minimum interval between queries). If a detection’s median depth projects to a 0.5m standoff within r_{far} , it is evaluated as an unmapped nearby detection and keyed by its 3D world position so that its evidence merges with any scene-graph node later formed at that location. If the initial query on an unmapped detection is negative, navigation continues without recording a penalty that might alias a nearby node.

Projected bounding box and VLM score mapping.

To compute a candidate node’s 2D projection (Sec. 3.4), its 3D surfels are voxelized at grid resolution $\rho = 0.05$ m and matched against back-projected depth pixels to determine the projected 2D bounding box, visible voxel fraction v_t , and depth-consistent box fraction d_t . The bounding box passed to the VLM is the highest-confidence 2D detection covering more than half of the projected box; if no such detection exists and $d_t > 0.5$, the projected bounding box itself is used so that a candidate missed by the 2D detector in the current frame can still be evaluated; otherwise, the highest-confidence detection overlapping the projected box is used. Whenever an approval relies on a low-confidence detection or on the projected box alone (which carries zero detector confidence), the agent additionally runs a full-frame scene verification query (Variant B in App. G).

When the VLM returns a single target probability p or a binary judgment $y \in \{0, 1\}$ with confidence c instead of factored appearance and

context probabilities, we map the output to a calibrated positive probability:

$$\Pi(y, p, c) = \begin{cases} \min(1, \max(0, p)), & p \neq \emptyset, \\ \bar{c}, & p=\emptyset, c \neq \emptyset, \\ \min\{\bar{c}, 1 - \bar{c}, \pi_-\}, & p=\emptyset, c=\emptyset, \\ \pi_+, & p=c=\emptyset, y=1, \\ \pi_-, & p=c=\emptyset, y=0, \end{cases} \quad (\text{E13})$$

where $\bar{c} = \min(1, \max(0, c))$ and $(\pi_+, \pi_-) = (0.85, 0.15)$. In the $y = 0$ branch, taking $\min\{\bar{c}, 1 - \bar{c}, \pi_-\}$ guards against models that report $P(\text{target})$ rather than confidence in their negative answer, ensuring that a negative judgment never contributes positive target log-odds.

Proximity distances.

All 3D coordinates are expressed in a common metric world frame with y vertical and (x, z) horizontal, and π_{xz} denotes projection onto the horizontal plane. Let $\mathbf{g}_v$ denote the candidate’s 3D centroid projected to the agent’s height and snapped to the nearest free cell on $\mathcal{O}_t$. Candidate proximity is measured by the eight-connected geodesic path length on the free-space mask:

$$d_{\text{node}} = d_{\text{geo}}(\mathbf{x}_t, \mathbf{g}_v), \quad (\text{E14})$$

with $d_{\text{node}} = \infty$ if no free-space path exists (distances to temporary active-viewpoint waypoints are never substituted for d_{node}). The evaluated 2D bounding box provides a second standpoint $\mathbf{g}_{\text{ev}}$, placed along the box-center ray at its median valid depth minus the standoff $\min D = 0.5$ m (lower-bounded by zero), with horizontal distance $d_{\text{ev}} = \|\pi_{xz}(\mathbf{x}_t - \mathbf{g}_{\text{ev}})\|_2$. The condition $d_{\text{ev}} \leq r$ in Eq. (10) is applied when evaluating unmapped detections or zero-weight projections, and omitted whenever a scene-graph candidate already has positive association weight $w_t > 0$ (in which case proximity is governed by d_{node}).

Active viewpoint selection.

When a verification query leaves a candidate inconclusive ($-\eta < \Lambda < \eta$), repeated queries from the same viewpoint cluster are suppressed. Following a negative or ambiguous view, the agent selects a new observation standpoint at the same radial

distance from the candidate, rotated by $\pm 90^\circ$ or 180° around it, snapped to the nearest free cell, and separated by at least $b = \min D \tan(\phi/2) = 0.41$ m from both the current pose and all previous verification poses. The closest reachable standpoint is pursued within a step budget of $\lceil d_{\text{geo}}/s \rceil + 12$. If the candidate’s bearing misalignment exceeds 15° after a negative view, the agent rotates in place to center the candidate before acquiring the next observation.

Multi-view evidence accumulation and decision rules.

Verification records are keyed by 3D world position (merging records within radius r) and grouped into pose-distinct viewpoint clusters $\{\mathcal{V}_k\}$ as defined in Sec. 3.4. Within each cluster, observations are pooled via Eq. (8) to produce the cumulative log-odds $\Lambda = \sum_{k:W_k>0} E_k$ and current-step score S_t (Eq. (9)); in the same-pose re-query ablation of Table 3, viewpoint clustering is disabled so each query forms its own cluster). Each negative evaluation ($L_t \leq 0$) is categorized as either a missing detection ($w_t = 0$), an *explicit mismatch* ($p_t^{\text{app}} < 0.5$, or $p_t^{\text{ctx}} < 0.5$ when surrounding context is visible), or a *close-up context mismatch* ($p_t^{\text{ctx}} < 0.5$ when the candidate fills the frame so surrounding context is cropped out).

Alongside Λ , we track the number of pose-distinct approvals A and pose-distinct explicit mismatches N via greedy chronological selection: a new approval increments A if it is separated by at least 8 steps and is pose-distinct (translation $\geq b = 0.41$ m, or heading change $\geq \phi/2 = 39.5^\circ$ when both frames contain detections) from all previously counted approvals, while a new explicit mismatch increments N if it differs by at least b in position or $\phi/2$ in heading from all previously counted mismatches. Writing $\mathcal{A}$ for the set of all prior approval steps and $\Delta_{t,t'} = \|\pi_{xz}(\mathbf{x}_t - \mathbf{x}_{t'})\|_2$, Algorithm 5 summarizes the complete multi-view update, candidate exclusion, and acceptance rules.

