MARVISNAV: MAKING MEMORY VISIBLE ON ROUTE CHOICES FOR ZERO-SHOT OBJECT NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

When searching for an object, people choose their next move by considering both likely target locations and places already explored. The current view can cue place-associated memories, bringing target relevance and prior exploration into the same spatial context. In many zero-shot object navigation (ZSON) methods, however, vision-language models (VLMs) infer promising search areas from egocentric images, but exploration history enters the decision through separate representations, *e.g.,* as text or maps. This separation either requires an additional fusion step to align semantic evidence with exploration state, or leaves the correspondence between memory and route choices implicit for the VLM to recover. We instead make exploration memory visible directly on visual route choice. We propose MarvisNav, a ZSON framework that maintains a topological graph and projects candidate nodes together with their exploration states onto the egocentric perception as memory-bearing visual route choices. These topological states capture local exploration progress beyond binary visitation. By binding exploration state directly to each visual candidate, MarvisNav enables the VLM to jointly evaluate target relevance and exploration state and make the final high-level selection without a separate post-hoc fusion or reranking stage. Without policy training, MarvisNav achieves state-of-the-art performance on HM3D (81.2% SR and 42.5% SPL), while competitive on MP3D. Moreover, it outperforms representative VLM-based methods with far fewer VLM calls (*e.g.,* 7.5% of WMNav). Real-robot experiments across diverse scenes further validate its practical deployability. Beyond MarvisNav, our study shows that memory representation shapes VLM decisions and ZSON performance, highlighting that effective memory use in ZSON depends not only on its availability, but also on how it is represented.

1 INTRODUCTION

Imagine a robot asked to find an office chair in a home it has never seen. To succeed, it needs to infer where the target is likely to be and explore efficiently. This challenge is formalized as Object Navigation. Standard supervised methods are typically trained on a closed set of target categories (Chaplot et al., 2020; 2021; Ramakrishnan et al., 2022; Ye & Yang, 2021), limiting their ability to generalize to new objects. Zero-shot Object Navigation (ZSON) instead leverages pretrained semantic knowledge to search for novel targets without category-specific training (Majumdar et al., 2022; Chen et al., 2023; Gadre et al., 2023; Kuang et al., 2024), making it a key step toward open-world tasks.

ZSON is a long-horizon search problem under partial observability (Zhang et al., 2025d). When the target is not visible, the agent needs to use semantic cues to infer promising search areas. Existing methods obtain such guidance through visual-semantic alignment, which scores target relevance from visual observations (Gadre et al., 2023; Yokoyama et al., 2024; Zhang et al., 2025c), or contextual semantic reasoning, which exploits scene context and spatial relations (Zhou et al., 2023; Yin et al., 2024; Cao et al., 2025). Recently, generative Vision-Language Models (VLMs) have been widely adopted for object search, leveraging strong multimodal understanding and broad knowledge for richer semantic and spatial reasoning over observations. Long-horizon search, however, requires a sequence of high-level decisions. A VLM does not by itself maintain the agent's exploration state across successive decisions. VLM-based methods therefore maintain explicit memory (Nie et al., 2025; Padilla et al., 2026) to avoid re-selecting semantically promising but already searched areas.

Semantic guidance and exploration memory, however, are often met in separate representations. VLMs form semantic judgments from the current egocentric view, whereas exploration memory may

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.

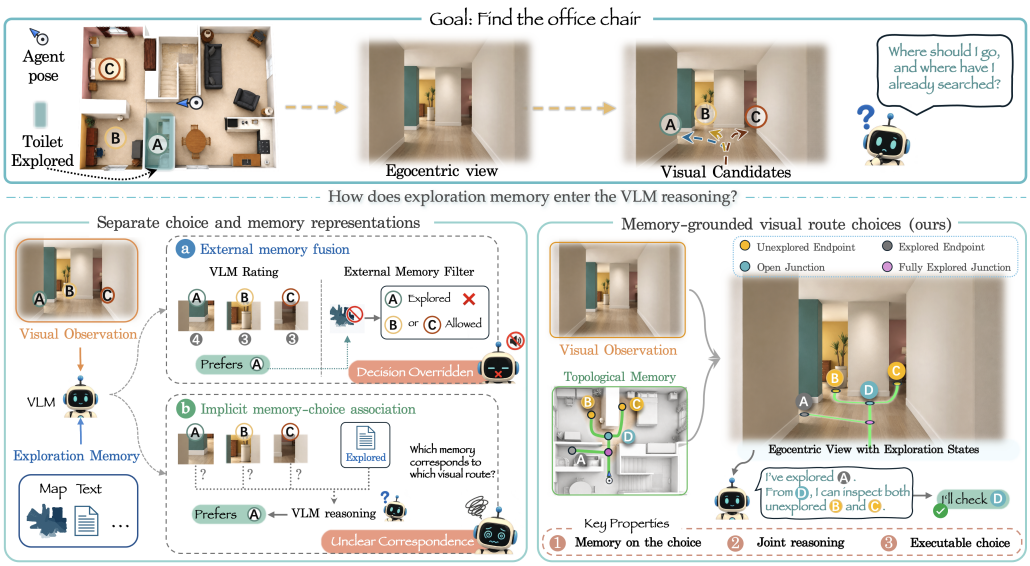


Figure 1: **Top:** Object search requires a VLM to consider both where the target might be found and which areas have already been searched. **Left:** With separate representations, memory either modifies the VLM's preference externally or leaves the VLM to infer its correspondence with the choices implicitly. **Right:** We instead ground topological exploration memory onto egocentric route choices, enabling joint reasoning over target relevance and exploration state. Junctions such as **D** are also valid choices. The selected choice directly specifies a navigation goal.

