EONAV: SUPPORT-CENTRIC MEMORY AND ACTIVE SEARCH FOR EVERYDAY OBJECT AGENTIC NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Everyday object navigation spans space and time: small targets may require close inspection of plausible supporting surfaces, while portable objects can relocate between searches. An agent must therefore decide where to inspect, how to use potentially stale sightings, and which new observations to trust. We introduce an *Everyday Object Navigation* benchmark with 560 static episodes and 600 ordered dynamic episodes across 35 scenes, jointly evaluating fine-grained discovery and repeated search under relocation. We further propose **EONav**, a training-free, VLM-centered framework that connects memory, planning, and active verification through relatively stable supports. Its memory agent retains multiple verified, timestamped target–support observations; their historical frequencies provide an empirical support-level prior, with recency as an additional cue. A VLM planning harness combines this history with current observations and inspection feedback to select supports to revisit, falling back to support-guided exploration when history is unavailable or exhausted. Active verification obtains closer or alternative views, redirecting search after rejections and admitting only confirmed observations to memory. EONav achieves 54.46% success rate (SR) and 26.58% success weighted by path length (SPL) in static search, and 61.50% SR and 40.14% SPL in dynamic search, outperforming the evaluated baselines. Temporal memory raises dynamic SR from 23.26% to 61.50%, while EONav uses 23.1% fewer tokens per episode than STRIVE. We also demonstrate real-robot object search and will release the code and benchmark.

1 INTRODUCTION

Object-goal navigation (ObjectNav) asks an agent to locate an object of a specified category (Batra et al., 2020). In homes and workplaces, everyday targets vary in size and in the precision required to find them: a salient object may be recognized from a broad view, whereas a marker pen may require close inspection of a particular surface (Figure 1 a). Search must therefore move from room-level exploration to support-level inspection and object-level verification. It also extends across time. Objects can relocate between searches, making the last observed location unreliable even when earlier observations remain informative.

First, existing benchmarks address parts of this problem but rarely combine these demands. HM3D ObjectNav mainly evaluates furniture and fixtures (Yadav et al., 2023); HM3D-OVON expands category coverage (Yokoyama et al., 2024b); and GOAT-Bench evaluates sequential, instance-specific goals with memory reuse (Khanna et al., 2024). These benchmarks do not jointly evaluate fine-grained discovery across everyday object scales and the reuse and revision of memory after object relocation.

Second, existing navigation methods offer complementary capabilities (Figure 1b). LightNav-0 and Uni-NaVid learn navigation policies from data (Wang et al., 2026a; Zhang et al., 2025a), while zero-shot methods such as VLFM and STRIVE search using observations acquired within an episode (Yokoyama et al., 2024a; Zhu et al., 2026). Pre-built maps support instance retrieval (Gu et al., 2024; Werby et al., 2024), and DualMap associates movable objects with stable anchors as scenes change (Jiang et al., 2025). Everyday object search nevertheless requires three decisions

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.

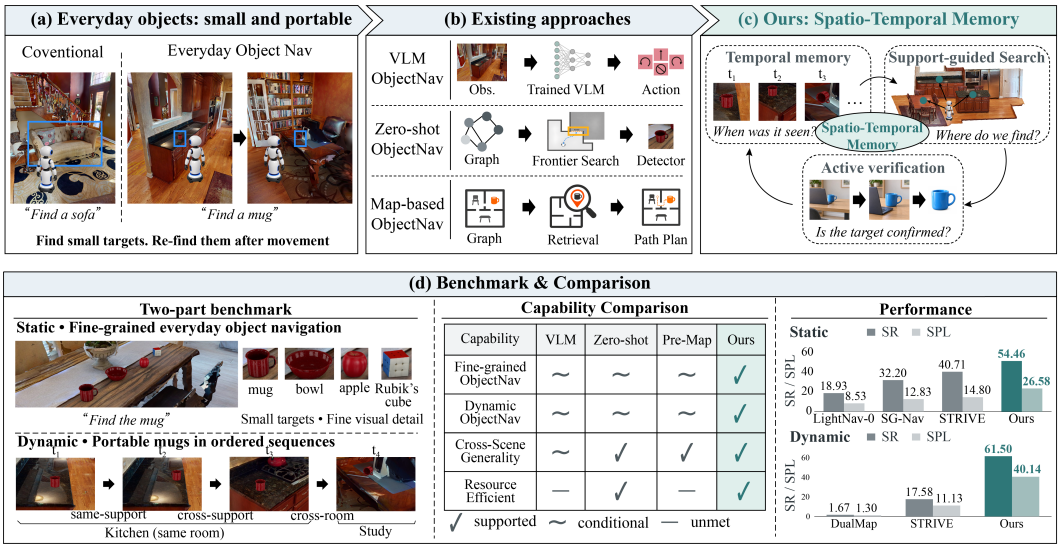

Figure 1: Everyday Object Navigation. (a) Small, portable targets require fine-grained search and repeated localization after relocation. (b) Existing navigation approaches provide complementary capabilities in learned control, semantic exploration, and map retrieval. (c) Our framework enables efficient and reliable everyday object navigation by integrating temporal memory, support-relavance guided search, and active verification. (d) The benchmark evaluates static object discovery and ordered relocation sequence. The plots report the performance of different methods.

to work together. *Where to revisit:* after an unobserved relocation, the last sighting may be stale, but earlier object–support associations can still guide search. *Where to inspect:* room-level exploration must lead to deliberate views of plausible supports, since a small target may be invisible from distant viewpoints. *What to remember:* uncertain detections must be checked before they guide future searches. The challenge is to connect historical evidence, fine-grained exploration, and reliable memory updates in a training-free search loop.

To bridge this gap, we introduce an *Everyday Object Navigation* benchmark to evaluate these capabilities (Figure 1d). It contains 560 independent static episodes and 600 dynamic episodes across 35 HM3D validation scenes, with the dynamic episodes organized into 50 ordered sequences of 12 searches. Targets remain stationary within an episode but may relocate between episodes. An automated pipeline proposes semantically plausible placements and checks their physical feasibility, visibility, and navigability. The benchmark thus evaluates both initial discovery and repeated search as object locations change. On this benchmark, STRIVE, the strongest evaluated existing baseline, achieves 40.71% success rate in static search and 17.58% in dynamic search. These results show substantial room for improvement in both initial discovery and repeated search.

We therefore introduce **EONav**, a training-free, VLM-centered agentic framework (Figure 1c). EONav closes a support-centered loop: past sightings guide search, failed inspections revise the candidate set, and verified discoveries extend memory. Its *memory agent* aligns supports across episodes and records multiple verified, timestamped target–support observations in a sparse spatiotemporal graph. Their frequencies provide an empirical support-level prior, complemented by recency and stored visual evidence. To make this memory actionable, we design a *VLM planning harness* that interleaves memory retrieval, navigation, and inspection feedback. It presents eligible historical supports alongside the current view, travel cost, and recent outcomes; VLM reasoning selects where to revisit, while the harness invokes navigation and verification and updates candidate eligibility. When usable history is absent or exhausted, it switches to support-guided exploration. During *active verification*, the agent obtains closer or alternative views before an uncertain detection can terminate search or enter memory. Rejections redirect the search, whereas confirmations create new memory events. Together, these mechanisms focus exploration on promising supports, redirect search when past locations prove stale, and prevent uncertain detections from corrupting future memory, improving both navigation success and path efficiency with fewer tokens.

