CONSISTNAV: IMPROVING ACTION CONSISTENCY IN ZERO-SHOT OBJECT NAVIGATION WITH SEMANTIC EXECUTIVE CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot object navigation has advanced rapidly with open-vocabulary detectors, image–text models, and language-guided exploration. However, even after current methods detect a plausible target hypothesis, the agent may still oscillate between exploration and pursuit, or abandon the object near success. We identify this failure mode as an action consistency gap: semantic evidence is repeatedly reinterpreted at each step without persistent commitment across the episode. We introduce ConsistNav, a training-free zero-shot ObjectNav framework built around a semantic executive composed of three coordinated modules: Finite-State Executive Controller stages target pursuit through guarded semantic phases; Persistent Candidate Memory accumulates cross-frame target evidence into stable object hypotheses; and Stability-Aware Action Control suppresses rotational stagnation, ineffective pursuit, and unverified stopping. This design changes neither the detector nor the low-level planner; instead, it controls when semantic evidence should influence navigation and when it should be suppressed or revisited. We conduct extensive experiments on HM3D and MP3D, where ConsistNav achieves state-of-the-art results among compared zero-shot ObjectNav methods and improves SR by 11.4% and SPL by 7.9% over the controlled baseline on MP3D. Ablation studies and real-world deployment experiments further demonstrate the effectiveness and robustness of the proposed executive mechanism.

1 INTRODUCTION

Object-goal navigation requires an embodied agent to find a target-category instance in an unseen environment and stop within the success radius (Batra et al., 2020). In zero-shot ObjectNav (ZSON), open-vocabulary detectors, vision–language models, and language-guided frontier exploration have improved semantic search without task-specific policy training (Radford et al., 2021; Khandelwal et al., 2022; Majumdar et al., 2022; Yu et al., 2023; Zhou et al., 2023; Yin et al., 2024; Yokoyama et al., 2024; Zhang et al., 2025). However, many modular systems still optimize *semantic evidence production* or *frontier scoring* more directly than the conversion of evidence into sustained action (Yamauchi, 1997; Zhou et al., 2023; Yin et al., 2024; Yokoyama et al., 2024; Zhang et al., 2025). When semantic observations are repeatedly reinterpreted against newly scored frontiers, an agent may oscillate between exploration and pursuit, abandon a plausible target near success, or stop from an insufficiently verified viewpoint. We call this the *action consistency gap*: useful evidence is observed, but the induced objective, commitment, and stop decision are not preserved through approach, verification, and termination.

We introduce ConsistNav, a training-free perception–planning–execution framework for ZSON centered on a semantic executive. The executive contains two coordinated components: *Persistent Candidate Memory (PCM)* and the *Finite-State Executive Controller (FSEC)*. PCM accumulates cross-frame target evidence into stable object hypotheses. FSEC stages target pursuit through guarded semantic phases and integrates *Stability-Aware Action Control (SAAC)* to suppress rotational stagnation, ineffective pursuit, and unverified stopping. Together, these components convert open-vocabulary semantic evidence into temporally consistent action while leaving the detector, semantic mapper, and low-level planner unchanged.

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

We evaluate ConsistNav through controlled experiments under the same underlying stack, isolating the effect of executive control from changes in perception or planning capacity. Beyond standard Success Rate (SR) and Success weighted by Path Length (SPL), we analyze diagnostic failure categories. Persistent memory provides the robot with accumulated environmental information and execution experience. The state-transition mechanism uses this information to guide an execution process analogous to questioning and reflection on candidate hypotheses, enabling the robot to reassess its current judgments and revise subsequent actions.

Our contributions are threefold:

- **Action consistency formulation.** We identify the *action consistency gap* in zero-shot ObjectNav. We introduce the Action Consistency Score (ACS) to quantify action consistency in navigation.
- **Corrective execution with persistent memory.** We introduce a training-free executive that couples persistent candidate memory with guarded state transitions. The former provides knowledge and information, while the latter uses this information to support an execution process analogous to questioning and correction, enabling evidence-supported target commitments to persist.
- **Experimental evaluation.** We assess ConsistNav's navigation performance on ObjectNav benchmarks through comparative experiments, staged ablations, and failure analysis. We also conduct real-world robot experiments to evaluate its deployment feasibility.

2 Related Work

2.1 Semantic Frontier Navigation

Zero-shot ObjectNav increasingly relies on pretrained visual–language representations. CLIP-style goal embeddings and open-vocabulary detections enable navigation to unseen categories (Khandelwal et al., 2022; Majumdar et al., 2022), while later systems add language priors, scene graphs, and multimodal commonsense reasoning (Gadre et al., 2023; Yokoyama et al., 2024; Yu et al., 2023; Shah et al., 2023; Zhou et al., 2023; Yin et al., 2024; Rajvanshi et al., 2024; Zhang et al., 2025). Representative semantic frontier methods combine occupancy mapping, frontier scoring, and semantic fusion to decide *where* evidence should guide exploration. ConsistNav studies *when* accumulated evidence should become a commitment and how it should be verified during action.

2.2 Classical Executive Control for ZSON

Our work is also connected to temporal abstraction, behavior trees, and modular semantic mapping, which formalize persistent goals, preconditions, recovery, and separated mapping–planning–action loops (Ghallab et al., 2004; LaValle, 2006; Sutton et al., 1999; Kaelbling et al., 1998; Fox et al., 1997; Colledanchise & Ögren, 2018; Yamauchi, 1997; Chaplot et al., 2020b;a). We do not claim finite-state control as new; instead, our novelty lies in identifying belief-to-action inconsistency in open-vocabulary semantic navigation and defining measurable failure modes. Thus, ConsistNav is not a generic state machine, but an executive layer specialized for failures that emerge when semantic evidence, frontier exploration, and stop decisions interact online.

2.3 Perception-Exploration-Control Structure for ZSON

Recent embodied pipelines combine open-vocabulary perception (Liu et al., 2024; Li et al., 2022; Minderer et al., 2022; Gu et al., 2021; Li et al., 2023; Kirillov et al., 2023; Zhang et al., 2023) with frontier exploration and semantic mapping (Yamauchi, 1997; Fox et al., 1997; Mur-Artal et al., 2015; Campos et al., 2021; McCormac et al., 2017; Rosinol et al., 2020). These components improve perception, but do not govern semantic evidence during execution. Learning-based ObjectNav and related formulations model such couplings under supervision or large-scale training (Wijmans et al., 2019; Chaplot et al., 2020b;a; Ramakrishnan et al., 2022; Ramrakhya et al., 2022; Yadav et al., 2023; Ramrakhya et al., 2023; Sutton et al., 1999; Kaelbling et al., 1998). ConsistNav instead targets the zero-shot setting, enforcing a lightweight reliability contract between perception and planning without retraining detector–planner-specific policies.