be carried as separate textual histories or dense maps. Bringing the two together therefore requires an additional fusion or alignment step before decision-making. When memory remains outside the VLM, the system must explicitly define how it changes the model's semantic judgment (Nie et al., 2025; Padilla et al., 2026), for example through hand-crafted candidate filtering or score adjustment. This places part of the choice outside the VLM and may constrain or override its judgment. When memory is instead supplied to the VLM, external fusion may be avoided, but the alignment problem remains. A textual history records past events in language, whereas a top-down map lays them out in an allocentric frame (Wu et al., 2026; Habibpour & Afghah, 2025a;b). Neither is expressed in the egocentric view the agent is reasoning over. The VLM must therefore infer how remembered places relate to the current view before using that memory to decide where to search next (Fig. 1, left).

Consider how a person searches for the chair in an unfamiliar house. They make both judgments from the same view, seeing which doorways look promising and which lead to areas already searched. A searched room is dismissed at a glance, while a doorway seen earlier remains a candidate if it still opens onto unexplored space. By making exploration cues visible on the corresponding spatial choices in views, we may then be able to give VLMs a similar decision interface, so that semantic evidence and exploration state are read together rather than aligned across representations.

To this end, we introduce **MarvisNav**, a ZSON framework that **ma**kes exploration memo**ry** **vis**ible to VLMs by grounding it directly on visual route choices. We first reduce continuous exploration into sparse high-level decisions, where the VLM selects among navigable search directions. We derive these choices from a Reduced Voronoi Graph (RVG) (Wu et al., 2024), where selected junctions serve as decision sites and local paths define candidate directions associated with reachable nodes. We extend this representation into a State-Aware RVG that associates each candidate with a local exploration state indicating whether nearby connected space may remain unexplored. At each decision, MarvisNav projects visible RVG paths into the egocentric RGB views and annotates candidates with local identifiers and exploration states, forming memory-bearing visual route choices (Fig. 1, right). A VLM then jointly evaluates target relevance and exploration state on these choices and selects an identifier in a single query. The selected identifier directly specifies the next navigation goal, enabling memory-conditioned route selection without additional semantic-memory fusion.

We evaluate MarvisNav on standard indoor ZSON benchmarks, where it achieves strong performance without policy training. It also outperforms representative VLM-based methods with far

fewer VLM queries, demonstrating practical efficiency for long-horizon navigation. We further validate its practical deployability on a Go2 robot across diverse scenes. Beyond MarvisNav, our study further demonstrates that exploration-memory representation can substantially affect both VLM preferences and navigation performance. These findings highlight the need to consider not only what an agent remembers, but also how that memory is made visible and associated with its choices.

2 Related Work

Semantic Guidance and Spatial Choices in ZSON. ZSON asks an agent to find an object in an unseen environment without category-specific training. A central problem in ZSON is determining where the target is likely to be found. Early approaches transfer pretrained vision-language knowledge to navigation through multimodal goal embeddings and open-vocabulary localization (Majumdar et al., 2022; Gadre et al., 2023). Later methods score the similarity between observations and the target with a pretrained image-text model (Radford et al., 2021) and accumulate it in maps to guide exploration toward promising areas (Yokoyama et al., 2024; Zhang et al., 2025c; Kuang et al., 2024). Large-language models (LLMs) extend this guidance beyond visual similarity by reasoning about object-room relations, scene context, and spatial structure (Zhou et al., 2023; Yin et al., 2024; Cao et al., 2025). VoroNav follows this line and uses a Voronoi Graph to organize navigable paths and sparse observation locations for language-model reasoning (Wu et al., 2024). More recently, generative VLMs have moved this reasoning onto the egocentric image, judging which directions look promising without an intermediate textual description (Nie et al., 2025; Gao et al., 2026). This shift requires spatial alternatives to be presented in a form that the VLM can visually inspect and select (Yang et al., 2023; Nasiriany et al., 2024), while remaining tied to navigation goals that the agent can execute. Recent works provide such an interface by rendering frontiers directly into the observation as selectable navigation goals (Padilla et al., 2026). However, these visual choices indicate where the agent can go, but not whether unexplored space remains beyond each option, preventing direct joint reasoning over semantic promise and exploration state for VLM.