EONav outperforms the evaluated baselines in both settings, achieving 54.46% success rate (SR) and 26.58% success weighted by path length (SPL) in static search, and 61.50% SR and 40.14% SPL in dynamic search. On dynamic sequences, removing temporal memory reduces SR to 23.26% and SPL to 14.82%, highlighting the value of retaining support histories. Compared with STRIVE, EONav also uses 23.1% fewer tokens per episode in dynamic search. Qualitative real-robot demonstrations further illustrate the framework in physical indoor environments. Our contributions are:

- **Everyday Object Navigation benchmark:** We evaluate multi-granularity fine-grained discovery of everyday objects and repeated search across episodes in which objects may relocate.
- **Support-centered temporal memory:** A spatiotemporal graph retains multiple verified target–support observations across episodes, using their frequency and recency to inform future search.
- **Feedback-driven VLM planning agent harness:** A training-free reasoning–action–feedback loop combines support history with current observations, rules out inspected locations, and switches to support-guided exploration when history is unavailable or exhausted.
- **Verification-guided search and memory updates:** Close or alternative views resolve uncertain detections; rejected candidates redirect search, while confirmed observations become evidence for subsequent episodes.

2 Related Work

Object Navigation Benchmarks. ObjectNav defines category-level search (Batra et al., 2020); HM3D-OVON expands the vocabulary (Yokoyama et al., 2024b), while InstanceImageNav and GOAT-Bench add instance-specific goals (Krantz et al., 2023; Khanna et al., 2024). GOAT-Bench also evaluates memory reuse with fixed object placements. Portable-object finding models habit-driven motion (Dorbala et al., 2024); SD-OVON generates semantically and physically plausible scene variants (Qiu et al., 2025), and EvoNav-Bench evaluates navigation under staged scene changes (Wang et al., 2026b). Our benchmark pairs small-object discovery with repeated search under object relocation.

Object Navigation Methods. Learned approaches include semantic exploration (Chaplot et al., 2020) and video-language policies (Zhang et al., 2025a). Zero-shot methods use language-grounded frontier maps (Yokoyama et al., 2024a) and scene graphs (Yin et al., 2024; 2025). STRIVE integrates room-level planning, viewpoint exploration, and verification (Zhu et al., 2026). SG-Nav uses re-perception (Yin et al., 2024), while SAP-Nav selects informative viewpoints (Pei et al., 2026). However, these methods are not specifically designed for small-object discovery and typically treat each search as an independent episode, requiring renewed exploration rather than reusing scene memory from previous searches.

Memory for Lifelong Navigation. GOAT and 3D-Mem retain instance-aware or snapshot-based experience (Chang et al., 2024; Yang et al., 2025). STEGNav represents navigation events (Chen et al., 2026b), STAR interleaves memory queries with actions (Chen et al., 2026a), and DynaMem updates memory as scenes change (Liu et al., 2025). DualMap associates movable objects with stable anchors and reselects candidates after unsuccessful observations (Jiang et al., 2025). EONav retains multiple timestamped target–support records and uses inspection feedback to guide revisitation and update memory.

3 Method

Task formulation. Given a target category g, the agent receives posed RGB-D observations $o_{k,t} = (I_{k,t}, D_{k,t}, P_{k,t})$ and must locate and stop near the target within $T_{\max}$ actions, without a pre-built scene map. Here, k and t index episodes and actions; I, D, and P denote the RGB image, depth, and pose. We consider independent searches without cross-episode memory and repeated searches retaining earlier experience in the same environment. Targets remain stationary within episodes but may relocate between them.

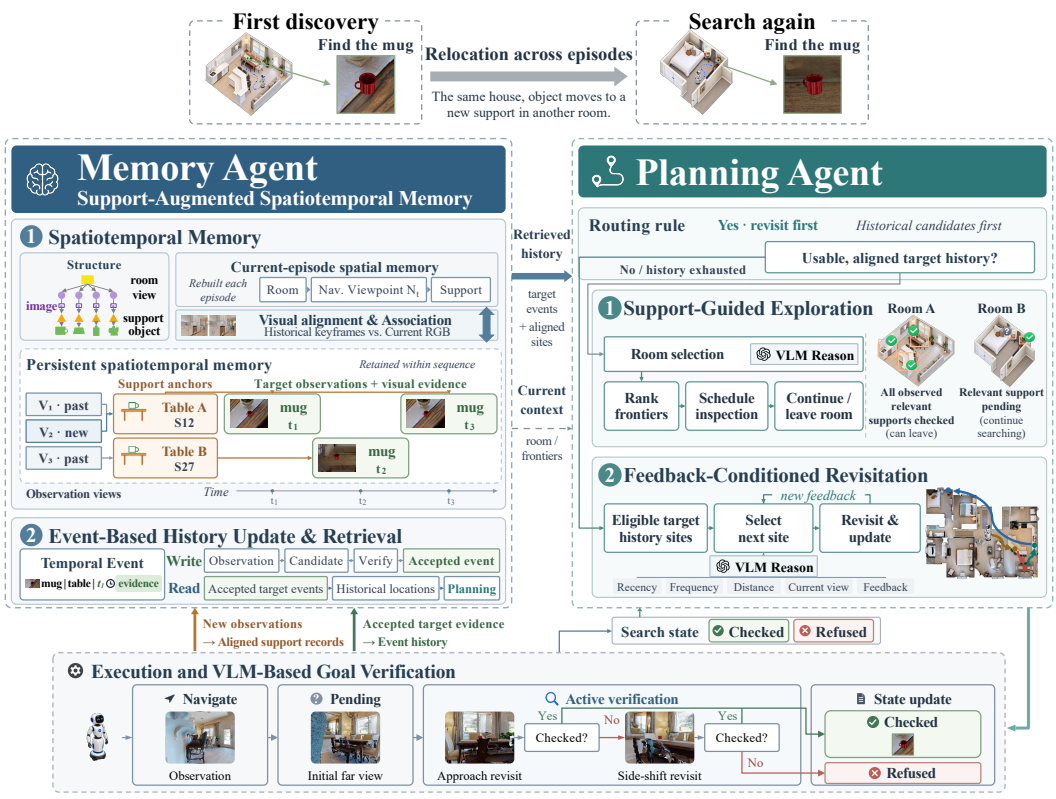

Figure 2: Overview of EONav. The memory agent combines current spatial context with persistent target histories anchored to supports. The planning agent revisits eligible historical locations or performs support-guided exploration. Active verification checks candidate targets, guides subsequent search through inspection feedback, and supplies verified observations for memory updates.

Overview. EONav combines a *memory agent* organizing spatial context and target history around supports (Section 3.1) with a *planning agent* selecting support-guided exploration or feedback-conditioned revisitation (Section 3.2). Both branches share execution and verification (Section 3.3), as shown in Figure 2.

3.1 Memory Agent: Support-Augmented Spatiotemporal Memory

The memory agent incrementally constructs a sparse spatiotemporal graph connecting rooms, viewpoints, supports, and objects (Figure 2). Its *room–viewpoint–support* structure provides the current episode's *spatial context*, while *support–object* relations form persistent *temporal events* describing where and when targets were observed. Relatively stable supporting surfaces connect these two components, with visual alignment establishing their correspondence across episodes.

Current-episode spatial context. Following STRIVE (Zhu et al., 2026), rooms group navigable viewpoints, and viewpoint–viewpoint edges encode direct traversability. We add support nodes reconstructed from segmented RGB-D observations, storing category, 3D position, and spatial extent. Each viewpoint–support edge records relative geometry and references a *keyframe*, a stored observation image used for cross-episode visual alignment. The spatial graph is rebuilt each episode to support zero-shot exploration when history is unavailable or unusable.