Alternative verification rules and post-approach checks.

The verification-rule ablations in Table 3 compare our multi-view log-odds decision rule against two baseline decision rules evaluated on the same positive-weight observations ($w_t > 0$):

Algorithm 5 Multi-view log-odds update and text-goal decision rules.

Require: candidate record $\mathcal{H}$, current pose $(\mathbf{x}_t, \theta_t)$, verdict (p_t, w_t, type) , distance d

- 1: Compute weighted log-odds $\ell_t = w_t \text{logit}(p_t)$ (Eq. (7))
- 2: **if** there exists a viewpoint cluster $\mathcal{V}_k \in \mathcal{H}$ whose initial pose is not pose-distinct from $(\mathbf{x}_t, \theta_t)$ **then**
- 3: Append (p_t, w_t, ℓ_t) to the first such cluster $\mathcal{V}_k$
- 4: **else**
- 5: Initialize a new viewpoint cluster $\mathcal{V}_{\text{new}} \leftarrow \{(p_t, w_t, \ell_t)\}$ in $\mathcal{H}$
- 6: **end if**
- 7: Update cluster statistics W_k, R_k, E_k and cumulative score $\Lambda = \sum_{k:W_k>0} E_k$ via Eq. (8)
- 8: Recompute pose-distinct approval count A and explicit-mismatch count N
- 9: Let stagnated $\leftarrow$ true if 8 queries have elapsed without creating a new cluster with $W_k > 0$
- 10: **if** $\Lambda \leq -\eta$ **or** (stagnated $\wedge \Lambda \leq 0$) **or** ($|\{\mathcal{V}_k : W_k > 0\}| \geq 8 \wedge (\Lambda < 0 \vee A = 0)$) **then**
- 11: **return** reject candidate (retire $\mathcal{H}$ and its radius- r neighborhood)
- 12: **end if**
- 13: **if** $\Lambda \geq \eta$ **and** $d \leq r_{\text{far}}$ **and** current verdict is not an explicit mismatch **then**
- 14: Condition 1 (current-view approval):
 $L_t > 0$
- 15: Condition 2 (multi-view corroboration):
 $A \geq N + 2$ (and for unmapped detections, at least two approval positions separated by $\geq r$)
- 16: Condition 3 (close-up translation confirmation):
 $N = 0 \wedge A \geq 1 \wedge$ (current view is close-up) $\wedge$
 $(\forall t' \in \mathcal{A} : \Delta_{t,t'} \geq b)$
- 17: **if** Condition 1 **or** Condition 2 **or** Condition 3 **then**
- 18: **return** accept candidate
- 19: **end if**
- 20: **end if**
- 21: **return** continue navigation

- *Count rule:* Accepts a candidate when the current frame yields a positive verdict within r_{far} and the pose-distinct counts A and N satisfy

$$A \geq N + 2. \quad (\text{E15})$$

Letting n_{cu} and $\tilde{n}_{\text{cu}}$ denote the pose-distinct and total counts of close-up context mismatches, the count rule rejects a candidate when

$$\begin{cases} N \geq 1 \vee n_{\text{cu}} \geq 3 \\ \vee \tilde{n}_{\text{cu}} \geq 6 \vee \text{stagnated}, & A = 0, \\ (N \geq 2 \wedge N > A) & A \geq 1. \\ \vee \tilde{n}_{\text{cu}} \geq 6 + 2A, \end{cases} \quad (\text{E16})$$

- *Single-verdict rule:* Accepts immediately when the current frame yields a positive verdict within r_{far} , and rejects upon receiving a single explicit mismatch or reaching the 8-query stagnation limit.

When an accepted candidate triggers a short close-range approach maneuver rather than an immediate stop, termination is confirmed after the maneuver under Eq. (9) through one of three post-approach checks: (1) if a single centering turn or forward step brings the candidate within $0.75r$ and either surrounding context remains visible (confirmed by the full-frame VLM check) or the target box lies in the lower image half with high detector confidence, the controller issues **STOP** on the next step; (2) for an unmapped nearby detection, the agent executes at most one centering turn and one forward step and confirms termination if the subsequent frame also approves the target and its 2D bounding-box area fraction a_t does not decrease; and (3) if two consecutive close-range views approve the same node and a full-frame VLM query confirms the surrounding scene context, the controller issues **STOP**.

E.2 Image Goals

Verification distance and camera alignment.

For image goals, terminal verification is triggered when the agent approaches the recovered reference camera center $\mathbf{C}$. When the pose cluster is confirmed by at least two views and has no prior negative VLM verdicts, the verification radius is $r - s = 0.75\text{ m}$, providing a one-step margin inside the $r = 1.0\text{ m}$ success radius around $\mathbf{C}$. If the pose estimate comes from a single view, the geodesic distance to $\mathbf{C}$ exceeds r , or the agent approaches from behind a camera pose displaced by more than

r from the localized 3D target point, the verification radius tightens to $\min D = 0.5\text{ m}$ (or the closest reachable free cell within r of the target point if forward motion is blocked). Before querying the pairwise VLM verifier, the agent rotates in place until its camera optical axis aligns within $\delta/2$ of the 3D target point (or aligns with the recovered reference optical axis $\mathbf{f}$ when the target point is closer than $\min D$).