Exploration Memory in VLM-Based Object Navigation. A VLM query does not retain exploration state across successive decisions, so VLM-based ZSON methods maintain explicit memory to avoid repeatedly considering already searched areas. Some methods keep memory outside the VLM's semantic reasoning: exploration history may restrict the candidates before querying VLMs (Gao et al., 2026; Padilla et al., 2026), or adjust its preference afterward through fixed filtering or score modification (Nie et al., 2025; Long et al., 2025). In both cases, semantic evidence and exploration state are evaluated separately and combined by an external rule, placing part of the decision outside the VLM and constraining its judgment. A more direct alternative is to expose the memory to the VLM itself. Text-based methods serialize past observations or actions into the prompt (Habibpour & Afghah, 2025a; Wu et al., 2026; Ji et al., 2025) for VLM reasoning. However, the VLM must implicitly associate this linguistic history with the current visual choices, which can make route-memory correspondence ambiguous. Rendering exploration history into a top-view map reduces that alignment problem by placing candidates and explored regions in a shared coordinate frame (Zhong et al., 2024). However, reasoning directly over such a map moves the VLM away from the egocentric visual domain in which its perception and semantic reasoning are most naturally trained (Nie et al., 2025). A further step is to bring memory into the VLM's egocentric view itself. Mem2Ego (Zhang et al., 2025b) projects candidate choices and visited locations into the view as separate markers. This preserves egocentric reasoning, but the correspondence between a candidate and its visitation memory must still be inferred from their image-space proximity. Moreover, a visited marker indicates where the agent has been, but not whether unexplored space remains beyond that location. Our work instead grounds topological exploration memory explicitly onto visual choices, indicating whether unexplored connected space remains beyond each choice. This allows the VLM to jointly reason over semantic evidence and exploration state, without recovering correspondences across representations or relying on external rules.

3 Motivation: How Memory Representation Shapes Decisions?

Before introducing our method, we conduct a motivating study of how the representation of exploration memory shapes VLM decisions in ZSON. Generative VLM-based methods choose where

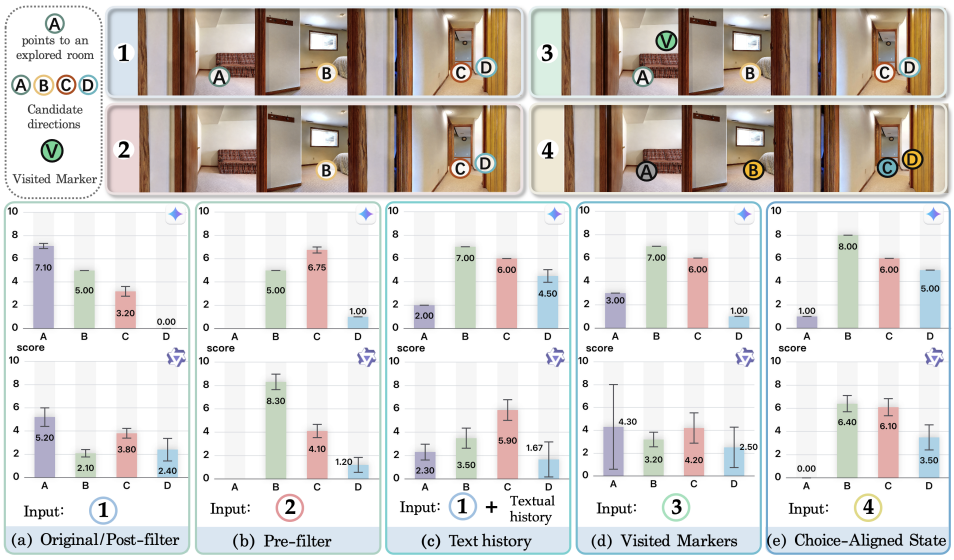


Figure 2: **Motivating comparison of memory representations.** Gemini-3.5-Flash and Qwen3.5-Plus score each candidate from 0-10 (higher is more worth exploring), with mean and standard deviation over 10 runs. Full prompts and representation-specific changes are detailed in Appendix D.

to explore by evaluating visual navigation choices, making memory representation part of the decision context. While prior methods represent exploration history in diverse ways, comparatively less attention has been paid to how the representation itself affects VLM decision-making. To study this effect in a controlled setting, we keep the scene and target fixed and vary only how exploration memory is represented, as illustrated in Fig. 2. The five representations span progressively tighter integration between exploration memory and the VLM's visual choices. **(a) Original / Post-filter** provides the reference VLM scores without exploration memory. It also represents the VLM's original judgment in post-filtering methods, where exploration memory is applied only after the VLM's evaluation (Nie et al., 2025). **(b) Pre-filtering** instead removes an explored candidate before VLM evaluation (Padilla et al., 2026). **(c) Textual History** provides prior navigation experience as linguistic context (Habibpour & Afghah, 2025a). **(d) Visited Markers** makes visitation memory visible in the egocentric view but represents it separately from the route candidates (Zhang et al., 2025b). **(e) Choice-Aligned State** is a simplified version of our method, which directly attaches the relevant exploration state to each candidate, making the memory-choice correspondence explicit.