Persistent temporal events. Across episodes, we retain support anchors and their observation views. Object nodes store target category and geometry. Each temporal event links category g to support s at episode k, with an observation viewpoint and visual evidence. Writes require visual confirmation (Section 3.3) and unambiguous geometric association with an aligned support; reads

use only accepted events from episodes before k. For supports with accepted history, let $\mathcal{T}_k(s,g)$ contain the distinct earlier episode indices at which g was verified on s. We define

$$n_k(s,g) = |\mathcal{T}_k(s,g)|, \quad r_k(s,g) = \max \mathcal{T}_k(s,g), \quad f_k(s,g) = \frac{n_k(s,g)}{\sum_{s'} n_k(s',g)}. \tag{1}$$

The episode-level count n_k, empirical frequency f_k, and last-observed episode r_k, together with stored observation views, guide support revisitation.

Visual alignment. We retrieve historical keyframes using DCT coefficients and HSV histograms. SIFT matching and depth back-projection provide multi-view 3D correspondences for estimating translation from the episode frame to the memory frame. Aligned supports are associated with historical anchors using category, geometry, and appearance.

3.2 Planning Agent

Routing between exploration and revisitation. The planning agent receives both current spatial context and retrieved target history. It follows the routing rule in Figure 2: revisit eligible historical locations when memory is aligned; otherwise, perform support-guided exploration. A historical support is eligible if it has an accepted target record and has neither been checked nor deferred as blocked in the current episode. The agent searches these eligible locations before resuming exploration. When history is absent, exhausted, or cannot be aligned, it explores using the current map.

3.2.1 Support-Guided Exploration

Room selection and viewpoint ranking. Room selection follows STRIVE's VLM planner, conditioned on the goal, observed room contents, connectivity, and navigation progress (Zhu et al., 2026). Within a room, STRIVE clears true frontiers before deciding whether to explore remaining inner frontiers. Our support-guided exploration (Figure 3) prioritizes frontiers near target-relevant supports and can leave once the observed relevant supports have been checked, without exhausting the room's frontiers. This focuses the action budget on useful inspection locations and provides close views even before a target is detected, helping reveal fine-grained objects that can be missed or misidentified from distant viewpoints. A VLM proposes target–support relevance scores $\rho(g, c_s) \in [0,1]$ for target g and support category c_s. We combine this relevance with travel distance and inspection feedback to rank candidate viewpoints:

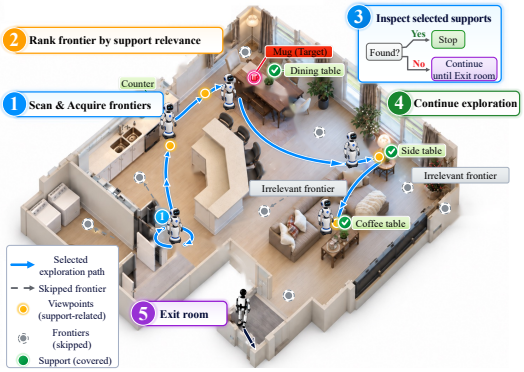

Figure 3: Schematic of support-guided exploration.

$$v_t^* = \arg\max_{v \in \mathcal{F}_t} \left[-\|p_v - p_t\|_2 + \mathrm{clip}(b_t(v,g), -B, B) \right]. \tag{2}$$

Here, $\mathcal{F}_t$ is the room's frontier set and p_v, p_t are horizontal positions. In priority order, $b_t(v,g)$ penalizes proximity to inspected supports, encourages additional views of partially observed supports, or rewards target–support relevance. The bias is zero without nearby supports; clipping limits its contribution to $[-B, B]$. The complete bias rule is given in Appendix C.

Support inspection and room exit. For each selected viewpoint, the agent inspects a nearby relevant unchecked support, when available, from a suitable position facing it. The shared detector–VLM verification procedure updates the support's search state and guides subsequent actions. Search continues while relevant supports remain pending. The agent may leave a room when it has met the minimum exploration requirement, checked all observed relevant supports, and encountered no new pending supports for a specified interval. It then selects another room, without visiting every frontier (Figure 3).

3.2.2 Feedback-Conditioned Revisitation

Selecting the next historical location. Figure 4 shows how the planner combines historical evidence with current feedback. For each eligible support, the memory agent supplies $f_k(s, g)$, $r_k(s, g)$, and the distance from the agent after alignment. The VLM also receives the current RGB view, remaining action budget, and recent navigation and inspection outcomes. It selects the next support using this context, allowing the ordering to change after each check. A sole eligible support is selected without VLM inference, whereas multiple eligible supports are evaluated through VLM-based reasoning.

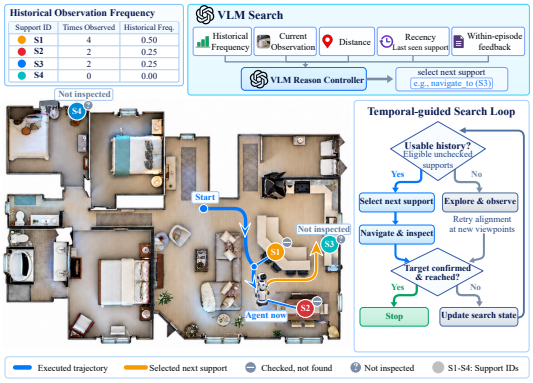

Figure 4: Feedback-conditioned search. Temporal evidence, spatial context, and inspection feedback guide historical-support selection. Without usable history, the agent explores and retries alignment.

Revisitation and feedback. The planning agent interleaves memory retrieval, navigation, and inspection in a reasoning–action–feedback loop (Yao et al., 2023; Li et al., 2026). Revisit outcomes update the eligible support set and inform subsequent decisions, excluding inspected supports and deferring blocked locations within the current episode. When no historical candidate remains eligible, the agent resumes support-guided exploration to discover new target locations. Verified observations extend temporal memory with new support–object events.

3.3 Execution and VLM-Based Goal Verification

Both planning branches share the navigation and verification procedure in Figure 2. A detector proposes target candidates, and then a VLM verifies the target using their appearance and surrounding context (Zhu et al., 2026). Candidates that pass this check but are still far away are marked as *pending*. The agent moves closer for further confirmation, repeating detection and VLM verification and checking 3D consistency with the original instance. If verification remains inconclusive, it may acquire an additional lateral view. Inspection outcomes update the search state and temporal memory, informing subsequent planning.

4 Everyday-Object Navigation Benchmark

Built from HM3D indoor scenes (Ramakrishnan et al., 2021) and YCB assets (Calli et al., 2015), our benchmark comprises static and dynamic sets for independent discovery of small, portable objects and repeated search after relocation, respectively.

As shown in Figure 5, scene semantics and RGB-D geometry identify candidate supports, and a VLM agent proposes semantically compatible object–support placements using scene context. Placements undergo Habitat-Sim physical settling and contact, stability, and collision checks. Navmesh-sampled goal viewpoints are checked for visibility through semantic rendering, while episodes are validated for reachability, navmesh-island consistency, and path geometry.

For dynamic sequences, a VLM agent uses object semantics and valid supports to plan relocations of the same instance that reflect plausible everyday usage, including local shifts and transitions between supports or rooms. Each state undergoes physical settling and render-based visibility checks before inclusion. Further construction details are provided in the appendix.

The static set contains 560 episodes across 35 scenes and 12 target categories. Each episode is an independent search. The dynamic set contains 600 episodes across 35 scenes, organized into 50 sequences of 12 episodes. Each sequence starts with empty memory and is evaluated chronologically, retaining observations from earlier episodes to support subsequent searches. This setting tests whether experience remains useful when target locations change.