Candidate rejection.

If the pairwise VLM check is negative, the controller records the rejection and continues navigation. A candidate pose cluster is rejected after two negative verdicts, or after a single rejection within $\min D$ of $\mathbf{C}$ (or when an aligned view of a single-view candidate inside the verification radius fails to re-detect the target). A rejected candidate can be reactivated only if new geometric matches are obtained from a viewpoint at least r away from all prior rejection poses.

E.3 Budget-Aware Fallback Return

Candidate priority and return-step estimation.

The fallback record $(\mathbf{x}^*, p^*, t^*)$ prioritizes geometrically grounded standpoints over ungrounded agent poses. Grounded standpoints are obtained by back-projecting a 3D target bounding box for text goals or a localized 3D surface or camera pose for image goals. Ungrounded agent poses are stored only when no 3D-grounded candidate has been observed. Among 3D-grounded candidates for text goals, candidates with multi-view VLM approvals take precedence and are ranked by their verification probability p^* ; for image goals, grounded candidates are ranked by their CLIP similarity score above the 0.70 threshold. Within the same tier, a new observation replaces the stored fallback candidate only when its score strictly exceeds p^* . If the corresponding 3D candidate is later rejected during close-range inspection, the primary fallback record is cleared, though its standpoint is kept in a secondary reserve (ranked by initial verification score) that is used at the hard step limit of Eq. (17) only if no unrefuted fallback candidate exists.

To compute the empirical turn fraction $f_{\text{turn}} = N_{\text{turn}} / \max\{1, N_{\text{turn}} + N_{\text{fwd}}\}$ in Eq. (16), N_{turn} and N_{fwd} count all previously executed turn and

forward steps during target pursuit, frontier exploration, visual servoing, and obstacle avoidance, excluding programmatic 360° panoramic sweeps. Once at least one forward navigation step has been executed ($N_{\text{fwd}} \geq 1$, so $0 \leq f_{\text{turn}} < 1$), Eq. (16) inflates the geodesic translation step estimate by $(1 - f_{\text{turn}})^{-1}$ to trigger the fallback return via Eq. (17).

Appendix F Additional Experimental Details

Baseline availability.

Both the qualitative comparison (Sec. 4.3) and the runtime analysis (Sec. 4.5) evaluate the publicly released IEVE Mask R-CNN variant, which pairs Swin-based goal classification with an RL exploration policy. Because the authors’ public repository does not include a complete runnable configuration and checkpoint set for IEVE Intern-Image (Lei 2024), we report that variant through its published metrics in Table 1. Similarly, the public DEBiT repository releases ImageNav code and weights but omits the task-specific adapters, configurations, and checkpoints required to run IIN (Bono 2025); DEBiT is therefore included via its published scores in Table 1 and omitted from trajectory and runtime comparisons.

Runtime breakdown.

Navigation time in Table F2 covers the complete online control loop, including simulator steps, remote VLM API calls, and diagnostic logging. Episode initialization is measured separately and includes environment setup, model loading, and goal preprocessing (including the initial text-goal LLM parsing request and response handling); post-episode file export and cleanup are excluded. Each per-action contribution is obtained by dividing the total wall-clock time spent in that module by the total number of executed actions for that modality; top-level module rows are mutually exclusive, and indented rows report subcomponents already included in their parent module.

The *Verification and VLM queries* category aggregates all verification routines invoked during an episode, whether or not they result in a **STOP** action. For IIN, this comprises panoramic 2×2 grid queries, bounding-box localization on selected

Table F2 Wall-clock runtime on the 36-scene timing subset. Per-action values include all executed actions within each modality; indented rows are included in the parent totals. “_” denotes an unused component.

Quantity	IIN	TN
<i>Episode means (s/episode)</i>		
Initialization (including LLM goal parsing for TN)	4.28	5.85
Navigation	71.97	85.09
<i>Navigation contributions (ms/action)</i>		
Scene representation update	57.85	75.50
YOLOE wait and fusion	—	0.85
Image-goal processing	65.69	—
CLIP RN50 and similarity	11.41	—
XFeat*	17.29	—
Initial 2-D RANSAC	17.27	—
Depth projection	0.07	—
Depth localization	15.71	—
PnP pose estimation	3.36	—
Planning and control	10.56	24.09
Verification and VLM queries	74.23	102.03
YOLOE prediction	—	3.60
YOLOE lazy initialization	—	2.57
YOLOE class updates	—	0.57
YOLOE result extraction	—	0.02
VLM API round trips	73.66	94.50
Simulator execution	2.10	2.10
Diagnostics and other overhead	1.24	0.08
Total navigation	211.67	203.79

panoramic views, and pairwise instance verification. For TN, it comprises unmapped nearby detection checks, distant candidate verification, and close-range single-view and full-frame VLM verification. The category includes local image cropping and encoding, JSON parsing, and any synchronous YOLOE detection calls invoked during verification. Its API subrow measures complete HTTP round trips (including network transfer, remote inference, and any timed-out requests). Initial LLM goal parsing is counted under episode initialization, whereas asynchronous YOLOE segmentation for mapping and XFeat*/PnP geometric processing are counted under *Scene representation update* and *Image-goal processing*, respectively.

Although the two modalities have similar mean times per action (211.67 ms for IIN vs. 203.79 ms for TN), their internal workloads differ. Image-goal feature matching and geometric pose recovery account for 31.0% of IIN navigation time; TN omits this branch, but spends correspondingly more time on semantic scene-graph updates, path planning, and VLM verification. Total episode duration also reflects trajectory length: TN averages 417.5 actions per episode versus 340.0 for IIN,

explaining why TN requires 85.09 s per episode compared with 71.97 s for IIN despite a slightly lower per-action average.