The results show that changing the memory representation can substantially alter both route scores and rankings. The two external filtering schemes in (a,b) reveal two distinct effects. Under post-filtering, the explored candidate is removed only after VLM evaluation, leaving the final choice to the residual ranking without reconsidering the remaining routes. More counterintuitively, removing candidate *A* before evaluation changes the VLM's relative preferences among the remaining choices: Gemini switches its top choice from *B* to *C*, while Qwen switches from *C* to *B*. This suggests that VLM preferences are sensitive to the composition of the candidate set. Providing memory directly to the VLM introduces a different challenge: its correspondence with the current visual choices may remain implicit. With **Textual History** in (c), Qwen assigns the explored candidate *A* a higher mean score than the unvisited *D* (2.30 vs. 1.67), showing that textual history may not reliably translate into the intended route preference. **Visited Markers** in (d) projects visited locations into the view makes exploration memory more direct to the VLM, but the marker and candidate remain separate visual elements. Gemini likewise scores explored candidate *A* above unvisited *D*, while Qwen shows less stable and higher variance across repeated queries, resulting unreliable judgment. In contrast, **Choice-Aligned State** in (e) makes the memory-choice correspondence explicit by attaching exploration state directly to each candidate. Both models clearly separate the explored candidate *A* from the unexplored alternatives. Notably, Qwen produces more differentiated scores and a clearer ranking with choice-aligned states than with separate visitation markers, with explored A scored 0.

These observations show that the representation of exploration memory directly influences how a VLM evaluates navigation choices. Making memory available, or even visually visible, is not suf-

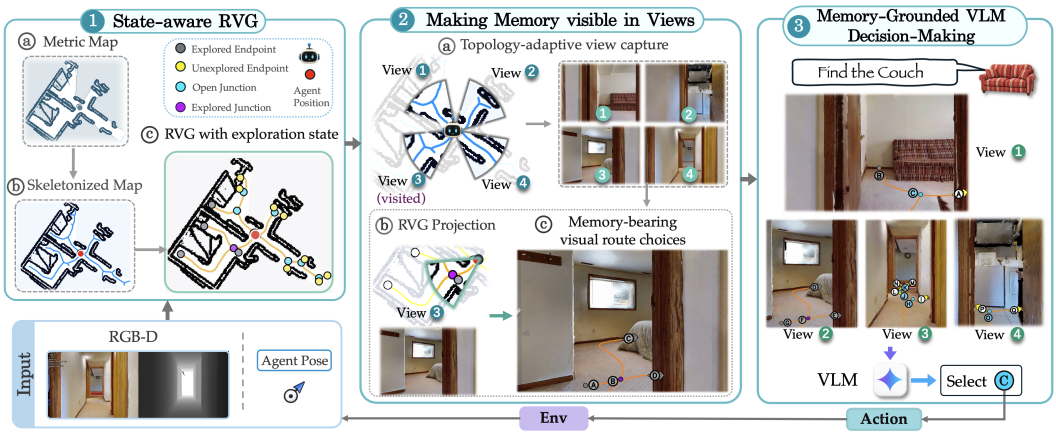


Figure 3: **Overview of MarvisNav.** (1) Given RGB-D observations and the agent pose, MarvisNav constructs a state-aware RVG by skeletonizing navigable space and assigning each node a local exploration state. (2) At each decision site, the local RVG topology determines the number and orientation of egocentric views. Visible paths and candidate nodes are projected into these views, where each candidate is annotated with a letter label and a color-coded exploration state. (3) Given the target and annotated views, a VLM jointly evaluates target relevance and exploration state. Its selected letter directly determines the next navigation goal.

ficient when its correspondence to the current route remains implicit. This motivates our MarviNav to make topological exploration memory both visible and choice-aligned, by grounding the relevant exploration state explicitly onto the egocentric route choice to which it applies. Rather than indicating only where the agent has been, each choice conveys whether useful unexplored space may remain beyond it, allowing the VLM to jointly reason about semantic promise and exploration state.

4 METHOD

Task formulation. We consider ZSON, in which the agent receives no category-specific navigation training for the target categories evaluated at test time. Each episode starts at pose $\mathbf{p}_0$ in an unknown scene with a language-specified target category c. At each step t, the agent receives its pose $\mathbf{p}_t$ and an egocentric RGB-D observation $\mathcal{O}_t = \{\mathbf{I}_t, \mathbf{D}_t\}$, where $\mathbf{I}_t$ and $\mathbf{D}_t$ are the RGB and depth images, and selects an action $\mathbf{a}_t$. An episode succeeds if the agent issues STOP at $t \leq T$ within distance d_s of an instance of c, where T is the episode step budget.