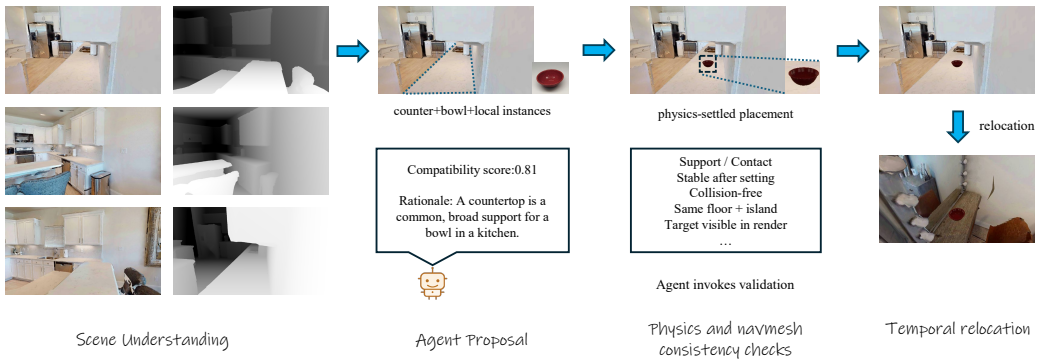

Figure 5: Benchmark construction. Scene context guides object–support proposals; physics, visibility, and navigation checks validate placements and episodes for static search and ordered relocation.

5 Experiments

We evaluate EONav on static everyday-object search and repeated search under object relocation. We first compare navigation performance with existing methods, then examine the contributions of support-guided exploration, temporal memory, and historical-location scheduling.

5.1 Experiment Setup

Benchmark and Metrics. We conduct simulation experiments on the Everyday-Object Navigation benchmark introduced in Section 4. All baselines are strictly evaluated using their official implementations. We report Success Rate (SR), which measures the proportion of successful episodes, and Success weighted by Path Length (SPL) Yin et al. (2025); Krantz et al. (2023), which accounts for both task success and path efficiency.

Implementation details. Each episode allows a maximum of 500 steps and a success distance of 1.0m. The simulated agent receives 640×480 RGB-D observations with a 79° horizontal field of view, a camera height of 0.88 m, and a depth range of 0.5–5.0 m. Forward actions move 0.25 m and turns rotate 30°. We use pretrained MM-GroundingDINO (Zhao et al., 2024) and SAM (Kirillov et al., 2023) for object detection and segmentation. For VLM, we use gemini-3.7-flash. For real-robot deployment, we use a Unitree G1-D platform. All experiments are conducted on RTX3090.

5.2 Main Results

Table 1: Static everyday-object navigation. Each episode is evaluated independently without cross-episode memory. SR and SPL are percentages.

Method	SR (%) ↑	SPL (%) ↑
End-to-end VLA methods		
Uni-NaVid (Zhang et al., 2025a)	0.95	0.28
LightNav-0 (Wang et al., 2026a)	18.93	8.53
Modular zero-shot methods		
ApexNav (Zhang et al., 2025b)	11.69	5.83
VLFM (Yokoyama et al., 2024a)	13.60	6.26
SG-Nav (Yin et al., 2024)	32.20	12.83
STRIVE (Zhu et al., 2026)	40.71	14.80
Ours	**54.46**	**26.58**

Static everyday-object navigation. As shown in Table 1, EONav outperforms all baselines, achieving 54.46% SR and 26.58% SPL. Compared with STRIVE, the strongest baseline, EONav

improves SR and SPL by 13.75 and 11.78 percentage points, respectively. These results support the effectiveness of combining support-guided exploration with active verification for small-object search.

Table 2: Dynamic everyday-object navigation and temporal-memory ablation. Steps and tokens are averaged over all episodes.

Method	SR (%) ↑	SPL (%) ↑	Avg. steps (all episodes)	Avg. tokens per episode ↓
DualMap	1.67	1.30	39.83	–
STRIVE	17.58	11.13	296.71	55,607
Ours w/o temporal memory	23.26	14.82	266.62	39,005
Ours	**61.50**	**40.14**	**235.06**	**42,777**

Dynamic everyday-object navigation. Table 2 compares EONav with DualMap (Jiang et al., 2025) and STRIVE (Zhu et al., 2026) under inter-episode object relocation and evaluates the contribution of temporal memory. EONav achieves 61.50% SR and 40.14% SPL, outperforming both baselines while using 20.8% fewer actions and 23.1% fewer tokens per episode than STRIVE. Enabling temporal memory improves SR and SPL by 38.24 and 25.32 percentage points, respectively, with only a 9.7% increase in token usage, demonstrating substantial gains at modest token overhead for repeated search as object locations change. Even without temporal memory, EONav achieves higher SR and SPL than STRIVE with fewer actions and tokens, supporting the effectiveness and efficiency of its support-guided exploration and verification pipeline.

5.3 Ablation Study

Table 3: Ablation studies: (a) historical-location scheduling on 600 dynamic episodes; (b) support-guided search in static episodes.

(a) Historical scheduling

Strategy	SR (%) ↑	SPL (%) ↑
Recency	60.00	36.74
Frequency	61.00	38.26
Frequency + Recency	60.33	39.27
Full	**61.50**	**40.14**

(b) Support-guided search

Variant	G	I	E	SR (%) ↑	SPL (%) ↑
Ours	✓	✓	✓	**54.46**	**26.58**
w/o Support ranking	–	✓	✓	52.73	25.59
w/o Local inspection	✓	–	✓	51.27	24.73
w/o Early room exit	✓	✓	–	53.09	24.05
All three off	–	–	–	52.14	22.31

Historical-location scheduling. Table 3(a) shows limited SR variation (60.00–61.50%), as scheduling reorders historical supports without excluding eligible sites. The inspection order nevertheless affects travel before finding the target, with the full scheduler improving SPL from 36.74% (Recency) to 40.14%, highlighting its benefit for search efficiency.

Support-guided search. Table 3(b) highlights two benefits of support-guided search: improved object discovery and more efficient exploration. Removing local inspection causes the largest individual SR drop (3.19 percentage points), despite retaining target confirmation, suggesting that deliberate support-facing observations are important for discovering small. Support ranking helps prioritize promising search regions, while the largest individual SPL drop without early room exit (2.53 points) supports its role in limiting low-yield exploration. Together, these components increase SPL from 22.31% to 26.58%, demonstrating the value of coordinating where to search, how to inspect, and when to move on for fine-grained object navigation.

5.4 Qualitative Results

Simulation. Figure 6 highlights how visual alignment makes temporal memory actionable, while inspection feedback guides candidate reselection. EONav explores until alignment is established,

then prioritizes historical supports. Inconclusive checks at A and B redirect search to C, where the target is confirmed. If historical candidates are exhausted, support-guided zero-shot exploration resumes within the remaining action budget.

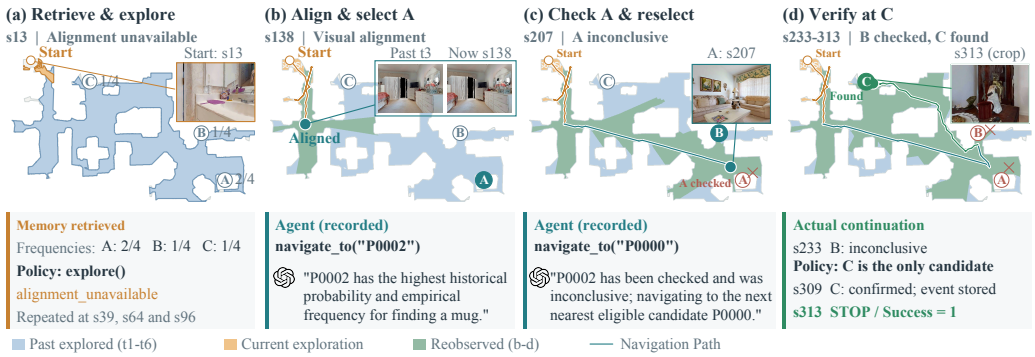

Figure 6: Qualitative example of temporal-memory-guided navigation. (a) Support-guided exploration before visual alignment. (b) Keyframe matching enables historical search toward support A. (c) An inconclusive inspection at A prompts reselection of B. (d) After B also yields no confirmation, EONav selects C and finds the target.