Remote VLM inference is the single largest contributor to wall-clock time. Verification and VLM queries account for 35.1% of IIN navigation time and 50.1% of TN navigation time, with HTTP round trips alone accounting for 34.8% and 46.4%, respectively. Across the 36-scene subset, the 418 IIN requests and 773 TN requests average 2.16 s and 1.84 s per call. Because VLM queries are invoked selectively on a small fraction of steps, the ≈ 0.2 s per-action average blends fast local control steps with occasional multi-second API calls.

Within IIN, geometric post-processing represents a notable local cost alongside neural feature extraction: CLIP RN50 encoding and XFeat* matching together take 28.70 ms/action, whereas 2D RANSAC filtering (17.27 ms), 3D depth clustering (15.71 ms), and EPnP-RANSAC pose estimation (3.36 ms) together take 36.34 ms/action. For TN, synchronous YOLOE calls during verification average 12.80 ms across 4,225 invocations (3.60 ms/action amortized over 15,031 total actions, plus 2.57 ms/action for on-demand model initialization). By contrast, the YOLOE detector used for per-frame scene-graph updates runs asynchronously in parallel with geometric surfel fusion, leaving only 0.85 ms/action of synchronization wait and semantic fusion overhead on the main thread.

Keyframe narratives.

Figure 4 illustrates how EViNAV revises candidate hypotheses over time. In IIN episode 250 (Fig. 4a), the view at step 76 shows a potted plant and doorway that resemble the background of the reference image, but the target armchair is not visible; when the panoramic grid query selects this view at step 87, the subsequent bounding-box localization query rejects the target’s presence, preventing a false stop. At step 215, the armchairs become visible behind a dining table, and by step 273 the agent reaches the recovered reference viewpoint where both PnP pose estimation and pairwise VLM verification succeed.

In IIN episode 490 (Fig. 4b), a panoramic query at step 150 initially flags a set of dining chairs seen at step 145. Approaching to step 171 reveals lattice chair backs rather than the

floral upholstery in the reference image; the close-range check at step 182 rejects the candidate and resumes exploration. The agent subsequently locates the floral chairs around the target table at step 290, passing both geometric and pairwise VLM verification.

For TN, episode 301 (Fig. 4d) shows an initial false positive on a partially visible sofa at step 4 that is immediately overturned at step 5 once striped bedding and a green pillow come into view, contradicting the white-bed instruction; the agent continues searching and stops at the true white bed at step 77 after accumulating confirmation across three viewpoints. In episode 855 (Fig. 4e), a partial view of a bunk bed at step 117 is followed by closer views revealing the white bed, ladder, drawers, and surrounding curtains, reaching the multi-view threshold η across two viewpoints at step 127. In episode 969 (Fig. 4f), a dark shelving unit is rejected at step 25, whereas the target sofa receives an initial distant approval at step 149; upon reaching the sofa at step 177, an in-place right turn centers the object in the frame at step 178, satisfying the post-maneuver bounding-box check and triggering STOP.

Episode 557 illustrates the benefit of close-range pose and proximity verification when a target is first spotted from across a room. While UniGoal (TN) observes the target chair in a distant corner without approaching it closely enough, EViNAV on IIN (Fig. 4c) spots the chairs beside the window at step 225, approaches to obtain a confirmed PnP pose at step 249, and passes pairwise verification at step 258; on TN (Fig. 4g), the agent steps forward at step 163 to enter the stopping radius and confirms STOP at step 164.

Hardware trial details.

Section 4.3 summarizes the four failed trials out of 24 on the physical robot. In the office IIN bin trial, the robot walks past the target bin at an oblique viewing angle that differs sharply from the handheld reference photo; XFeat* produces too few geometric inliers to recover the camera pose, and the episode ends at the 500-step limit. In the laboratory IIN failure, a missing stereo depth patch on the leg of a table allows the robot to clip the table corner during exploration: although the table support is visible near the left border of the RGB image, the corresponding depth region

contains no valid stereo returns, while valid floor and background returns keep the overall valid-depth coverage (78.9% center, 64.9% left, 96.0% right) above the low-level depth check’s minimum thresholds (8% center, 50% left/right), so the bumper terminates the run on contact.

Both TN hardware failures occur on the rectangular table target and end at the step budget. In the first trial, the target table is visible ahead-left at step 425 and the full-frame VLM check recognizes the scene, but YOLOE fails to detect a bounding box on that table, leaving the agent to track a neighboring table node on the right that remains 1.23 m away (outside the 1.0 m stopping radius of Eq. (10)). Because the full-frame approval stores the step-425 robot pose as an ungrounded fallback candidate, the budget-aware fallback return takes over at step 446 and terminates near that pose at step 483. In the second trial, the robot inspects several candidate tables whose appearance or surrounding context fails VLM verification and exhausts the 500-step budget. Across all hardware runs (Fig. 5), trajectories exhibit more frequent in-place turns and local detours than in Habitat due to low-level stereo depth-clearance interventions that replace forward actions with in-place turns or stop an advance early, including checks triggered by near-field depth voids beside observed structures. Repeated blockage also triggers replanning, producing local detours and repeated turns. Although we calibrate the camera-to-base extrinsics before each trial, uneven or sloped flooring and worn base wheels can cause the camera tripod to shift relative to the base during motion. The resulting extrinsic mismatch can make nominal in-place rotations appear as circular arcs in the estimated trajectories.