Overview. MarvisNav turns continuous exploration into sparse high-level decisions, as shown in Fig. 3. As the agent moves, it integrates depth observations into a global metric grid map and incrementally constructs a state-aware RVG whose junctions, endpoints, and connecting paths represent traversable space. Junctions serve as sparse decision sites. At each site, visible paths and their graph continuations define candidate routes, whose exploration states are determined by frontier association and local graph connectivity. MarvisNav projects the relevant paths and candidate nodes into topology-adaptive RGB views, annotating each candidate with a decision-local letter and a color that encodes its exploration state. These annotations form memory-bearing visual route choices that present scene semantics and exploration state together. The VLM selects a letter that maps to the corresponding RVG node and specifies a metric navigation goal for the classical navigation stack. After the goal is executed, new observations update the metric grid map, RVG, and exploration states, and the updated graph supplies the next decision site. This loop continues until the target is found or the step budget is exhausted.

4.1 STATE-AWARE REDUCED VORONOI GRAPH

MarvisNav maintains an online state-aware RVG that compresses observed free space into a sparse topology for route-level exploration. Following VoroNav (Wu et al., 2024), we first construct an RVG in which junctions provide decision sites and graph paths define route choices. We then aug-

ment its nodes with exploration states derived from frontier association and local graph connectivity, allowing the graph to capture local exploration progress beyond binary visitation.

RVG Construction. At step t, we integrate the depth image $\mathbf{D}_t$ using the associated pose $\mathbf{p}_t$ to maintain a global 2D grid map $\mathbf{M}_t$. Each grid cell is labeled as free, occupied, or unknown, and observations accumulated earlier in the episode are retained. Following ApexNav (Zhang et al., 2025c), we remove mapping artifacts and keep the free-space region that is reachable from the agent. We skeletonize this region and construct the RVG $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, where the vertex set $\mathcal{V}_t = \mathcal{V}_t^{\text{junc}} \cup \mathcal{V}_t^{\text{end}}$ contains junction and endpoint vertices, and the edge set $\mathcal{E}_t$ contains the traversable skeleton paths that connect them. We refer to the elements of $\mathcal{V}_t$ as RVG nodes throughout the paper. Skeleton points with degree one define endpoint nodes, whereas points with degree three or greater define junction nodes. We trace the paths between these vertices, merge nearby junctions, and prune short terminal branches. The graph is updated online as newly observed free space changes the map. We also derive a frontier mask from $\mathbf{M}_t$ by identifying the boundary between observed free space and unknown space. Associating this mask with the graph provides the frontier information used to determine each node's exploration state.

Topology-conditioned exploration state. Rather than representing exploration memory with a binary visited or unvisited label, we assign each RVG node a topology-conditioned state that summarizes its local exploration status. Visitation alone is insufficient because a visited junction may still expose an unexplored branch, while an unvisited endpoint may no longer offer a useful exploration direction. We define the exploration-state set as $\mathcal{S} = \{S_{\text{ue}}, S_{\text{ee}}, S_{\text{ij}}, S_{\text{lj}}\}$ and assign every node $v \in \mathcal{V}_t$ exactly one state through the mapping $\sigma_t : \mathcal{V}_t \to \mathcal{S}$. Augmenting the RVG with this mapping yields the state-aware graph $\widetilde{\mathcal{G}}_t = (\mathcal{V}_t, \mathcal{E}_t, \sigma_t)$. The four state values match the visual legend in Fig. 3. S_{ue} denotes an *unexplored endpoint* associated with an active frontier, whereas S_{ee} denotes an *explored endpoint* with no remaining frontier association. S_{ij} denotes an *open junction* with at least one directly connected node associated with an active frontier, whereas S_{lj} denotes an *explored junction* with no such neighbor. These states summarize exploration progress in the local graph neighborhood and provide the topological memory used for subsequent route decisions.

4.2 Grounding Topological Memory in Egocentric Views

Given the current state-aware RVG $\widetilde{\mathcal{G}}_t$, MarvisNav grounds topological exploration memory in the agent's egocentric observations. This involves topology-adaptive view capture, visibility-aware graph projection, and the construction of memory-bearing visual route choices.

Topology-adaptive view capture. Let d index the high-level decisions made by the VLM. At decision d, which occurs at the current navigation step t, MarvisNav captures RGB-D views oriented toward the relevant RVG paths around the current decision site. Paths with substantially overlapping viewing directions share one view. The number of captured views, denoted by K_d, therefore adapts to the local graph topology. We denote the RGB and depth images of view k by $\mathbf{I}_d^k$ and $\mathbf{D}_d^k$, respectively, where $k = 1, \ldots, K_d$. Unlike WMNav (Nie et al., 2025), which constructs a fixed six-view 360° panorama at every decision, MarvisNav captures only the views needed to cover the relevant route alternatives. For example, a single view is sufficient when only one new doorway is available beyond the incoming corridor. This topology-adaptive capture excludes walls and other irrelevant directions from the VLM input.