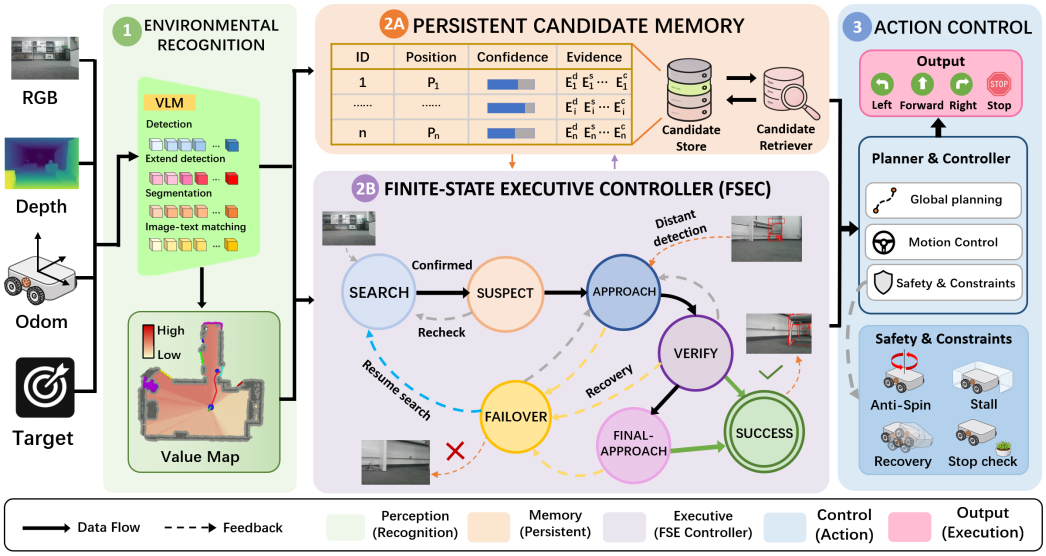


Figure 1: **ConsistNav overview.** Perception (1) supplies semantic evidence to Persistent Candidate Memory (2A) and FSEC (2B). Memory supports candidate assessment, while verification and recovery outcomes feed back into memory. Action control (3) executes navigation and authorized stopping. Figure 2 details the transition guards.

3 METHOD

3.1 PROBLEM FORMULATION: ACTION CONSISTENCY IN OBJECTNAV

In zero-shot ObjectNav, detecting a plausible target does not guarantee sustained pursuit or valid termination. Observation fluctuations may cause an agent to alternate between exploration and pursuit, abandon a supported target, or stop prematurely. We term this mismatch between accumulated semantic evidence and navigation behavior the **action consistency gap**.

We define **action consistency** as sustaining target pursuit while supported by evidence, abandoning the target when clear counterevidence emerges, and stopping when the task completion conditions are met. We evaluate these capabilities using four method-independent event sets: commitment persistence (P), commitment release (R), valid termination (T+), and rejection of invalid termination (T−). Their pass rates are

$$C_k = \frac{1}{|\mathcal{E}_k|} \sum_{e \in \mathcal{E}_k} \psi_e, \qquad k \in \{\mathrm{P}, \mathrm{R}, \mathrm{T}+, \mathrm{T}-\}, \tag{1}$$

where $\mathcal{E}_k$ contains eligible, fully observed events and $\psi_e \in \{0, 1\}$ indicates whether event e passes its predefined criterion. Thus, $|\mathcal{E}_k| = n_k^{\text{pass}} + n_k^{\text{fail}}$; censored and unscorable events are excluded from the denominator and reported separately. For T+, a pass requires a voluntary STOP within the fixed response window that satisfies the official Habitat success condition; timeout- or environment-forced termination cannot pass. Appendix C.1 specifies the event and aggregation rules.

We assign equal weight to commitment persistence, commitment release, and termination, and equal weight to the two types of termination events. For positive component rates, the **Action Consistency Score (ACS)** is defined as

$$\mathrm{ACS} = \exp\left[\frac{1}{3}\left(\log C_{\mathrm{P}} + \log C_{\mathrm{R}} + \log\sqrt{C_{\mathrm{T}+} C_{\mathrm{T}-}}\right)\right]. \tag{2}$$

When all four event sets are nonempty, we set $\mathrm{ACS} = 0$ if any component rate is zero, using the continuous extension described in Appendix C.1.

ACS complements SR and SPL by measuring how consistently semantic evidence guides navigation behavior, including the maintenance, revision, and termination of target commitments.

3.2 OVERVIEW

ConsistNav is a training-free framework that connects accumulated semantic evidence to persistent and revisable target commitments. As shown in Figure 1, it builds on existing perception, mapping, and low-level planning modules through two coupled components: **Persistent Candidate Memory** and **Finite-State Executive Controller (FSEC)**.

At step t, the agent receives a target category g, an RGB-D observation $o_t = (I_t, D_t)$, and a pose estimate $x_t = (p_t, \theta_t)$. Perception and mapping produce spatially grounded semantic evidence e_t and a map $\mathcal{M}_t$. Let r_t summarize the execution feedback available before the current decision, including navigation progress and candidate-specific verification and recovery outcomes. The candidate memory is updated as

$$\mathcal{C}_t = \mathcal{U}(\mathcal{C}_{t-1}, e_t, r_t)\,, \tag{3}$$

where $\mathcal{U}$ associates observations with persistent candidates and updates their evidence and availability. This memory provides a shared basis for maintaining supported hypotheses and reconsidering contradicted candidates.

The executive maintains a control state ξ_t, including the active candidate i_t, the execution phase q_t, and the history summaries required for progress monitoring and recovery. Given the updated memory and a fixed low-level planner $\mathcal{P}$, its state and action are computed as

$$(\xi_t, a_t) = \mathcal{F}(\xi_{t-1}, \mathcal{C}_t, \mathcal{M}_t, x_t, r_t; \mathcal{P})\,. \tag{4}$$

Here, $\mathcal{F}$ manages subgoal selection, verification, recovery, and STOP authorization, with checks for rotational stagnation and ineffective pursuit. The executive selects an action $a_t \in \mathcal{A}$ from {forward, left, right, stop}. At the same time, verification and recovery outcomes update candidate evidence and guide subsequent decisions to maintain or revise target commitments. Sections 3.3 and 3.4 detail the memory updates and executive rules, respectively.

3.3 PERSISTENT CANDIDATE MEMORY

As shown in Figure 1 (2A), Persistent Candidate Memory implements the memory update, associating observations with persistent hypotheses and maintaining the evidence required for commitment management.

Persistent candidates. The memory $\mathcal{C}_t$ maintains persistent candidate records that link observations across time. Each candidate i stores its position μ_i, accumulated target confidence $\hat{c}_i$, negative evidence n_i, and target and non-target observation counts $m_{i,+}$ and $m_{i,-}$. Matched observations update the existing record, with candidate positions refined by exponential smoothing; unmatched target observations initialize new records.

Evidence updates. For candidate i at step t, matched observations update its accumulated target confidence $\hat{c}_i^{(t)}$, while verification failures increase its accumulated negative evidence $n_i^{(t)}$:

$$\begin{aligned} \hat{c}_i^{(t)} &= \min\Big(1, \hat{c}_i^{(t-1)} + \alpha y_{i,t} - 0.12\alpha(1 - y_{i,t})\Big)\,, \\ n_i^{(t)} &= n_i^{(t-1)} + \beta z_{i,t}. \end{aligned} \tag{5}$$

Here, $y_{i,t}$ indicates whether a matched observation supports the target category, and $z_{i,t}$ indicates verification failure. The parameters α and β control confidence updates and verification penalties, respectively. Temporary detection loss alone does not delete a candidate.