Appendix G Language-Model Prompts

A single frozen VLM (gemini-3.1-flash-lite; Sec. 4.1) serves all five language and multimodal roles via an OpenAI-compatible endpoint without automatic retries or constrained decoding; JSON formatting is specified in the prompt and validated client-side, where any malformed response defaults conservatively to a non-approval or null selection. Braces in the listings below are literal

prompt characters except for interpolated *{italic}* placeholders.

(1) Goal parser (text goals).

Called once at the start of each TN episode with text-only input, where $\{limit\} = 4$ and $\{instruction\}$ is the episode’s language instruction.

[system] You extract target detector queries for indoor navigation. Return JSON only.

Read the indoor navigation instruction and extract the main target object for an open-vocabulary detector. Return a JSON object with this schema:

```
{
  "canonical_goal": "lowercase coarse object name",
  "target_phrases": ["short detector query phrases"],
  "context_phrases": ["short detector query phrases for distinctive objects placed around the target"]
}
```

Rules:

- The target is the one object the instruction is ABOUT (its grammatical subject, the object whose own appearance is being described). Objects mentioned only as surroundings, neighbours, or things placed on, under, beside, or behind the target are context, never the target.
- canonical_goal must name the category of that subject object.
- target_phrases must refer to the same target object.
- Keep useful color/material/shape/size words if present.
- If the instruction identifies one specific target object, return exactly 1 target phrase.
- Only return multiple target_phrases when the instruction is genuinely ambiguous and multiple alternative detector queries are necessary.
- When returning multiple target_phrases, order them from most specific to least specific.
- Use the shortest phrase that still uniquely captures the described target.
- Do not include sentences, only short phrases.
- Do not add a generic fallback phrase unless it is necessary because the instruction is ambiguous.
- target_phrases must contain 1 to $\{limit\}$ items.
- context_phrases name physical objects the instruction places near the target (furniture, appliances, fixtures, decor), as short detector queries. Exclude walls, floors, ceilings, doors, rooms, lighting conditions, colours, and any object of the target’s own category. Return an empty list when the instruction describes no such objects; at most $\{limit\}$ items.

Examples:

- Unique target:


```
{
    "canonical_goal": "chair",
    "target_phrases": ["brown wooden chair"],
    "context_phrases": ["dining table"]
  }
```
- Ambiguous target:


```
{
    "canonical_goal": "chair",
    "target_phrases": ["brown wooden chair", "wooden chair"],
    "context_phrases": []
  }
```

Instruction:

```
{instruction}
```

(2) Single-view target verifier (text goals).

Two prompt variants share the same system prompt and produce the verification outputs mapped in Eqs. (6) and (E13). Variant A is the factored candidate-grounded query (Sec. 3.4): the image displays the evaluated 2D candidate bounding box outlined in red without any text label

(so detector class names cannot bias the VLM), and returns separate `appearance_match` (p_t^{app}) and `context_match` (p_t^{ctx}) probabilities, with 0.5 returned for `context_match` whenever surrounding landmarks are either not mentioned in the instruction or not visible in the frame. Variant B evaluates the unannotated full frame as a scene-level check.

[system] You verify whether a highlighted indoor object exactly matches a navigation instruction. Return JSON only.

[A: factored, boxed] You are checking whether the highlighted object is the exact target for an indoor navigation episode.
Full navigation instruction:
{instruction}

The image shows the current observation with the candidate object boxed in red.
Return JSON with this schema:
{`"is_exact_target"`: true/false, `"confidence"`: 0.0-1.0, `"appearance_match"`: 0.0-1.0, `"context_match"`: 0.0-1.0, `"reason"`: "short explanation"}
Rules:
- `appearance_match` is the probability that the boxed object itself matches what the instruction says the target IS (category, material, colour, shape). Judge only the boxed pixels.
- `context_match` is the probability that the visible surroundings match what the instruction says is AROUND the target. Return 0.5 when the instruction states no surroundings, or when the surroundings it describes are not visible in this image.
- Report both numbers independently; do not let one decide the other.
- Confidence measures your confidence in your verdict itself (whether `is_exact_target` is true or false).
- This is exact-target verification, not category-only matching.
- Lighting conditions affect perceived colors; do not treat slight color differences as a mismatch.
- Do not invent unseen details.
- Reason must be one short sentence.

[B: scene level] You are checking whether the exact target object is visible in this image for an indoor navigation episode.
Full navigation instruction:
{instruction}

Is the exact described target visible strongly enough that the agent should STOP now?
Note: the camera may be VERY close to the target, so large objects like beds, sofas, tables, and TV monitors may fill most of the frame or only be partially visible.
Return JSON with this schema:
{`"is_exact_target"`: true/false, `"confidence"`: 0.0-1.0, `"reason"`: "short explanation"}
Rules:
- Confidence measures your confidence in your verdict itself (whether `is_exact_target` is true or false).
- Return true only if the target object itself is clearly identifiable somewhere in the current image.
- If the instruction contains distinctive surrounding context or multiple same-category cues, require visible evidence that this is that specific target, not just any object of the same category.
- If the instruction is sparse and mostly names the category, a clear category match is enough.

- Return false if the object is absent, only faintly implied by nearby context, or only a tiny ambiguous fragment is visible.
- Partial close-up views are acceptable only when the object is still clearly identifiable as the target.
- If the image is ambiguous and the distinctive cues from the instruction are not checkable, prefer false.
- Reason must be one short sentence.