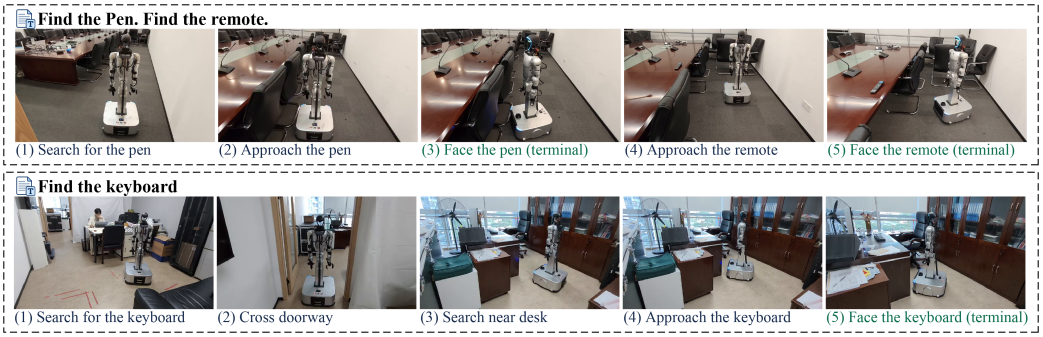

Figure 7: Real world implementation.

Real-Robot Deployment Fig. 7 presents real-world searches for a pen, a remote control, and a keyboard in conference-room and office environments. The robot searches near relevant supports, approaches the requested object, and stops facing it. The keyboard-search task additionally involves navigating between rooms before inspecting the desk area. Our method handles high-level search planning and target verification, while Unitree's built-in system provides low-level motion control and obstacle avoidance. These examples provide qualitative evidence that our support-guided search and verification pipeline is applicable to everyday-object navigation in physical indoor environments.

6 Conclusion

We presented EONav for finding small, movable objects in everyday environments. Its memory and planning agents combine support-guided exploration with temporal revisitation, using active verification to guide search and update memory. We also introduced an Everyday Object Navigation benchmark for independent discovery and repeated search after relocation. Experiments show higher success rates and SPL than the evaluated baselines in both settings, with ablations supporting the contributions of support-guided search and temporal memory. Real-robot demonstrations further illustrate the framework's applicability to object search in physical indoor environments.

AI USE STATEMENT

We used AI tools for literature retrieval, English translation, and language polishing. AI tools also assisted in analyzing experimental results and drafting text based on experimental data and outlines provided by the authors. The authors subsequently reviewed and revised the AI-assisted text.

REFERENCES

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, 2020.

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research. In *International Conference on Advanced Robotics*, pp. 510–517, 2015. doi: 10.1109/ICAR.2015.7251504.

Matthew Chang, Théophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, Roozbeh Mottaghi, Jitendra Malik, and Devendra Singh Chaplot. GOAT: GO to any thing. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado (eds.), *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.073.

Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Russ R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Taijing Chen, Sateesh Kumar, Junhong Xu, Georgios Pavlakos, Joydeep Biswas, and Roberto Martín-Martín. Searching in Space and Time: Unified Memory-Action Loops for Open-World Object Retrieval. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026a.

Yang Chen, Zhenyu Huang, Wenbo Fu, Danyang Peng, Shi-Yu Tian, Kun-Yang Yu, and Lan-Zhe Guo. STEGNav: Spatio-Temporal Event Graph Reasoning for Multimodal Lifelong Object Navigation, 2026b.

Vishnu Sashank Dorbala, Bhrij Patel, Amrit Singh Bedi, and Dinesh Manocha. Personalized Embodied Navigation for Portable Object Finding, 2024.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 5021–5028. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610243.

Jiajun Jiang, Yiming Zhu, Zirui Wu, and Jie Song. DualMap: Online Open-Vocabulary Semantic Mapping for Natural Language Navigation in Dynamic Changing Scenes. *IEEE Robotics and Automation Letters*, (12):12612–12619, 2025. doi: 10.1109/LRA.2025.3621942.

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Théophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. Goat-bench: A benchmark for multi-modal lifelong navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 16373–16383. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01549.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 3992–4003. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00371.

Jacob Krantz, Théophile Gervet, Karmesh Yadav, Austin S. Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 10882–10891. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01002.

Yijian Li, Changze Li, Hantian Shi, Jiaying Luo, Jiyuan Cai, Ming Yang, and Tong Qin. AgenticNav: Zero-Shot Vision-and-Language Navigation as a Tool-Calling Harness, 2026.

Peiqi Liu, Zhanqiu Guo, Mohit Warke, Soumith Chintala, Chris Paxton, Nur Muhammad (Mahi) Shafiullah, and Lerrel Pinto. Dynamem: Online dynamic spatio-semantic memory for open world mobile manipulation. In *IEEE International Conference on Robotics and Automation, ICRA 2025, Atlanta, GA, USA, May 19-23, 2025*, pp. 13346–13355. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11127619.

Xuetong Pei, Jian Liu, Vidura Munasinghe, Bo Miao, U-Xuan Tan, Wenrui Ding, and Na Zhao. SAP-Nav: Spatial Semantic Representation Meets Active Perception for Hierarchical Open-Vocabulary Object Navigation, 2026.

Dicong Qiu, Jiadi You, Zeying Gong, Ronghe Qiu, Hui Xiong, and Junwei Liang. SD-OVON: A Semantics-aware Dataset and Benchmark Generation Pipeline for Open-Vocabulary Object Navigation in Dynamic Scenes, 2025.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Shaoan Wang, Aocheng Luo, Fei Huang, Jingyi Xu, Xiaoyang Wang, Yueyu Wang, Qianli Ma, Fan Yang, Ran Mei, Jia Wei, Jiangpeng Hu, Xuhao Liu, Hongming Chen, Yuanbin Shao, Yiyang Lin, Ziliang Li, Liang Pan, Xinhang Liu, Yuntao Ma, and Tingxiang Fan. LightNav-0: Eliciting VLM Spatial Intelligence for Generalist Embodied Navigation, 2026a.

Xilin Wang, Guoxi Zhang, Hongming Xu, Zhuofan Zhang, Tianxu Wang, and Lifeng Fan. EvoNav-Bench: Benchmarking Lifelong Navigation in Evolving Environments, 2026b.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado (eds.), *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.077.

Karmesh Yadav, Jacob Krantz, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Jimmy Yang, Austin Wang, John Turner, Aaron Gokaslan, Vincent-Pierre Berges, Roozbeh Mottaghi, Oleksandr Maksymets, Angel X Chang, Manolis Savva, Alexander Clegg, Devendra Singh Chaplot, and Dhruv Batra. Habitat Challenge 2023. `https://aihabitat.org/challenge/2023/`, 2023.

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3d-mem: 3d scene memory for embodied exploration and reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 17294–17303. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01612.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

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. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 19057–19066. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01775.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: vision-language frontier maps for zero-shot semantic navigation. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 42–48. IEEE, 2024a. doi: 10.1109/ICRA57147.2024.10610712.

Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. HM3D-OVON: A dataset and benchmark for open-vocabulary object goal navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pp. 5543–5550. IEEE, 2024b. doi: 10.1109/IROS58592.2024.10802709.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks. In *Proceedings of Robotics: Science and Systems*, 2025a. doi: 10.15607/RSS.2025.XXI.013.

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*, (11):11530–11537, 2025b. doi: 10.1109/LRA.2025.3606388.

Xiangyu Zhao, Yicheng Chen, Shilin Xu, Xiangtai Li, Xinjiang Wang, Yining Li, and Haian Huang. An Open and Comprehensive Pipeline for Unified Object Grounding and Detection, 2024.

Haokun Zhu, Zongtai Li, Zhixuan Liu, Wenshan Wang, Ji Zhang, Jonathan Francis, and Jean Oh. STRIVE: Structured Representation Integrating VLM Reasoning for Efficient Object Navigation. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

A BENCHMARK CONSTRUCTION AND DATA GENERATION

The static and temporal benchmarks are generated separately. The static set contains 560 episodes; the temporal set contains 50 tracks with 12 evaluated relocations each, plus one initial state per track.

A.1 SCENE CONTEXT AND CANDIDATE OBJECT PLACEMENTS

The pipeline fuses RGB-D instances, canonicalizes receptacle labels, fits and refines support planes, scores object–support relevance, and settles YCB objects in Habitat-Sim. The runner uses semantic masks and an allowlist for receptacle selection; the LLM receives the receptacle label and nearby object labels, and returns region-conditioned category scores in $[0, 1]$ (the prompt contract is in Appendix B). Candidate objects are retained after support, stability, height, drift, and collision checks. The temporal release uses 35 scene manifests and 326 placements.

For each candidate plane, nearby object labels and the receptacle label are passed to the JSON LLM scorer. Candidate YCB objects are sampled on the plane, settled in Habitat-Sim, and exported only when the support and collision checks succeed.

A.2 STATIC OBJECTNAV EVALUATION SET

The static evaluation set contains 560 episodes from 35 HM3D scenes and 12 YCB categories. Goals are category-level and targets remain fixed within an episode. Table 4 reports its distribution.

Table 4: Category-level statistics for the archived static benchmark.

Goal category	Episodes	Goal records	Goal viewpoints	Scenes
Apple	121	43	516	21
Bowl	33	12	144	7
Cracker box	6	1	12	1
Cup	51	18	216	11
Marker	12	4	48	2
Mug	102	46	552	22
Mustard bottle	5	3	27	1
Plate	105	47	564	20
Rubik's cube	78	35	418	15
Sugar box	9	1	12	1
Tennis ball	6	1	12	1
Toy airplane	32	7	84	6
Total	560	218	2,605	35 unique

Goal viewpoints are sampled on the navmesh and require at least 48 visible target pixels. Starts share the goal floor and satisfy finite geodesic, geodesic-to-Euclidean ratio (≥ 1.05), and distance-bucket constraints. The archived buckets are 336 episodes below 5 m, 196 at 5–10 m, and 28 at 10–15 m. Figure 8 shows examples.

Static episodes are formed by collecting render-verified goal viewpoints for each eligible object category, sampling same-floor starts, and retaining the nearest reachable goal that satisfies the path-ratio and distance-bucket constraints.

A.3 TEMPORAL RELOCATION EXTENSION

The temporal set contains 50 persistent-object tracks across 35 scenes. Each track has an initial state $t00$ and 12 ordered relocations; object identity is preserved while pose and support may change. Destinations are selected from the placement pool using five rules: same-plane local shift, receptacle shift, room shift, local contrast, and long relocation. The scheduler selects an eligible unoccupied destination, settles and renders each state, and rejects invalid or non-visible transitions. Exported episodes require a visible goal view, a same-floor/navmesh-island start, and a valid path. The release has 600 relocation episodes and 650 rendered states; Figure 9 shows examples.

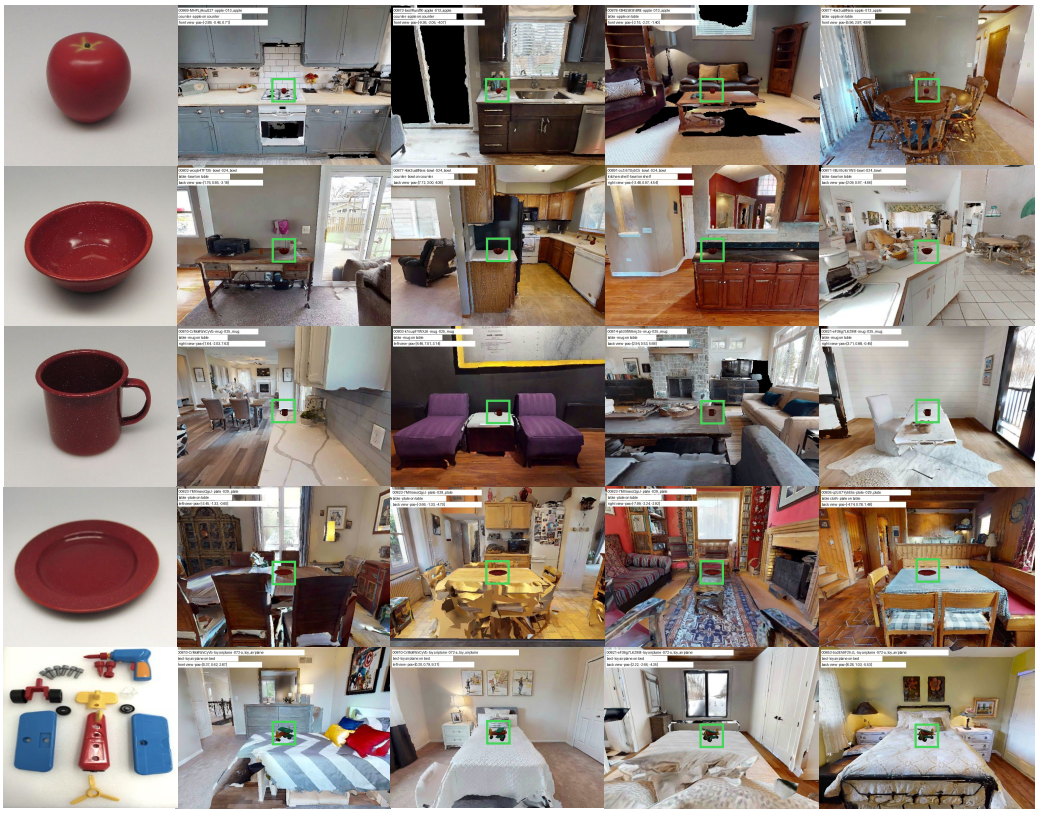

Figure 8: Representative episodes from the static ObjectNav evaluation set. Each example shows a fixed scene state with a YCB target object.