Eligibility and priority. The candidate evidence score $s_i^{(t)} \in [0, 1]$ summarizes the accumulated evidence and failure penalties of candidate i at step t:

$$s_i^{(t)} = \mathrm{clip}_{[0,1]}\left(w_c \hat{c}_i^{(t)} + w_r \frac{m_{i,+}^{(t)}}{m_{i,+}^{(t)} + m_{i,-}^{(t)} + \varepsilon} + w_o \min\left(1, \frac{m_{i,+}^{(t)}}{6}\right) + w_\iota \bar{\iota}_i^{(t)} - w_f \phi_i^{(t)} \right). \tag{6}$$

Here, $\bar{\iota}_i$ stores image–text matching (ITM) history, and ϕ_i summarizes failure penalties. Eligible candidates pass confidence and evidence-score thresholds, have more target than non-target observations, and are neither marked failed nor under cooldown. They are ranked by accumulated evidence

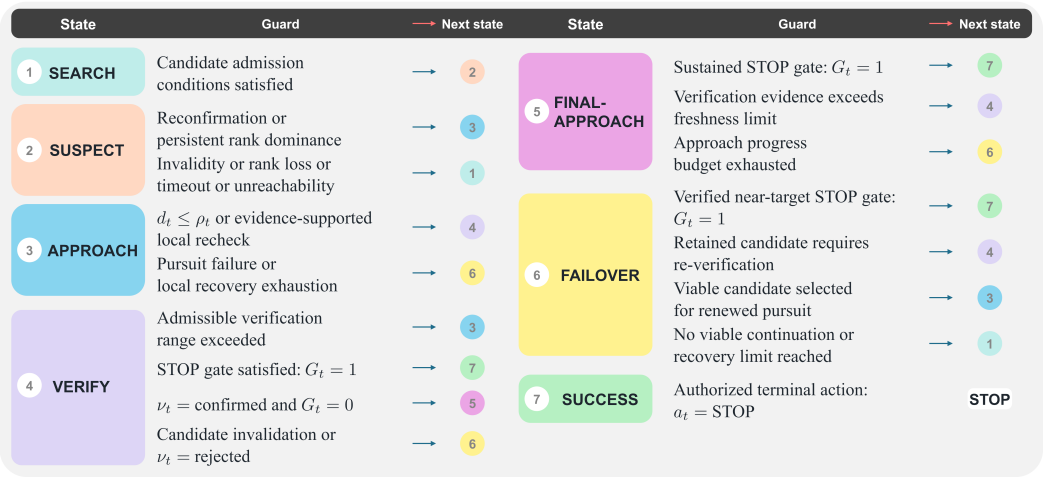


Figure 2: **Guarded state transitions.** Numbers identify states and self-transitions are omitted. ρ_t controls distance-based entry into verification, ν_t denotes verification status (pending, confirmed, or rejected), and G_t denotes the phase-specific STOP gate.

with penalties for staleness, failure history, revisits, and failed regions. The priority function and parameter settings, including the weights $\mathbf{w}$, are provided in the appendix. This ranking supplies candidates to the executive.

3.4 Finite-State Executive Controller

ConsistNav couples persistent candidate memory with a seven-state FSEC that maintains an active candidate i_t and an execution phase q_t. As shown in Figure 2, guarded state transitions coordinate pursuit, verification, recovery, and termination.

Memory-supported reassessment. Selected eligible candidates enter SUSPECT and advance to APPROACH after reconfirmation or persistent rank dominance. Accumulated evidence supports continued pursuit through temporary detection loss, while VERIFY re-examines the candidate hypothesis. Its distance-based entry condition is

$$d_t \le \rho_t, \qquad \rho_t = r_v + \mathbb{I}[k_t^{\text{app}} \ge N_{\text{verify}}]\, r_{\text{extra}}. \tag{7}$$

Here, d_t is the current candidate distance, k_t^{app} counts steps since entering APPROACH, N_{verify} is the expansion threshold, and r_v and r_{extra} are the base radius and expansion. Reaching the threshold enlarges the radius without independently triggering verification. For a valid candidate within range, a bounded scan precedes confirmation or rejection. Its status ν_t is pending, confirmed, or rejected; an unfinished scan alone does not trigger FAILOVER.

Corrective execution and feedback. When a supported candidate encounters ineffective pursuit, FAILOVER can retain its identity while adjusting the subgoal or requesting re-verification. Rejection or unsuccessful recovery updates candidate evidence and failure records, guiding reselection or a return to SEARCH. Confirmed candidates requiring further motion enter FINAL-APPROACH; stale confirmation prompts renewed verification.

Entry into SUCCESS requires $G_t = 1$, where G_t is the phase-specific STOP gate combining semantic, geometric, and temporal checks. Verification and recovery outcomes update memory and guide subsequent decisions. Within FSEC, SAAC uses rotation and progress checks, bounded recovery, and STOP interception outside SUCCESS to mitigate rotational deadlock and motion stalls and resume pursuit after local recovery. Detailed transitions and safeguards appear in Appendix B.

Persistent memory provides accumulated environmental evidence and execution experience, enabling the executive to reassess candidate hypotheses and revise actions in a process analogous to questioning and reflection. This feedback sustains target focus and coordinates candidate assessment with execution.

Table 1: ObjectNav results on HM3Dv2, HM3Dv1, and MP3D. Prior rows are literature-reported contextual baselines; controlled same-stack comparisons are described in the text and ablations.

			HM3Dv2		HM3Dv1		MP3D	
Method	Unsupervised	Zero-shot	SR	SPL	SR	SPL	SR	SPL
PONI (Ramakrishnan et al., 2022)	No	No	–	–	–	–	31.8	12.1
ZSON (Majumdar et al., 2022)	Yes	No	–	–	25.5	12.6	15.3	4.8
OpenFMNav (Kuang et al., 2024)	Yes	Yes	–	–	54.9	24.4	37.2	15.7
InstructNav (Long et al., 2024)	Yes	Yes	58.0	20.9	–	–	–	–
VLFM (Yokoyama et al., 2024)	Yes	Yes	63.6	32.5	52.5	30.4	36.4	17.5
TriHelper (Zhang et al., 2024)	Yes	Yes	–	–	56.5	25.3	–	–
SG-Nav (Yin et al., 2024)	Yes	Yes	49.6	25.5	54.0	24.9	<u>40.2</u>	16.0
ApexNav (Zhang et al., 2025)	Yes	Yes	<u>76.2</u>	<u>38.0</u>	<u>59.6</u>	<u>33.0</u>	39.2	<u>17.8</u>
ConsistNav (ours)	Yes	Yes	**84.2**	**41.2**	**63.2**	**34.8**	**50.6**	**25.7**

4 EXPERIMENTS

We describe the evaluation setup, compare with prior ObjectNav methods, analyze failures and component ablations, evaluate action consistency, and report real-world deployment results.

4.1 BENCHMARKS AND IMPLEMENTATION DETAILS

Datasets. We evaluate under the Habitat ObjectNav protocol (Savva et al., 2019; Batra et al., 2020) on three benchmark settings: MP3D (Chang et al., 2017), HM3Dv1 (Ramakrishnan et al., 2021), and HM3Dv2. HM3Dv1 uses the HM3D-Semantics-v0.1 validation split across 20 scenes and 6 goal categories; HM3Dv2 uses the v0.2 validation split across 36 scenes and the same categories; MP3D uses a validation split across 11 Matterport3D scenes and 21 categories. Together, these benchmarks test transfer across scene scale, reconstruction source, and category distribution.