Visibility-aware RVG projection. For each captured view, we transform the current RVG nodes and sampled edge points from the world frame into the camera frame and project them using the camera intrinsics. A projected point is retained only when it falls inside the image and its projected depth agrees with the observed depth $\mathbf{D}_d^k$. This depth-consistency test removes graph geometry occluded by the observed scene. We overlay the retained path segments and nodes on $\mathbf{I}_d^k$, which grounds the local RVG structure in the visual scene. If a candidate node v_i is visible, we mark its projected location. If the node lies outside the image while part of its associated path remains visible, we place an arrow where that path exits the image. This arrow denotes the same candidate rather than introducing another one. Duplicate arrows are merged within a view, and an arrow is omitted when another view at the same decision already represents the corresponding route continuation.

Memory-bearing visual route choices. MarvisNav constructs each annotated RGB image $\widetilde{\mathbf{I}}_d^k$ from $\mathbf{I}_d^k$ by assigning every displayed candidate v_i a decision-local letter label $\ell_{d,i}$ and coloring its marker or arrow according to the current exploration state $\sigma_t(v_i)$. If the candidate appears in multiple views, it retains the same letter throughout decision d. These annotated images form memory-bearing visual route choices because the scene context, executable route, and corresponding exploration state appear together in the same egocentric view.

4.3 Memory-Conditioned VLM Decision and Goal Execution

Memory-conditioned VLM decision. At each decision site, the VLM receives the annotated images $\{\widetilde{\mathbf{I}}_d^k\}_{k=1}^{K_d}$ and a text prompt $q(c)$ instantiated with the target category c. The prompt explains the decision-local letter labels and state colors, instructs the VLM to balance target relevance against remaining exploration value, and specifies the required output. Prompt examples are provided in Appendix E. The VLM selects the letter associated with the route that is both semantically promising for the target and worth exploring, and it also reports whether the target may already be visible:

$$(\hat{\ell}_d, z_d) = \mathrm{VLM}\left(q(c), \{\widetilde{\mathbf{I}}_d^k\}_{k=1}^{K_d}\right), \tag{1}$$

where $\hat{\ell}_d$ is the selected letter and the scalar $z_d \in \{0, 1\}$ indicates whether the target may be visible. For every choice, the surrounding RGB content provides evidence about where the target may be, while the marker color conveys the exploration state of the corresponding route. Their spatial alignment allows the VLM to use both sources of evidence in one reasoning step, without a separate rule that combines or reranks semantic and memory scores afterward.

Goal execution and target approach. The system resolves $\hat{\ell}_d$ through the decision-local label mapping to its candidate node v_i and metric coordinate $\mathbf{x}(v_i)$. If no target detection is accepted, this coordinate becomes the next goal for the classical navigation stack, which handles path planning, collision avoidance, and low-level control. A separate detector processes incoming observations throughout exploration. A detection above its standard confidence threshold causes the agent to leave route-level exploration and approach the detected target. When $z_d = 1$, MarvisNav lowers the detector threshold for the views captured at that decision so that a weaker detection supported by the VLM can also initiate target approach. If no detection is accepted, the agent continues toward the selected RVG goal. The resulting observations update $\mathbf{M}_t$, $\widetilde{\mathcal{G}}_t$, and the node states until the agent reaches the target and issues STOP, or $t = T$.

5 Experiment

Our experiments systematically assessed the proposed MarvisNav around the following questions:

- How does MarvisNav perform against existing ZSON methods? (5.1)
- How do different memory representations affect navigation performance? (5.2)
- Does MarvisNav remain robust across VLM backbones, and at what query cost? (5.3)
- Can MarvisNav perform object searches on real robots without task-specific training? (B)

We report Success Rate (SR,%) and Success weighted by Path Length (SPL,%) under the standard ObjectNav evaluation protocol. SR measures the fraction of episodes completed successfully, while SPL additionally penalizes successful trajectories that are longer than the shortest optimal path. Due to space constraints, we report the real-robot experiments in Appendix B.

5.1 Comparison with Baselines

Table 1 compares MarvisNav with representative policy-based, embedding-based, LLM-based, and generative-VLM-based methods. We list trained policies for reference. Among the compared methods, recent generative-VLM approaches form the strongest baselines. Since prior methods use different VLMs reported in their own papers, Sec. 5.3 also compares representative methods under a shared backbone. MarvisNav performs particularly well on HM3D v0.2, achieving 81.2% SR

Table 1: **Comparison with ObjectNav baselines** on HM3D v0.1, HM3D v0.2, and MP3D. ZS denotes zero-shot ObjectNav, without task-specific fine-tuning. Results are taken from the original papers when available; otherwise, we use reproductions reported in (Zhang et al., 2025c): HM3D-v0.2(L3MVN, VLFM, and SG-Nav), and MP3D (L3MVN and OpenFMNav).