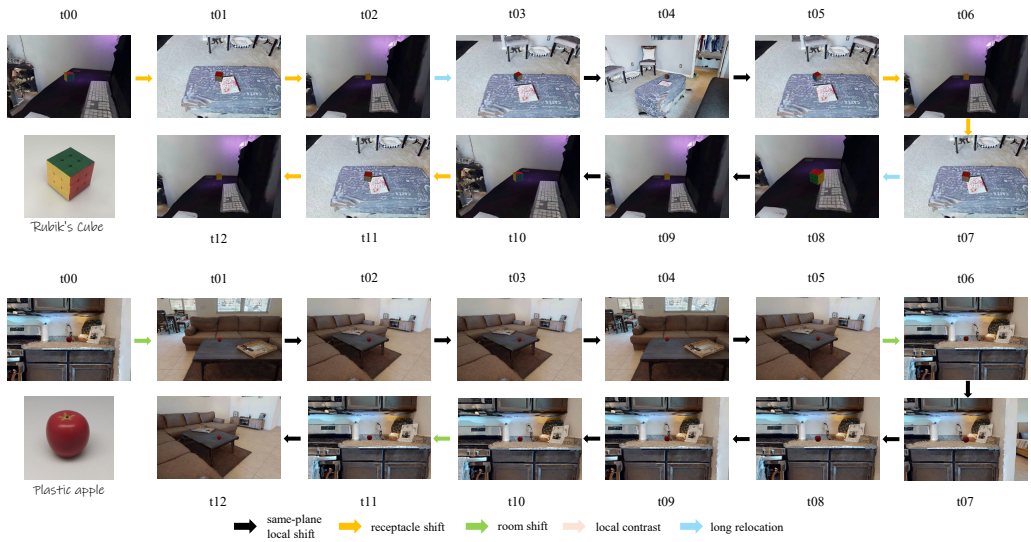

Figure 9: Representative states from a dynamic relocation track. The target object identity persists across states while its placement changes; time indices denote order only.

Dynamic tracks start from one persistent object and repeatedly select an eligible destination according to the relocation rules, while avoiding occupied supports when possible. Each state is rendered

Table 5: Scene 00839-zt1RVoi7PcG, floor 1: the Rubik's cube moves among bedroom desk/table support planes and local slots. LLM relevance scores guide candidate support selection; physics validation filters placements; sequence rules determine the temporal order.

Time	Actual move (support plane)	XZ (m)	Generation rule	LLM-generated change explanation
t01	desk 00001 → table 00012	7.20	receptacle shift	Update the remembered support type from desk to table.
t02	table 00012 → desk 00001	7.34	receptacle shift	Track the return to the desk rather than the previous table association.
t03	desk 00001 → table 00005	7.31	long relocation	Replace the old location belief after a long move.
t04	table 00005 → another slot on 00005	0.19	same-plane local shift	Update the precise XZ slot on the table.
t05	table 00005 → another slot on 00005	0.19	same-plane local shift	Distinguish nearby slots on the same support.
t06	table 00005 → desk 00001	7.31	receptacle shift	Update the support association from table to desk.
t07	desk 00001 → table 00012	7.34	receptacle shift	Track another support-type change to the table.
t08	table 00012 → desk 00001	7.29	long relocation	Revise the long-range belief instead of searching the old table.
t09	desk 00001 → another slot on 00001	0.33	same-plane local shift	Remember the new desk slot while room and support stay unchanged.
t10	desk 00001 → another slot on 00001	0.21	same-plane local shift	Continue updating the precise position on the desk.
t11	desk 00001 → table 00012	7.20	receptacle shift	Remember the table as the current support, not the earlier desk.
t12	table 00012 → desk 00001	7.34	receptacle shift	Correct the support-type belief after returning to the desk.

Table 6: Scene 00880-Nfvxx8J5NCo, floor 0: the apple moves between bedroom and dining-room tables and among local slots on each table. LLM relevance scores guide candidate support selection; physics validation filters placements; sequence rules determine the temporal order.

Time	Actual move (support plane)	XZ (m)	Generation rule	LLM-generated change explanation
t01	bedroom table 00002 → dining-room table 00003	4.98	room shift	Update the apple's room belief from bedroom to dining room.
t02	dining-room table 00003 → another slot on 00003	0.24	same-plane local shift	Track a new XZ slot on the same dining-room table.
t03	dining-room table 00003 → another slot on 00003	0.35	same-plane local shift	Continue tracking the local position in the dining room.
t04	dining-room table 00003 → another slot on 00003	0.24	same-plane local shift	Update the slot without changing room or support.
t05	dining-room table 00003 → another slot on 00003	0.24	same-plane local shift	Avoid treating a support plane as one immutable position.
t06	dining-room table 00003 → bedroom table 00002	4.86	room shift	Update the remembered room back to bedroom.
t07	bedroom table 00002 → another slot on 00002	0.35	same-plane local shift	Track the specific slot on the bedroom table.
t08	bedroom table 00002 → another slot on 00002	0.29	same-plane local shift	Continue tracking local movement on that table.
t09	bedroom table 00002 → another slot on 00002	0.17	same-plane local shift	Revise the exact XZ position despite the small move.
t10	bedroom table 00002 → another slot on 00002	0.20	same-plane local shift	Avoid confusing nearby slots on the bedroom table.
t11	bedroom table 00002 → dining-room table 00003	5.09	room shift	Update the room belief to dining room again.
t12	dining-room table 00003 → another slot on 00003	0.24	same-plane local shift	Remember the final slot on the dining-room table.

and retained only if the target is visible and its exported ObjectNav episode has a valid goal, start, and path. Tables 5 and 6 show two tracks.

B BENCHMARK CONSTRUCTION PROMPT TEMPLATES

The templates below cover vocabulary construction, region inference, object selection, and receptacle weighting. The vocabulary template is a reconstructed workflow specification, not an archived generation transcript. Capitalized fields denote supplied inputs. JSON structures are schematic: type placeholders indicate the required output types.

Table 7: Summary of the static and dynamic temporal Everyday-Object Navigation evaluation sets.

Static		**Dynamic Temporal**	
Quantity	Value	Quantity	Value
HM3D scenes	35	HM3D scenes	35
Episodes	560	Persistent-object sequences	50
YCB goal categories	12	Relocation episodes	600
Distinct goal records	218	Episodes per sequence	12
Valid goal viewpoints	2,605	Target categories	4

B.1 Region inference and object–receptacle relevance

B.1: Infer region and score object–receptacle relevance

Return only JSON with schema {"region": "room type", "scores": {"object_category": number_between_0_and_1}, "rationales": {"object_category": "short reason"}}. Infer the room region from nearby labels, then score how likely each object belongs on this receptacle in that room. Use everyday indoor commonsense and avoid giving high scores to objects that are physically or semantically unnatural for the receptacle.
Nearby object labels: NEARBY_LABELS
Receptacle label: RECEPTACLE_LABEL
Object categories: OBJECT_CATEGORIES

B.2 Scene-specific YCB object selection

B.2: Select YCB categories for a scene

You are constructing an ObjectNav benchmark scene. Task: choose 4-5 YCB object categories that are natural for this scene. Use only allowed_ycb_candidates. For the target release, the caller must prefilter excluded categories such as sponge and lego_duplo before constructing this list; this prompt does not perform asset exclusion. Prefer everyday small objects, semantic diversity, and compatibility with the listed room types and manually checked receptacle instances. Avoid objects that are unnatural for all listed receptacles. Select categories, not asset IDs; asset resolution is performed downstream by the placement code. Return exactly one JSON object with schema: {"scene_id": string, "selected_objects": [{"object_category": string, "reason": string, "preferred_room_types": [string], "preferred_receptacles": [string]}]} No markdown, no prose outside JSON.
Input: SCENE_ID, ALLOWED_YCB_CANDIDATES, ROOM_TYPE_COUNTS, RECEPTACLE_CANDIDATES

B.3 Scene-specific object–receptacle distributions

B.3: Assign scene-specific receptacle weights

You are constructing scene-specific object-receptacle relations. Task: for each selected object, assign a probability distribution over the listed receptacle candidate_id values. Use semantic compatibility with room type, receptacle type, and physics-settled evidence. Each probability should be non-negative and refer only to candidate_id values from the input. The implementation validates the IDs, removes invalid or non-positive entries, and normalizes the remaining values to sum to 1.0. Return exactly one JSON object with schema: {"scene_id": string, "object_distributions": [{"object_category": string, "distribution": [{"candidate_id": string, "probability": number, "rationale": string}]}] } No markdown, no prose outside JSON.
Input: SCENE_ID, SELECTED_OBJECTS, RECEPTACLE_CANDIDATES

The evaluated relocation sequences use twelve ordered time indices without time-of-day or day labels. Daily and long-term routines are not evaluated.