Evaluation and implementation. We report the standard ObjectNav metrics SR and SPL (Anderson et al., 2018). All variants share the Habitat protocol—a $[0, 5]\,\mathrm{m}$ sensing range, $0.2\,\mathrm{m}$ success radius, and $0.88\,\mathrm{m}$ RGB-D camera height—and the same YOLO-World/GroundingDINO, MobileSAM, and BLIP-2 stack. The internal $r_{\text{stop}} = 0.28\,\mathrm{m}$ condition is a candidate-centered map-space cue within the multi-cue STOP gate; it neither substitutes for nor guarantees benchmark success. Habitat independently validates an executed STOP using its ground-truth distance-to-goal threshold of $0.2\,\mathrm{m}$. Protocol and shared-stack details appear in Appendix Table C.1, with hardware specifications in Table C.2.

4.2 COMPARISON WITH STATE-OF-THE-ART

Table 1 compares ConsistNav with representative ObjectNav methods on HM3Dv2, HM3Dv1, and MP3D. Prior rows are literature-reported contextual baselines; the primary evidence is a controlled same-stack comparison against the Non-executive baseline, which keeps the detector, mapper, planner, simulator protocol, and action budget fixed. Under this setting, ConsistNav improves SR/SPL by $+8.0\%/+3.2\%$ on HM3Dv2, $+3.6\%/+1.8\%$ on HM3Dv1, and $+11.4\%/+7.9\%$ on MP3D. Among the methods listed in Table 1, it obtains the strongest reported results on all three benchmarks, indicating that persistent memory, finite-state commitment, verified stopping, and bounded failover improve reliability without changing the underlying stack.

Across the three benchmarks, the executive changes outcomes that depend on target commitment and termination, whereas infeasible targets and topologically separated goals remain outside the scope of execution control. This distinction is important: the analysis attributes the gains to recoverable decision errors rather than claiming that the executive changes environmental reachability.

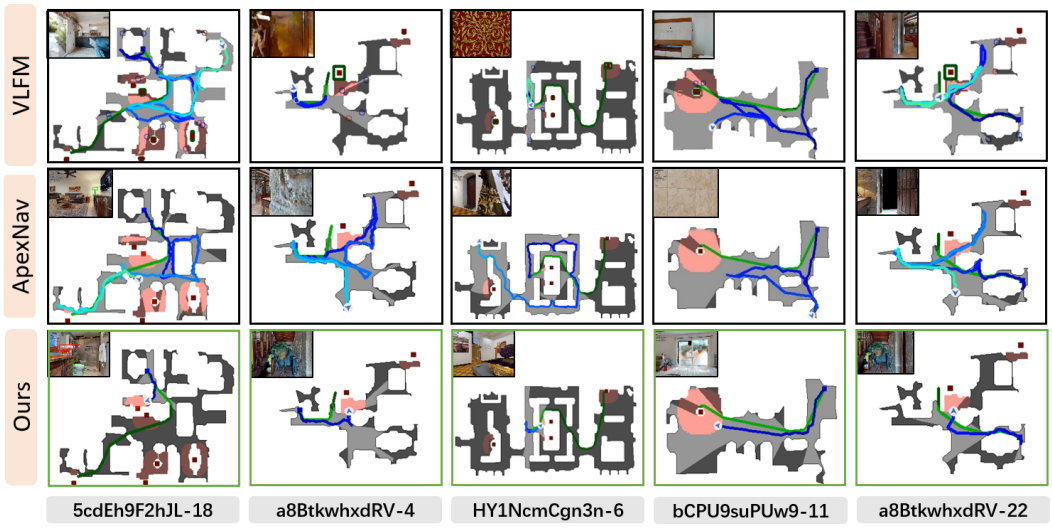


Figure 3: **Simulation results on HM3Dv2.** Qualitative comparison of ConsistNav, VLFM, and ApexNav. Each column shows one episode; green/blue paths denote reference/agent trajectories, and green/black frames denote success/failure.

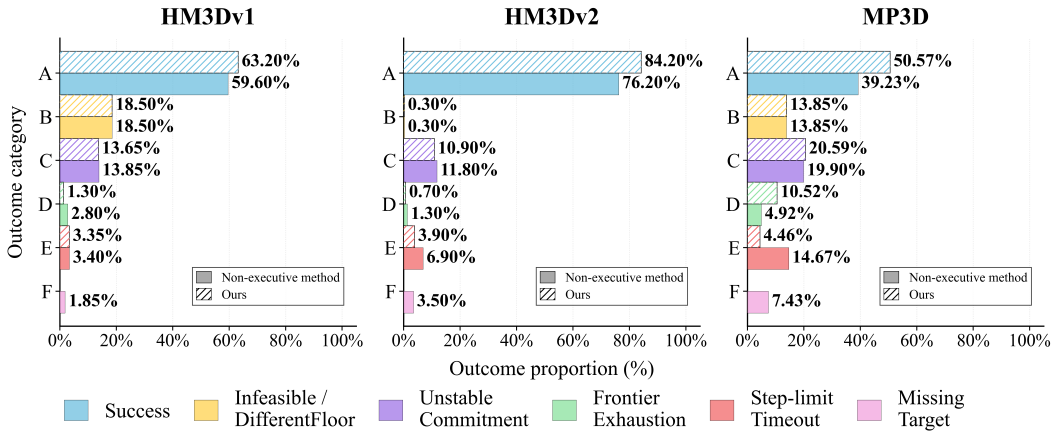


Figure 4: **Failure-cause comparison.** Outcome statistics for the Non-executive method and **ConsistNav** on HM3Dv1, HM3Dv2, and MP3D, covering verified success and five residual failure modes.

4.3 Failure Cause Analysis

We assign each episode to one outcome after inspecting the trajectory, semantic detections, stop decision, and target passage. As shown in Fig. 4, we use six categories: (1) *Success*, reaching and verifying the target; (2) *Infeasible / Different Floor*, unreachable or topologically separated targets; (3) *Unstable Commitment*, observed evidence followed by abandonment or unreliable stopping; (4) *Frontier Exhaustion*, no remaining frontier goals; (5) *Step-limit Timeout*, search or recovery exceeding the action budget; and (6) *Missing Target*, passing the target without verification.

Figure 4 compares ConsistNav with the Non-executive method and shows that the executive shifts controllable failures toward verified success. On HM3Dv2 and MP3D, success rises by $+8.00\%$ and $+11.34\%$ while timeout drops by 3.00% and 10.21%, indicating that memory, guarded commitment, and verified stopping sustain promising pursuits and reduce missed or premature termination. MP3D frontier exhaustion increases from 4.92% to 10.52%, reflecting a conservative trade-off: weak candidates become explicit search failures rather than unstable commitments, while infeasible and late-discovery cases remain dataset-level limits.

Table 2: Executive ablation on HM3Dv2 under the same underlying stack. PCM, FSEC, and SAAC indicate enabled executive components.

Variant	PCM	FSEC	SAAC	SR	SPL
Non-executive baseline	✗	✗	✗	76.20	38.00
PCM only	✓	✗	✗	77.10	39.60
FSEC only	✗	✓	✗	75.50	37.75
PCM + FSEC	✓	✓	✗	81.00	40.75
ConsistNav	✓	✓	✓	**84.20**	**41.24**

Table 3: Real-world deployment over four target categories, with 15 trials per target.