			HM3D v0.1		HM3D v0.2		MP3D	
Reasoning	Method	ZS	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
Policy	PONI (Ramakrishnan et al., 2022)	×	–	–	–	–	31.8	12.1
	Uni-NaVid (Zhang et al., 2025a)	×	–	–	73.7	37.1	–	–
Embedding similarity-based guidance	ZSON (Majumdar et al., 2022)	√	25.5	12.6	–	–	15.3	4.8
	CoW (Gadre et al., 2023)	√	–	–	–	–	7.4	3.7
	VLFM (Yokoyama et al., 2024)	√	52.5	30.4	63.6	32.5	36.4	17.5
	OpenFMNav (Kuang et al., 2024)	√	54.9	24.4	–	–	37.2	15.7
	ApexNav (Zhang et al., 2025c)	√	59.6	33.0	76.2	38.0	39.2	17.8
LLM-Based Semantic Reasoning	ESC (Zhou et al., 2023)	√	39.2	22.3	–	–	28.7	14.2
	L3MVN (Yu et al., 2023)	√	50.4	23.1	36.3	15.7	34.9	14.5
	VoroNav (Wu et al., 2024)	√	42.0	26.0	–	–	–	–
	SG-Nav (Yin et al., 2024)	√	54.0	24.9	49.6	25.5	40.2	16.0
	UniGoal (Yin et al., 2025)	√	54.5	25.1	–	–	41.0	16.4
Generative VLM-Based Visual Reasoning	TopV-Nav (Zhong et al., 2024)	√	52.0	28.6	–	–	35.2	16.4
	His-aug (Habibpour & Afghah, 2025a)	√	46.0	24.8	–	–	–	–
	InstructNav (Long et al., 2025)	√	–	–	58.0	20.9	–	–
	WMNav (Nie et al., 2025)	√	58.1	31.2	72.2	33.3	45.4	17.2
	DRIVE-Nav (Gao et al., 2026)	√	55.5	35.9	72.4	41.3	41.8	22.6
	RememNav (Wu et al., 2026)	√	60.0	**36.8**	67.8	36.6	**49.8**	**24.3**
	OpenFrontier (Padilla et al., 2026)	√	–	–	77.3	35.6	40.7	17.8
	MarvisNav (ours)	√	**60.5**	36.2	**81.2**	**42.5**	42.1	20.4

and 42.5% SPL. These exceed the strongest reported baseline results for each metric by 3.9 percentage points in SR over OpenFrontier and 1.2 points in SPL over DRIVE-Nav. On HM3D v0.1, MarvisNav obtains the highest SR among the compared methods, while its SPL remains close to the best result. It also outperforms Uni-NaVid, which is trained on millions of navigation samples, by 7.5 percentage points in SR. Performance on MP3D is competitive but not uniformly leading. MarvisNav achieves higher SPL but lower SR than WMNav, while ReMemNav outperforms it on both metrics. Overall, these results show that competitive navigation can be achieved with direct, memory-conditioned VLM selection, without an additional semantic-memory fusion stage.

5.2 Ablation Study on Memory Representation

Table 2: Ablation of memory representations.

Representation	To VLM	SR (%) ↑	SPL (%) ↑
No Memory	–	71.9	35.4
Post-filtering	×	74.3	37.1
Pre-filtering	×	75.0	37.2
Textual History	√	73.6	36.6
Visited Markers	√	76.5	40.6
Choice-aligned	√	**81.2**	**42.5**

To evaluate how memory representation affects navigation, we compare six MarvisNav variants on HM3D v0.2 with Gemini-3.5-Flash and the remaining navigation pipeline fixed. All variants share RVG-based candidate generation and egocentric projection of paths and nodes, varying only memory cues and filtering. *No Memory* keeps candidate colors unchanged regardless of exploration status. *Post-filtering* uses this display and applies exploration-based filtering after VLM evaluation, whereas *Pre-filtering* omits explored candidates from the visual overlay before evaluation. *Textual History* supplements the *No Memory* views with summaries of the ten most recent node visits. *Visited Markers* adds green markers beside visited visual candidates, leaving the original candidate markers unchanged. Our *Choice-aligned* representation instead encodes topology-conditioned exploration states directly in candidate markers.

As shown in Table 2, *Choice-aligned* achieves the highest SR and SPL at 81.2% and 42.5%, exceeding *Visited Markers*, the strongest alternative, by 4.7 and 1.9 percentage points, respectively. All memory variants improve over *No Memory*, but their benefits differ. In particular, *Textual History*

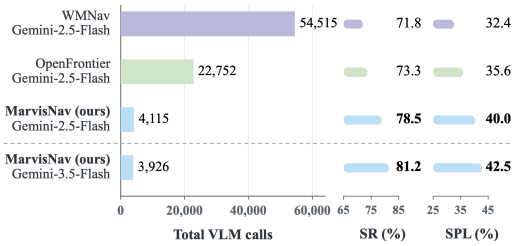
underperforms both external filtering variants, indicating that supplying history directly to the VLM does not necessarily yield greater navigation gains. *Visited Markers* performs better, but presenting topology-conditioned states directly on the candidates provides further improvements. These results support the combined use of topological exploration state and explicit memory-choice alignment, highlighting that effective navigation depends not merely on having memory, but also on how it is represented for current choices.

5.3 VLM Backbone Ablation and Query Efficiency

Table 3: Ablation of VLMs on HM3D v0.2.