C SUPPORT-GUIDED EXPLORATION DETAILS

C.1 TARGET–SUPPORT RELEVANCE

A VLM proposes target–support relevance scores $\rho(g, c_s) \in [0, 1]$ for target g and support category c_s.

Fixed search weights include mug–table 1.0, mug–desk 0.9, mug–counter 0.8, mug–nightstand 0.6 and mug–shelf 0.4. Apple–table is 1.0, apple–counter 0.9, apple–desk 0.4 and apple–shelf 0.3. Omitted pairs for known targets receive zero. These are heuristic weights, not generation probabilities or quantities estimated from temporal events.

C.2 VIEWPOINT RANKING

We combine this relevance with travel distance and inspection feedback to rank candidate viewpoints:

$$v_t^* = \arg\max_{v \in \mathcal{F}_t} \left[-\|p_v - p_t\|_2 + \mathrm{clip}(b_t(v, g), -B, B) \right],$$

$$b_t(v, g) = \begin{cases} -\lambda_{\mathrm{I}}, & \text{near an inspected support,} \\ +\lambda_{\mathrm{P}}, & \text{otherwise, if another view is pending,} \\ \lambda_{\mathrm{R}} \max\limits_{s \in \mathcal{S}_t(v)} \rho(g, c_s), & \text{otherwise.} \end{cases} \tag{3}$$

Here, $\mathcal{F}_t$ contains the room's candidate frontier viewpoints, p_v and p_t are horizontal positions, and $\mathcal{S}_t(v)$ contains supports near v. The penalty λ_{I} discourages redundant inspection, the bonus λ_{P} encourages an additional view of a partially observed support, and λ_{R} weights the VLM-proposed relevance. The inspected-support penalty takes precedence over both bonuses. The bound B limits their effect on travel cost. When no support is nearby, the maximum is defined as zero and selection uses distance to continue discovering new supports.

D ADDITIONAL RESULTS

Table 8: Performance of the full system grouped by target history available before each dynamic episode. A valid record is a verified, non-conflicting target observation. It covers the current location if its recorded target position is within 0.50 m horizontally and 0.35 m vertically of the current target in the common memory frame. Ground truth is used only for this offline grouping. N denotes the number of episodes; SR and SPL are percentages.

Target-history state	N	SR (%) ↑	SPL (%) ↑
Latest valid record covers target	128	78.13	57.63
Only an older record covers target	94	79.79	54.75
History exists, current location uncovered	208	54.33	30.11
No valid target history	170	47.65	31.17

Performance under different history states. Table 8 partitions all 600 dynamic episodes according to the evidence available before navigation. When the latest valid record covers the current target, EONav achieves 78.13% SR and 57.63% SPL. In 94 episodes, only an older record covers the target; EONav achieves 79.79% SR and 54.75% SPL. This group identifies search conditions in which older records remain relevant, since the most recent record alone does not cover the current target. When history exists but leaves the current location uncovered, EONav achieves 54.33% SR; with no valid target history, it achieves 47.65%. These outcomes show that the framework can also complete searches when stored target locations are insufficient. This breakdown characterizes different search conditions; its groups contain different episodes and do not constitute a matched memory ablation.

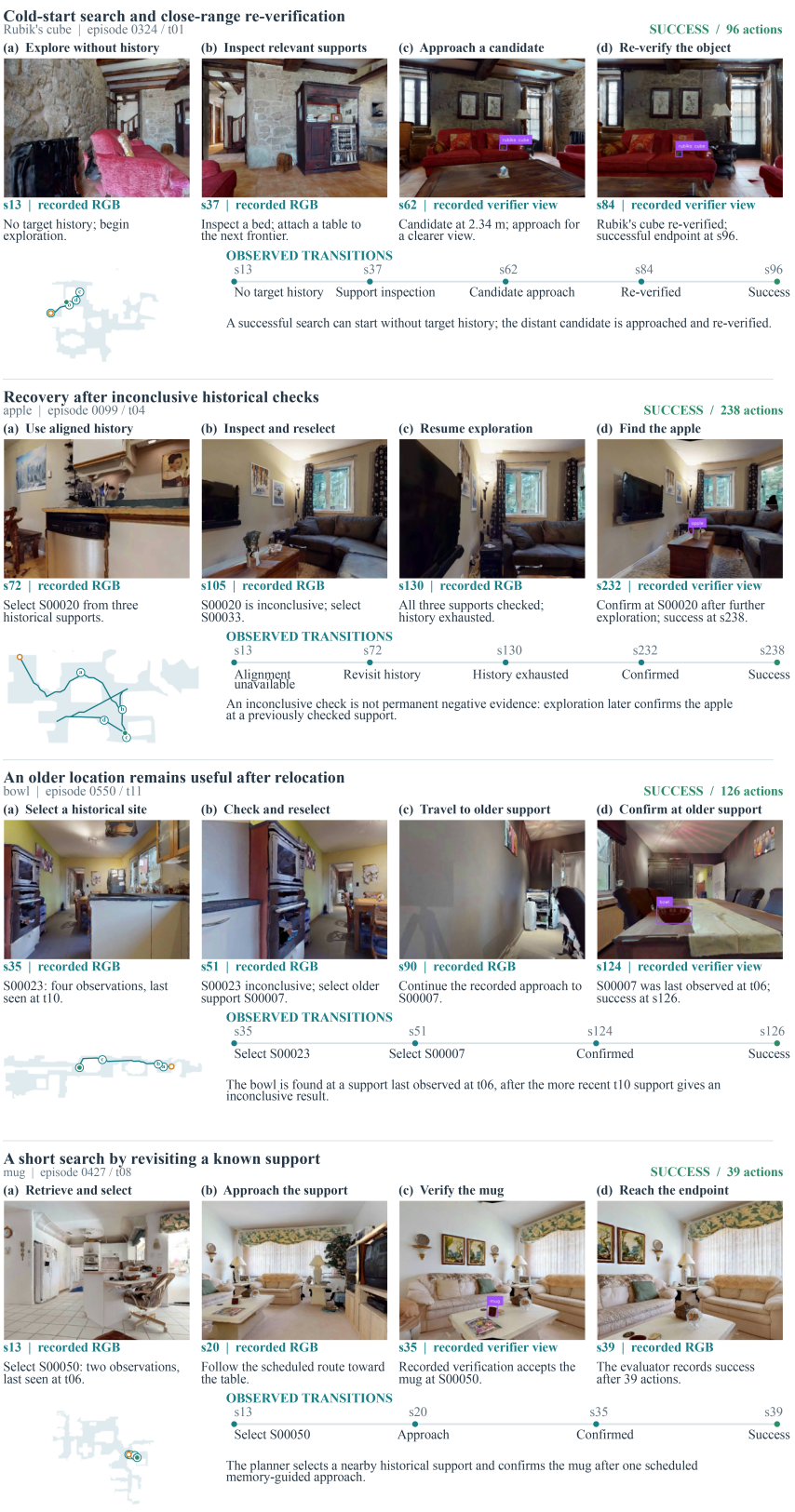

Figure 10: Qualitative example of successful cases in dynamic settings.



Figure 11: Qualitative example of failure cases in dynamic settings.