(3) Pairwise instance verifier (image goals).

Implements V_{pair} in Eq. (3), taking the reference goal image (Image A) followed by the current camera frame (Image B) and returning `object_evidence` (ϵ_t) and `context_only` (c_t^{ctx}) alongside the binary decision and confidence.

You compare two images for an instance-image navigation stop. Image A is the goal target image. Image B is the robot's current camera view. Return JSON only:
{`"same_target_visible"`: true/false, `"confidence"`: 0.0-1.0, `"object_evidence"`: "strong|weak|none", `"context_only"`: true/false, `"reason"`: "short"}. Note: confidence measures your confidence in your verdict itself (whether `same_target_visible` is true or false). The target is the main subject of Image A, the object the photograph is centred on, not the room or the other objects around it. Answer true only if Image B clearly shows the same physical object instance as Image A, not just the same category, room, layout, or background context. Set `context_only` true when the match is based mainly on room layout, furniture arrangement, walls, windows, lighting, or style instead of the target object itself. Partial close views are OK only if distinctive object evidence matches. If uncertain, answer false.

(4) Panoramic grid query (image goals).

Issued once per completed 360° sweep on a 2×2 grid ($\{n\}=4$) of numbered cardinal views (Image B) paired with the reference image (Image A). A non-zero selection inserts a unit-quality directional entry into the CLIP similarity buffer and triggers the bounding-box query below, and cannot directly trigger or veto a STOP.

Image A is a reference photograph of one specific object instance. Image B is a $\{n\}$ -panel grid of views captured by a robot turning in place; each panel is numbered in its top-left corner. Return JSON only: {`"panel"`: <0- $\{n\}$ >, `"confidence"`: 0.0-1.0, `"reason"`: "short"}. Give the number of the single panel that shows the same physical object instance as image A, even if it is small, partially visible, or seen from a different angle. Answer 0 when no panel shows that object, when several panels are equally plausible, or when the only resemblance is room layout, furniture style, or background.

(5) Panoramic view bounding-box query (image goals).

Issued on the full-resolution image of the view selected by the panoramic grid query above. A positive detection with a valid bounding box is

back-projected using its median depth to create an inspection waypoint (App. D.1, *Panoramic sweeps and VLM-guided localization*).

Image A is a reference photograph of one specific object instance. Image B is the robot's current camera view. Return JSON only: {"visible": true/false, "confidence": 0.0-1.0, "box_2d": [ymin, xmin, ymax, xmax], "reason": "short"}. Set visible true only if Image B shows the same physical object instance as Image A, not merely the same category, a similar object, or the same room, even if it is small, partially visible, or seen from a different angle. box_2d is the tight bounding box of that object in Image B with coordinates normalized to 0-1000 as [ymin, xmin, ymax, xmax]. If it is not visible, set box_2d to null.

Appendix H Alternative Multimodal Models

We evaluate EViNAV across four alternative VLMs: two open-weight models (Qwen3.5-2B and Qwen3.5-4B) and two commercial models (gemini-3.5-flash-lite and gpt-6-luna). Following the simulation protocol of Sec. 4.1 on all 1,000 validation episodes per modality, we substitute only the VLM used across the five roles in App. G, keeping all prompts, perception modules, navigation parameters, and evaluation criteria identical.

Reasoning configuration.

No API request specifies an explicit reasoning-budget parameter. By default, both gemini-3.5-flash-lite and the primary gemini-3.1-flash-lite model use minimal thinking (Google 2026), whereas gpt-6-luna defaults to medium reasoning (OpenAI 2026). Because Qwen3.5-4B enables extended thinking by default (Qwen Team 2026), we disable thinking on the local serving endpoint to maintain comparable response latency.

As shown in Table H3, IIN performance is highly consistent across all five VLMs (81.9%–84.1% SR), reflecting the strong geometric grounding provided by local feature matching and PnP pose recovery. TN relies more heavily on open-ended language parsing and fine-grained visual-context judgment, and consequently shows wider variation across model scales (from 29.5% SR with Qwen3.5-2B to 50.3% with Qwen3.5-4B and 50.2%–56.5% with the commercial models). Among the alternatives, gpt-6-luna matches the

Table H3 EViNAV with alternative multimodal models under the simulation protocol of Sec. 4.1. SR and SPL are percentages rounded to one decimal place; higher is better. The original model's results are repeated from Table 1 for reference.

Multimodal model	IIN		TN	
	SR ↑	SPL ↑	SR ↑	SPL ↑
<i>Original model (reference)</i>				
gemini-3.1-flash-lite	84.1	35.2	55.0	17.4
<i>Open-weight models</i>				
Qwen3.5-2B	82.7	31.5	29.5	13.8
Qwen3.5-4B	81.9	29.7	50.3	13.0
<i>Recent commercial models</i>				
gemini-3.5-flash-lite	82.3	27.7	50.2	17.9
gpt-6-luna	84.1	34.1	56.5	14.5

highest IIN SR (84.1%) and achieves the highest TN SR (56.5%), while gemini-3.5-flash-lite achieves the highest TN SPL (17.9%).