Target	Episodes	Success
Couch	15	10/15
Plant	15	12/15
Chair	15	14/15
TV	15	11/15
Overall	60	47/60

Table 4: Action consistency analysis on 150 shared episodes across 22 scenes. C_{P}, C_{R}, $C_{\mathrm{T}+}$, and $C_{\mathrm{T}-}$ denote commitment persistence, commitment release, valid termination, and rejection of invalid termination, respectively.

Method	$C_{\mathrm{P}} \uparrow$	$C_{\mathrm{R}} \uparrow$	$C_{\mathrm{T}+} \uparrow$	$C_{\mathrm{T}-} \uparrow$	ACS↑	SR↑	SPL↑
Non-executive	0.7143	0.6889	0.8333	0.1089	0.5292	13.33%	6.01%
PCM only	1.0000	0.9773	0.9444	0.0706	0.6319	12.00%	6.14%
FSEC only	0.7273	1.0000	0.2500	0.9911	0.7127	9.33%	4.60%
ConsistNav	0.8519	1.0000	0.2500	0.9758	**0.7493**	**19.33%**	**9.43%**

4.4 Ablation Study

Table 2 evaluates the three executive components under the same perception–mapping–planning stack. **(a) PCM.** PCM alone improves SR/SPL from 76.20/38.00 to 77.10/39.60. **(b) FSEC.** FSEC alone achieves 75.50/37.75. **(c) PCM + FSEC.** Combining PCM with FSEC raises performance to 81.00/40.75. **(d) SAAC.** Adding SAAC further improves SR/SPL to 84.20/41.24. Overall, the full ConsistNav improves SR/SPL by 8.00/3.24 percentage points over the non-executive baseline.

These results support the complementary roles of persistent memory, guarded execution, and action stabilization. PCM accumulates candidate evidence across observations, supporting target assessment and yielding a modest benefit on its own. FSEC uses this evidence to maintain supported commitments, reassess candidates, and coordinate verification and state transitions. Although FSEC alone does not improve performance, its gains when combined with PCM suggest that guarded execution benefits from persistent evidence. SAAC further strengthens this combination by detecting rotational stagnation and insufficient progress, invoking bounded recovery, and intercepting unauthorized STOP actions.

4.5 Action Consistency Analysis

To assess action consistency, we compare four executor configurations on the same 150 episodes across 22 scenes, yielding 600 navigation trajectories. In Table 4, PCM alone raises commitment persistence from 0.7143 to 1.0000, while configurations with FSEC attain $C_{\mathrm{R}} = 1.0000$ on eligible events and higher rejection of invalid termination. The termination components reveal a trade-off: configurations without FSEC have higher $C_{\mathrm{T}+}$ but low $C_{\mathrm{T}-}$, whereas guarded execution suppresses invalid STOP decisions while making valid termination more conservative.

Each ACS component is evaluated on its prespecified, fully observed opportunities and pooled from passed/eligible event counts. Censored or unscorable cases are excluded rather than treated as failures, and the score aggregates the resulting component rates. This protocol separates the quality of evidence-to-action decisions from episode outcomes that cannot be scored reliably; the complete definitions and event counts are provided in Appendix C.1.

On this experiment, full ConsistNav achieves the highest ACS (0.7493), SR (19.33%), and SPL (9.43%). Relative to the Non-executive baseline, ACS increases from 0.5292 to 0.7493, while SR and SPL increase by 6.00 and 3.42 percentage points, respectively. These comparisons illustrate the interaction between candidate memory, guarded transitions, and action stabilization, while highlighting valid termination as a remaining limitation. Event counts appear in Appendix Table C.3.

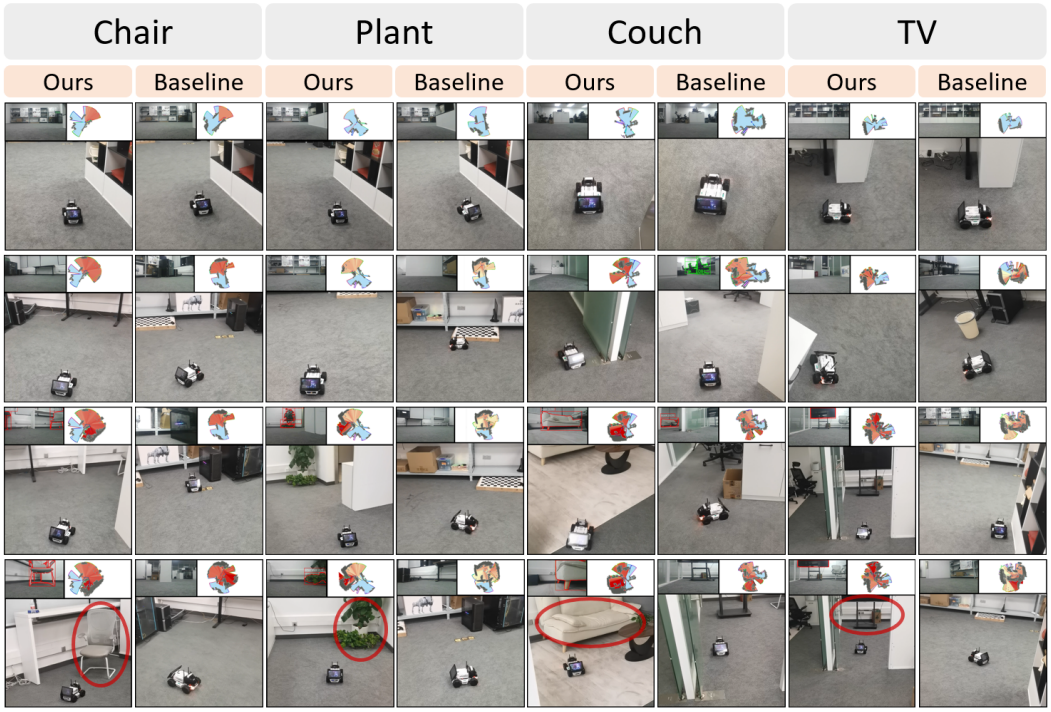


Figure 5: **Real-world deployment comparison.** Non-executive baseline and ConsistNav on four target tasks using AgileX LIMO, illustrating target pursuit, close-range verification, and stopping under real sensing and control latency.

4.6 Real-world Deployment

We deploy ConsistNav on an AgileX LIMO robot and run inference on an NVIDIA RTX 4090 workstation. Figure 5 provides qualitative real-world comparisons with baseline, illustrating representative behaviors during target pursuit, close-range verification, and stopping under real sensing and control latency. Table 3 reports the physical-scene evaluation: 60 navigation tasks are conducted in a $10 \times 10\,\mathrm{m}^2$ office scene and a $30 \times 10\,\mathrm{m}^2$ constructed home scene across four target categories—chair, plant, couch, and TV—with 15 trials per category. ConsistNav successfully completes 47 of the 60 tasks, achieving an overall success rate of 78.33%. Together, these qualitative and quantitative results support the practical feasibility of the proposed semantic executive in physical indoor environments, complementing the simulation-based evaluation.

5 Conclusion

ConsistNav addresses the action consistency gap in zero-shot ObjectNav: detecting a plausible target does not ensure sustained pursuit or valid termination. Its training-free design couples persistent candidate memory with a Finite-State Executive Controller. Accumulated evidence and execution feedback enable candidate reassessment and action correction, forming an execution process analogous to questioning and reflection. This coordination preserves supported commitments, revises them when contradicted, and conditions stopping on verification. We also introduce the Action Consistency Score (ACS), which complements SR and SPL by measuring commitment persistence, release, and termination. Controlled comparisons on HM3D and MP3D show improvements in success rate and SPL with perception and planning held fixed. Real-world robot experiments further support deployment feasibility.

AI USE STATEMENT

Generative AI tools were used for writing, editing, and formatting assistance only. All AI-assisted text was reviewed and revised by the authors, who take full responsibility for the paper's methods, experiments, claims, and final content.

ETHICS STATEMENT

This work evaluates navigation in public simulation benchmarks and controlled, low-speed indoor robot tests. It does not involve human-subject research or private data.

REPRODUCIBILITY STATEMENT

The paper reports benchmark splits, evaluation settings, implementation parameters, hardware, and runtime details. Code, episode lists, and configuration files will be released in an anonymized repository.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020.

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020a.

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in neural information processing systems*, 33:4247–4258, 2020b.

Michele Colledanchise and Petter Ögren. *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018.

Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE robotics & automation magazine*, 4(1):23–33, 1997.

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23171–23181. IEEE, 2023.

Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated planning - theory and practice*. Elsevier, 2004. ISBN 978-1-55860-856-6.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14809–14818. IEEE, 2022.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Yuxuan Kuang, Hai Lin, and Meng Jiang. Openfmnav: 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, 2024.

Steven Michael LaValle. *Planning algorithms*, volume 1. Cambridge university press Cambridge, UK, 2006.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. IEEE, 2022.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in neural information processing systems*, 35:32340–32352, 2022.

John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pp. 4628–4635. IEEE, 2017.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European conference on computer vision*, pp. 728–755. Springer, 2022.

Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Abhinav Rajvanshi, Karan Sikka, Xiao Lin, Bhoram Lee, Han-Pang Chiu, and Alvaro Velasquez. Saynav: Grounding large language models for dynamic planning to navigation in new environments. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pp. 464–474, 2024.

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18868–18878. IEEE, 2022.

Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5163–5173. IEEE, 2022.

Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17896–17906. IEEE, 2023.

Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 1689–1696. IEEE, 2020.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346. IEEE, 2019.

Dhruv Shah, Błażej Osiński, Brian Ichter, and Sergey Levine. LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pp. 492–504. pmlr, 2023.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav. *arXiv preprint arXiv:2303.07798*, 2023.

Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pp. 146–151. IEEE, 1997.

Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *Advances in neural information processing systems*, 37: 5285–5307, 2024.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42–48. IEEE, 2024.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3554–3560. IEEE, 2023.

Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

Lingfeng Zhang, Qiang Zhang, Hao Wang, Erjia Xiao, Zixuan Jiang, Honglei Chen, and Renjing Xu. Trihelper: Zero-shot object navigation with dynamic assistance. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10035–10042. IEEE, 2024.

Mingjie Zhang, Yuheng Du, Chengkai Wu, Jinni Zhou, Zhenchao Qi, Jun Ma, and Boyu Zhou. ApexNAV: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. *IEEE Robotics and Automation Letters*, 10(11):11530–11537, 2025. doi: 10.1109/LRA.2025.3606388.

Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International conference on machine learning*, pp. 42829–42842. PMLR, 2023.

TABLE OF CONTENTS IN APPENDIX

A Persistent Candidate Memory **15**
A.1 Candidate Priority 15
A.2 Revisit and Failed-Region Penalties 15
A.3 Parameter Settings 15
B Finite-State Executive Controller **16**
B.1 Guarded State Transitions 16
B.2 Execution Safeguards 16
B.3 Parameters and Control Procedure 17
C Experimental Setup and Reproducibility **19**
C.1 Simulation Protocol and Shared Implementation 19

A PERSISTENT CANDIDATE MEMORY

This section supplements Section 3.3 with candidate priority computation and memory-related parameter settings.

A.1 CANDIDATE PRIORITY

Let π_i denote the priority score of candidate i, with larger values indicating higher selection priority among eligible candidates. We denote the aggregate history penalty, revisit penalty, and failed-region penalty by P_i, R_i, and F_i, respectively. For compactness, time superscripts are omitted, and all quantities are evaluated at step t. In Eq. (6), $\mathbf{w} = (w_c, w_r, w_o, w_\iota, w_f) = (0.28, 0.22, 0.18, 0.18, 1.00)$ weights confidence, the target-observation ratio, repeated observation support, ITM history, and failure penalties, respectively. The priority is computed as

$$\pi_i = \hat{c}_i - n_i + 0.05 \min(5, m_{i,+}) + 0.45 s_i + 0.12 \operatorname{clip}(\iota_i) - 0.04 \min(6, m_{i,-}) - P_i, \quad \text{(A.1)}$$

where the aggregate history penalty is

$$P_i = 0.10 \min\left(6, \frac{\mathrm{age}_i}{\max(0.5, T_f)}\right) + 0.35 f_i + R_i + F_i. \quad \text{(A.2)}$$

Here, ι_i is the exponential moving average of ITM scores used for ranking, f_i is the recorded failure count, age_i denotes candidate age, and T_f is the age-normalization timescale. The remaining evidence terms follow Section 3.3.

A.2 REVISIT AND FAILED-REGION PENALTIES

The revisit penalty is

$$R_i = 0.5 \exp\left(-\frac{\Delta t_i}{30}\right), \quad \text{(A.3)}$$

where Δt_i denotes the elapsed time since the relevant recorded visit. Let v_i be the map cell containing candidate i, $H_t(v)$ the stored failure cost of cell v, and $\mathcal{N}_3(v_i)$ the 3×3 neighborhood centered at v_i. The failed-region penalty is

$$F_i = \max_{v \in \mathcal{N}_3(v_i)} H_t(v). \quad \text{(A.4)}$$

Each recorded failure increases the corresponding stored failure cost by 0.5. Thus, f_i records candidate-specific failure counts, whereas F_i summarizes spatial failure costs. No additional fixed penalty is applied solely because a candidate has been visited. During cooldown, the candidate priority is overridden to $-\infty$. Candidate selection additionally obeys the eligibility conditions in Section 3.3.

A.3 PARAMETER SETTINGS

The constant $\varepsilon > 0$ in Eq. (6) prevents division by zero in the observation-count ratio. Table A.1 lists the principal association, evidence-update, and admission parameters, which are fixed across the evaluated simulation benchmarks. Executive and planning parameters appear in Table B.2.

Table A.1: Principal parameters for persistent candidate memory.

Parameter	Description	Value
r_m	Spatial association radius	$0.6\,\mathrm{m}$
λ	Position smoothing coefficient	0.30
α	Confidence update coefficient	0.20
β	Verification-failure evidence increment	0.20
δ	Confidence decay factor after failure	0.80
τ_c	Minimum candidate confidence	0.15
τ_{cons}	Minimum candidate evidence score	0.42

B FINITE-STATE EXECUTIVE CONTROLLER

B.1 GUARDED STATE TRANSITIONS

Table B.1 summarizes the guarded transitions illustrated in Figure 2. Candidate eligibility and priority follow Section 3.3. If no exit condition applies, the executive retains its current state and continues the permitted exploration, pursuit, verification, or recovery action.