References

- Bajcsy R, Aloimonos Y, Tsotsos JK (2018) Revisiting active perception. *Autonomous Robots* 42(2):177–196. <https://doi.org/10.1007/s10514-017-9615-3>
- Barsellotti L, Bigazzi R, Cornia M, et al (2024) Personalized instance-based navigation toward user-specific objects in realistic environments. In: *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, pp 11228–11250
- Bono G (2025) DEBIT: Clarification on Instance-ImageNav implementation availability. Author response in GitHub issue #3, URL <https://github.com/naver/debit/issues/3#issuecomment-3196182395>
- Bono G, Antsfeld L, Chidlovskii B, et al (2024) End-to-end (instance)-image goal navigation through correspondence as an emergent phenomenon. In: *International Conference on Learning Representations*
- Cao Y, Zhang J, Yu Z, et al (2025) CogNav: Cognitive process modeling for object goal navigation with LLMs. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp 9550–9560

- Chang M, Gervet T, Khanna M, et al (2024) GOAT: GO to any thing. In: Robotics: Science and Systems (RSS), <https://doi.org/10.15607/RSS.2024.XX.073>
- Chaplot DS, Gandhi D, Gupta S, et al (2020) Learning to explore using Active Neural SLAM. In: International Conference on Learning Representations (ICLR)
- Chen J, Li G, Kumar S, et al (2023) How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers. In: Robotics: Science and Systems (RSS), <https://doi.org/10.15607/RSS.2023.XIX.075>
- Dai A, Papatheodorou S, Funk N, et al (2020) Fast frontier-based information-driven autonomous exploration with an MAV. In: IEEE International Conference on Robotics and Automation (ICRA), pp 9570–9576, <https://doi.org/10.1109/ICRA40945.2020.9196707>
- Deng Y, Yuan S, Bethala GCR, et al (2026a) Hierarchical scoring with 3D Gaussian splatting for instance image-goal navigation. *IEEE Robotics and Automation Letters* 11(10):11307–11313. <https://doi.org/10.1109/LRA.2026.3723288>
- Deng Y, Yuan S, Fang Y (2026b) AnyImageNav: Any-view geometry for precise last-meter image-goal navigation. *arXiv preprint arXiv:2604.05351*
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1:269–271. URL <http://jmvidal.cse.sc.edu/library/dijkstra59a.pdf>
- Dinkelbach W (1967) On nonlinear fractional programming. *Management Science* 13(7):492–498
- Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6):381–395. <https://doi.org/10.1145/358669.358692>
- Gadre SY, Wortsman M, Ilharco G, et al (2023) CoWs on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 23171–23181
- Goetting D, Singh HG, Loquercio A (2025) End-to-end navigation with vision-language models: Transforming spatial reasoning into question-answering. In: Proceedings of the International Conference on Neuro-symbolic Systems, Proceedings of Machine Learning Research, vol 288. PMLR, pp 22–35
- Google (2026) Gemini thinking. URL <https://ai.google.dev/gemini-api/docs/generate-content/thinking>
- Gu Q, Kuwajerwala A, Morin S, et al (2024) ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In: IEEE International Conference on Robotics and Automation (ICRA), pp 5021–5028, <https://doi.org/10.1109/ICRA57147.2024.10610243>
- Guo W, Xu X, Yin H, et al (2025) IGL-Nav: Incremental 3D Gaussian localization for image-goal navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 6808–6817
- Huang C, Mees O, Zeng A, et al (2023) Visual language maps for robot navigation. In: IEEE International Conference on Robotics and Automation (ICRA)
- Huang X, Zhao S, Wang Y, et al (2026) MSG-Nav: Unleashing the power of multi-modal 3D scene graph for zero-shot embodied navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 37154–37163
- Hughes N, Chang Y, Carlone L (2022) Hydra: A real-time spatial perception system for 3D scene graph construction and optimization. In: Robotics: Science and Systems (RSS)
- Jang WS, Kim UH (2026) Context-Nav: Context-driven exploration and viewpoint-aware 3D spatial reasoning for instance navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9626–9636

- Khanna M, Ramrakhya R, Chhablani G, et al (2024) GOAT-Bench: A benchmark for multi-modal lifelong navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 16373–16383, <https://doi.org/10.1109/CVPR52733.2024.01549>
- Krantz J, Lee S, Malik J, et al (2022) Instance-specific image goal navigation: Training embodied agents to find object instances. arXiv preprint [arXiv:2211.15876](https://arxiv.org/abs/2211.15876)
- Krantz J, Gervet T, Yadav K, et al (2023) Navigating to objects specified by images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 10916–10925
- Kuang Y, Lin H, Jiang M (2024) OpenFM-Nav: Towards open-set zero-shot object navigation via vision-language foundation models. In: Findings of the Association for Computational Linguistics: NAACL 2024, pp 338–351, <https://doi.org/10.18653/v1/2024.findings-naacl.24>
- Lei X (2024) IEVE: Official implementation. GitHub repository, URL <https://github.com/XiaohanLei/IEVE/tree/398414ab78cda1b7a77a63f9cae0c30ddd1a82fa>
- Lei X, Wang M, Zhou W, et al (2024) Instance-aware exploration-verification-exploitation for instance ImageGoal navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 16329–16339
- Lei X, Wang M, Zhou W, et al (2025) GaussNav: Gaussian splatting for visual navigation. IEEE Transactions on Pattern Analysis and Machine Intelligence 47(5):4108–4121. <https://doi.org/10.1109/TPAMI.2025.3538496>
- Lepetit V, Moreno-Noguer F, Fua P (2009) EPnP: An accurate $O(n)$ solution to the PnP problem. International Journal of Computer Vision 81(2):155–166. <https://doi.org/10.1007/s11263-008-0152-6>
- Long Y, Cai W, Wang H, et al (2024) InstructNav: Zero-shot system for generic instruction navigation in unexplored environment. In: Conference on Robot Learning (CoRL)
- OpenAI (2026) GPT-6 Luna model. URL <https://developers.openai.com/api/docs/models/gpt-6-luna>
- Patten T, Martens W, Fitch R (2018) Monte Carlo planning for active object classification. Autonomous Robots 42(2):391–421. <https://doi.org/10.1007/s10514-017-9626-0>
- Potje G, Cadar F, Araujo A, et al (2024) XFeat: Accelerated features for lightweight image matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2682–2691
- Qwen Team (2026) Qwen3.5-4B model card. URL <https://huggingface.co/Qwen/Qwen3.5-4B>
- Radford A, Kim JW, Hallacy C, et al (2021) Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research, vol 139. PMLR, pp 8748–8763
- Ramakrishnan SK, Gokaslan A, Wijmans E, et al (2021) Habitat-Matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. In: Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track
- Savva M, Kadian A, Maksymets O, et al (2019) Habitat: A platform for embodied AI research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 9339–9347
- Sethi N, Lodel M, Ferranti L, et al (2026) STEM: Semantic target search and exploration using MAVs in cluttered environments. Autonomous Robots 50:32. <https://doi.org/10.1007/s10514-026-10247-6>
- Sun X, Chen P, Fan J, et al (2023) FGPrompt: Fine-grained goal prompting for image-goal navigation. In: Advances in Neural Information Processing Systems, pp 12054–12073

- Sun X, Liu L, Zhi H, et al (2024) Prioritized semantic learning for zero-shot instance navigation. In: European Conference on Computer Vision, Springer, pp 161–178
- Taioli F, Zorzi E, Franchi G, et al (2025) Collaborative instance object navigation: Leveraging uncertainty-awareness to minimize human-agent dialogues. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 18781–18792
- Tateno K, Tombari F, Navab N (2015) Real-time and scalable incremental segmentation on dense SLAM. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 4465–4472
- Wald A (1945) Sequential tests of statistical hypotheses. The Annals of Mathematical Statistics 16(2):117–186. <https://doi.org/10.1214/aoms/1177731118>
- Wang A, Liu L, Chen H, et al (2025a) YOLOE: Real-time seeing anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 24591–24602
- Wang B, Lin J, Liu C, et al (2025b) MG-Nav: Dual-scale visual navigation via sparse spatial memory. arXiv preprint [arXiv:2511.22609](https://arxiv.org/abs/2511.22609)
- Wang S, Luo Y, Chen X, et al (2026) VLingNav: Embodied navigation with adaptive reasoning and visual-assisted linguistic memory. arXiv preprint [arXiv:2601.08665](https://arxiv.org/abs/2601.08665)
- Wasserman J, Yadav K, Chowdhary G, et al (2022) Last-mile embodied visual navigation. In: Proceedings of the 6th Conference on Robot Learning (CoRL), Proceedings of Machine Learning Research, vol 205. PMLR, pp 666–678
- Werby A, Huang C, Büchner M, et al (2024) Hierarchical open-vocabulary 3D scene graphs for language-grounded robot navigation. In: Robotics: Science and Systems (RSS), <https://doi.org/10.15607/RSS.2024.XX.077>
- Wu SC, Wald J, Tateno K, et al (2021) Scene-GraphFusion: Incremental 3D scene graph prediction from RGB-D sequences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7515–7525
- Yadav K, Majumdar A, Ramrakhya R, et al (2023a) OVRL-V2: A simple state-of-art baseline for ImageNav and ObjectNav. arXiv preprint [arXiv:2303.07798](https://arxiv.org/abs/2303.07798)
- Yadav K, Ramrakhya R, Ramakrishnan SK, et al (2023b) Habitat-matterport 3D semantics dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 4927–4936
- Yamauchi B (1997) A frontier-based approach for autonomous exploration. In: IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp 146–151
- Yin H, Xu X, Wu Z, et al (2024) SG-Nav: Online 3D scene graph prompting for LLM-based zero-shot object navigation. In: Advances in Neural Information Processing Systems (NeurIPS), pp 5285–5307
- Yin H, Xu X, Zhao L, et al (2025) UniGoal: Towards universal zero-shot goal-oriented navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 19057–19066
- Yokoyama N, Ha S, Batra D, et al (2024) VLFM: Vision-language frontier maps for zero-shot semantic navigation. In: IEEE International Conference on Robotics and Automation (ICRA), pp 42–48, <https://doi.org/10.1109/ICRA57147.2024.10610712>
- Yu B, Kasaei H, Cao M (2023) L3MVN: Leveraging large language models for visual target navigation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3554–3560
- Zhang M, Du Y, Wu C, et al (2025) Apex-Nav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. IEEE Robotics and Automation Letters 10(11):11530–11537. <https://doi.org/10.1109/LRA.2025.3606388>

- Zhou B, Zhang Y, Chen X, et al (2021) FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters* 6(2):779–786
- Zhou K, Zheng K, Pryor C, et al (2023) ESC: Exploration with soft commonsense constraints for zero-shot object navigation. In: *International Conference on Machine Learning (ICML)*, pp 42829–42842
- Zhou Z, Hu Y, Zhang L, et al (2025) BeliefMap-Nav: 3D voxel-based belief map for zero-shot object navigation. In: *Advances in Neural Information Processing Systems*, pp 92526–92554
- Ziliotto F, Campari T, Serafini L, et al (2025) TANGO: Training-free embodied AI agents for open-world tasks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 24603–24613

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [ESM1.mp4](#)
- [ESM2.mp4](#)