Verification and stopping. The effective approach-to-verification radius ρ_t is defined in Eq. (7). Reaching N_{verify} enlarges this radius and does not independently trigger VERIFY. The excessive-distance return from VERIFY uses its own admissible-range check. For a valid candidate within that range, verification remains pending until the bounded scan is complete. The executive then evaluates the acceptance conditions, assigning $\nu_t = \text{confirmed}$ or $\nu_t = \text{rejected}$. Candidate invalidation can interrupt the scan. Thus, an individual missed detection or an incomplete scan is not itself verification failure. STOP authorization is detailed in Section B.2.

Candidate identity during recovery. FAILOVER can retain the active candidate while refreshing its subgoal or requesting re-verification. Reselection uses updated eligibility and priority records and may return the same candidate. The executive returns to SEARCH when neither permitted continuation nor viable selection remains, or the recovery limit is reached. Rank loss without reconfirmation releases only a provisional SUSPECT commitment; a committed candidate does not change merely because rankings fluctuate.

B.2 EXECUTION SAFEGUARDS

These safeguards implement Stability-Aware Action Control (SAAC) within FSEC. In the staged ablation (Table 2), FSEC enables guarded phase control, and SAAC additionally enables these safeguards.

Candidate-directed subgoals. In SEARCH, the planner follows frontier exploration. In APPROACH and FINAL-APPROACH, a candidate-directed subgoal is selected from the reachable set $\mathcal{G}_t$:

$$\gamma_t^\star = \arg\min_{\gamma \in \mathcal{G}_t} \left[d(\gamma, \mu_{i_t}^{(t)}) + \lambda_v \mathbb{I}[\gamma \in \mathcal{R}_t] + \lambda_f \mathbb{I}[\gamma \in \mathcal{F}_t] \right]. \tag{B.1}$$

Here, $d(\gamma, \mu_{i_t}^{(t)})$ is the planner distance to the active candidate, $\mathcal{R}_t$ and $\mathcal{F}_t$ denote visited and failed regions, and λ_v and λ_f weight their penalties. Changing the local subgoal can therefore preserve candidate identity while discouraging repeated or failed approaches.

Rotation and progress checks. Let $\Delta p_t = \|p_t - p_{t-1}\|$ and $\Delta\theta_t = |\theta_t - \theta_{t-1}|$. The spin counter b_t^{spin} increases when $\Delta\theta_t > 0$ but $\Delta p_t < \delta_{\text{move}}$, and resets after sufficient translation. Once $b_t^{\text{spin}} > B_{\text{spin}}$, the current or resampled subgoal must satisfy a minimum translation margin. Translational progress and improvement in distance to the active candidate are monitored over a window of K_s steps, using a distance-improvement margin ϵ_d. Insufficient progress invokes verification, local recovery, or failover according to candidate support and proximity. Distance histories refer to the active candidate and are reinitialized when its identity changes.

Bounded recovery and memory feedback. Unsuccessful pursuit updates the candidate's negative evidence n_i, confidence, failure history, and availability, and records failed regions in $\mathcal{F}_t$. Local escape attempts are limited by a B_r-step budget. When permitted recovery does not restore progress, the executive revises the stopping point, reselects a viable candidate, or resumes exploration according to Table B.1. Failure updates influence the evidence score and priority in Section 3.3 and Appendix A.

STOP authorization. The phase-specific gate G_t combines target-hit evidence, confidence, observation support, ITM evidence, and the applicable checks on proximity, persistence, and evidence freshness. Confirmation alone does not authorize STOP, and historical best distance alone does not establish current stopping validity. Unexpected STOP commands outside SUCCESS are intercepted and routed to recovery or heading correction. SUCCESS authorizes the terminal action; the environment evaluates task success independently.

B.3 Parameters and Control Procedure

The executive uses a fixed parameter set across all simulated benchmarks rather than dataset-specific tuning. Tables B.1 and B.2 collect the guarded transitions and the executive and planning settings, respectively. Candidate-memory parameters are reported separately in Table A.1. Algorithm B.1 then gives the complete decision order.

Table B.1: Guarded transitions of the Finite-State Executive Controller.

State	**Execution role**	**Transition conditions → Next state**
SEARCH	Explore frontiers and rank candidate hypotheses.	Strong or persistent candidate evidence, or budget-triggered selection of an available candidate → SUSPECT.
SUSPECT	Maintain a provisional candidate and seek confirmation.	Reconfirmation or persistent rank dominance → APPROACH; candidate invalidation, loss of rank without reconfirmation, timeout, or an unreachable subgoal → SEARCH.
APPROACH	Retain the active candidate and pursue candidate-directed subgoals.	Entering the effective verification radius, or satisfying an evidence-conditioned near-target recovery trigger → VERIFY; pursuit failure or exhausted local recovery → FAILOVER.
VERIFY	Accumulate evidence over a bounded verification scan.	Missing or invalid candidate → FAILOVER; excessive distance → APPROACH. After the scan: stopping conditions satisfied → SUCCESS; target confirmed but further approach required → FINAL-APPROACH; no acceptance condition satisfied → FAILOVER.
FINAL-APPROACH	Preserve the verified target and complete the final approach.	Sustained close-range stopping conditions → SUCCESS; stale verification evidence → VERIFY; non-progress budget exhausted → FAILOVER.
FAILOVER	Continue a verified commitment when feasible, or recover through candidate reselection.	Close-range stopping conditions satisfied → SUCCESS; re-verification conditions satisfied → VERIFY; a viable candidate selected → APPROACH; no viable candidate or recovery limit reached → SEARCH. Otherwise, continue permitted local recovery.
SUCCESS	Authorize STOP.	Issue STOP; episode termination is handled by the environment.

Table B.2: Executive and planning parameters.

Parameter	Description	Value
r_v	base verify radius	0.8 m
T_s	suspect timeout	6.0 s
N_{search}	forced search-to-suspect budget	60 steps
N_{late}	late historical-candidate fallback budget	120 steps
N_{verify}	verify-radius expansion threshold	18 steps
r_{stop}	candidate-centered STOP-gate radius	0.28 m
r_{near}	near-stop radius	0.45 m
N_{near}	near-stop persistence length	10 steps
N_{turn}	verify turn budget	14 actions
τ_{conf}	verify confidence threshold	0.30
M_{obs}	verify observation threshold	2
τ_{itm}	ITM gate floor	0.12
$N_{\text{late-stop}}$	late-stop minimum step	220
N_{hold}	final close hold ticks	3
K_{frontier}	frontier shortlist size	12
K_{stop}	stop-point shortlist size	6
B_{room}	room-prior frontier bonus	22.0
N_{scan}	frontier scan budget per cluster	3

Executive control procedure. Algorithm B.1 gives the decision order, with per-state guards specified in Table B.1. If no exit applies, the current phase is retained.

Algorithm B.1: ConsistNav executive decision procedure.

Input: previous control state ξ_{t-1}, updated memory $\mathcal{C}_t$, map $\mathcal{M}_t$, pose x_t, and execution feedback r_t
Output: updated control state ξ_t and action a_t

Initialize $\xi_t \leftarrow \xi_{t-1}$; update progress and recovery summaries using r_t; let q, i denote its phase and candidate;
if $q =$ APPROACH **then**
  Update the candidate-directed subgoal and ρ_t using Eqs. (B.1) and (7);
  if pursuit failure with no permitted near-target recheck, or exhausted local recovery **then** record failure; $q \leftarrow$ FAILOVER;
  else if $d_t \leq \rho_t$ or an evidence-conditioned near-target recheck applies **then** $q \leftarrow$ VERIFY;
else if $q =$ VERIFY **then**
  if candidate missing or invalid **then** $q \leftarrow$ FAILOVER;
  else if candidate exceeds the admissible verification range **then** $q \leftarrow$ APPROACH;
  else
    Accumulate verification evidence;
    if scan incomplete **then** retain $q =$ VERIFY and pending status ν_t;
    else if $G_t = 1$ **then** $\nu_t \leftarrow \mathrm{confirmed}$; $q \leftarrow$ SUCCESS;
    else if candidate confirmed and further approach required **then** $\nu_t \leftarrow \mathrm{confirmed}$; $q \leftarrow$ FINAL-APPROACH;
    else $\nu_t \leftarrow \mathrm{rejected}$; record failed verification; $q \leftarrow$ FAILOVER;
else if $q =$ FINAL-APPROACH **then**
  if sustained close-range STOP gate satisfied **then** $q \leftarrow$ SUCCESS;
  else if verification evidence stale **then** $q \leftarrow$ VERIFY;
  else if non-progress budget exhausted **then** record failure; $q \leftarrow$ FAILOVER;
else if $q =$ FAILOVER **then**
  Evaluate permitted continuation, re-verification, and reselection; apply the applicable exit in Table B.1;
  Update i on reselection; otherwise retain it for the same commitment, including permitted local recovery when no exit applies;
else if $q =$ SEARCH **then**
  Apply the candidate-admission guard in Table B.1; select i on entry to SUSPECT;
else if $q =$ SUSPECT **then**
  if candidate reconfirmed or persistently dominant **then** $q \leftarrow$ APPROACH;
  else if candidate invalid, rank lost without reconfirmation, timed out, or subgoal unreachable **then** $q \leftarrow$ SEARCH;
Apply phase-entry counter resets; on entry to VERIFY, initialize the scan and set $\nu_t \leftarrow \mathrm{pending}$;
On return to SEARCH, release the active commitment;
Set $a_t \leftarrow \mathrm{STOP}$ if $q =$ SUCCESS; otherwise select the phase-conditioned navigation, scan, or recovery action a_t, intercepting unauthorized STOP and updating control accordingly;
Store the resulting phase, candidate, and history summaries in ξ_t after action corrections;
Record verification and recovery outcomes for subsequent memory and control updates;
return ξ_t, a_t;

C Experimental Setup and Reproducibility

C.1 Simulation Protocol and Shared Implementation

The controlled comparisons in Sec. 4 use the same sensing, perception, mapping, planning, and evaluation settings. The executive requires no training. Table C.1 summarizes the simulation protocol and shared implementation.

ACS event eligibility and denominators. For each component k, an event enters $\mathcal{E}_k$ only when its method-independent trigger occurs and its complete, fixed response window is observed. Let n_k^{pass}, n_k^{fail}, n_k^{cens}, and n_k^{uns} denote passed, failed, censored, and unscorable events. We use

$$|\mathcal{E}_k| = n_k^{\text{pass}} + n_k^{\text{fail}}, \qquad C_k = \frac{n_k^{\text{pass}}}{n_k^{\text{pass}} + n_k^{\text{fail}}}.$$

Censored and unscorable events are excluded from the denominator and reported separately. Component counts are pooled before applying Eq. (2); per-episode ACS values are not averaged.

Valid and invalid termination events. The controller's $r_{\text{stop}} = 0.28\,\text{m}$ condition uses estimated candidate-centered map geometry and contributes only one cue to G_t. It is distinct from Habitat's ground-truth distance-to-goal. A T+ event is a prespecified task-valid stopping opportunity and passes only when the policy voluntarily issues STOP within the frozen 12-action response window and Habitat validates the action using distance-to-goal $\leq 0.2\,\text{m}$. A fully observed opportunity without such a STOP is a failure; timeout termination, an environment-forced termination, or an internal controller success flag is not a pass. A T− event is a prespecified invalid stopping opportunity and passes when the policy withholds STOP throughout the same response window.

Boundary cases of ACS. For nonempty event sets, Eq. (2) has the continuous extension

$$\text{ACS} = C_{\text{P}}^{1/3} C_{\text{R}}^{1/3} C_{\text{T+}}^{1/6} C_{\text{T−}}^{1/6}, \qquad C_k \in [0, 1].$$

Thus, ACS equals zero whenever any component rate is zero. If an event set is empty, its pass rate is undefined; the corresponding component and the overall ACS are reported as not available (N/A), rather than assigned zero. No ε-smoothing is applied.

Action consistency event counts. Table C.3 reports the passed/eligible event counts underlying Table 4. A shared episode set does not imply equal event denominators, because configurations encounter different eligible events along their trajectories.

Table C.1: Simulation protocol and shared implementation.

Parameter	Value
Evaluation protocol	
Task	Habitat ObjectNav (Savva et al., 2019; Szot et al., 2021)
Simulator	Habitat-Lab / Habitat-Sim v0.3.1
Action space	FORWARD, LEFT, RIGHT, STOP
Action budget	500 actions per episode
Success condition	Executed valid STOP with Habitat distance-to-goal $\leq 0.2\,\text{m}$
Task metrics	SR and SPL
Shared perception and mapping	
Open-vocabulary detector	GroundingDINO / YOLO-World
Segmentation model	MobileSAM (ViT-T)
Image–text matching model	BLIP-2
RGB-D sensing range	$[0, 5]\,\text{m}$
Valid map-depth range	$[0, 4.99]\,\text{m}$
Retained map-height band	$[0.28, 1.18]\,\text{m}$
Map representation	Occupancy / semantic SDF
Map extent	$80 \times 80\,\text{m}^2$
Exploration and planning	
Path search	Target-conditioned A*

Table C.2: Hardware and sensor specifications for the real-world deployment.

Parameter	Value
Mobile platform	
Platform model	AgileX LIMO
Dimensions (L × W × H)	$322 \times 220 \times 251\,\mathrm{mm}$
Mass	$4.8\,\mathrm{kg}$
Maximum speed	$1.0\,\mathrm{m/s}$ (unloaded)
Minimum turning radius	$0.4\,\mathrm{m}$
Drive mode	Ackermann
Power supply	$12\,\mathrm{V}$ battery
RGB-D sensor	
Camera model	Orbbec DaBai RGB-D
Depth range	0.3–$3.0\,\mathrm{m}$
RGB resolution	640×480
RGB–depth alignment	Aligned
Horizontal field of view	$\approx 67.5^\circ$
Camera mounting height	$0.16\,\mathrm{m}$

Table C.3: Passed/eligible event counts for the 150-episode shared evaluation in Table 4. Each denominator is $n_k^{\mathrm{pass}} + n_k^{\mathrm{fail}}$.

Method	Persistence (P)	Release (R)	Valid STOP (T+)	Invalid STOP rejection (T−)
Non-executive	10/14	31/45	20/24	11/101
PCM only	7/7	43/44	17/18	6/85
FSEC only	8/11	53/53	5/20	111/112
ConsistNav (full)	23/27	51/51	11/44	121/124