VLM	SR (%) ↑	SPL (%) ↑
Qwen3-VL-8B	76.0	38.3
Qwen3.5-9B	76.5	37.8
Qwen3.7-plus	77.3	39.3
Gemma4-26B	78.9	38.5
Gemini-2.5-Flash	78.5	40.0
Gemini-3.5-Flash	**81.2**	**42.5**

Figure 4: VLM query efficiency on HM3D v0.2.



To assess sensitivity to VLM choice, we evaluate MarvisNav with five VLMs on HM3D v0.2 while keeping the rest of the pipeline fixed. As shown in Table 3, MarvisNav maintains strong performance across VLMs: even with lighter Qwen3-VL-8B, it achieves 76.0% SR and 38.3% SPL. To separate framework gains from VLM strength, we further compare methods using the same Gemini-2.5-Flash backbone. MarvisNav still outperforms WMNav (Nie et al., 2025) and OpenFrontier (Padilla et al., 2026) (Fig. 4), indicating that the gains are not attributable to VLM strength alone, but also to how MarvisNav represents exploration memory and integrates it into the navigation pipeline. Using Gemini-3.5-Flash further improves performance to 81.2% SR and 42.5% SPL. Overall, MarvisNav retains strong performance with weaker VLMs while continuing to benefit from more capable backbones.

To assess the efficiency of our approach, we compare total VLM queries rather than runtime, as response latency varies with serving conditions. This is important for real-world deployment of VLM-based ZSON, where frequent queries increase navigation waiting time. We reproduce WMNav and OpenFrontier, two representative VLM-based ZSON methods with open-source official implementations, and evaluate all methods using Gemini-2.5-Flash on the same $1{,}000$ HM3D v0.2 episodes. As shown in Fig. 4, MarvisNav requires $4{,}115$ VLM calls, only 7.5% and 18.1% of those used by WMNav ($54{,}515$) and OpenFrontier ($22{,}752$), respectively, while achieving higher SR and SPL. With Gemini-3.5-Flash, it falls further to $3{,}926$ and success rate reaches 81.2%. Notably, higher query frequency does not translate into better navigation performance in this controlled comparison. This query efficiency is consistent with MarvisNav's design: sparse decision sites limit when the VLM is queried, while choice-aligned exploration memory supports more effective decisions.

6 Conclusion

We introduced MarvisNav, a framework for VLM-based ZSON that makes topological exploration memory visible on egocentric route choices. A state-aware reduced Voronoi graph captures local exploration progress beyond binary visitation and organizes navigation into sparse high-level decisions. By grounding exploration states directly on visual candidates, MarvisNav enables the VLM to jointly evaluate target relevance, exploration state, and make the final route selection without post-hoc semantic-memory fusion. Experiments on HM3D and MP3D demonstrate strong navigation performance without task-specific policy training, while comparisons under a shared VLM backbone show better performance with substantially fewer queries than representative baselines. Controlled comparisons and navigation ablations support the benefits of combining topological exploration states with explicit memory-choice alignment. Real-robot experiments further demonstrate closed-loop object search in the physical world, helping to bridge the sim-to-real gap. Taken together, these findings show that exploration memory is most effective for VLM-based navigation when its state is clearly represented and directly aligned with the agent's available choices.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experimental design, and to assist with the discussion and presentation of experimental results. We additionally used generative AI for brainstorming, literature search and summarization, language editing, debugging, code refinement, and the refinement of figures. We did not use generative AI to generate experimental data, formulate mathematical claims or proofs, or autonomously conduct experiments. All AI-assisted suggestions were critically reviewed by the authors: methodological and experimental decisions were validated through our own implementation and experiments, literature suggestions were checked against the original sources, and all AI-assisted text and figures were manually reviewed and revised. All AI-assisted code was manually reviewed, tested for correctness, and validated through experiments by the authors. We take responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the algorithmic and implementation details required to reproduce MarvisNav in the main paper and Appendix, including map and RVG construction, exploration-state assignment, visual route grounding, VLM-based decision-making, and navigation execution. The Appendix further documents the exact VLM prompts, model configurations, inference settings, and experimental protocols used in our evaluations and controlled comparisons. We specify the benchmark splits, sensor and agent configurations, evaluation metrics, and the provenance of baseline results. For the HM3D v0.2 benchmark audit, we additionally provide the scene and episode IDs of the identified problematic episodes together with their issue categories to enable independent verification and reproducible evaluation. Upon publication, we commit to release the full source code, configuration files, and evaluation scripts. We will also release the reconstructed 3D model used solely for visualization as a reusable asset for future visualization and qualitative analysis.

REFERENCES

Diogo Almeida. Introducing System One Models & Jev. TypeSafe AI Blog, September 2026. URL https://typesafe.ai/blog/introducing-system-one-models-and-jev. Published September 15, 2026. Accessed September 26, 2026.

Yihan Cao, Jiazhao Zhang, Zhinan Yu, Shuzhen Liu, Zheng Qin, Qin Zou, Bo Du, and Kai Xu. Cognav: Cognitive process modeling for object goal navigation with llms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9550–9560, 2025. doi: 10.1109/ICCV51701.2025.00891.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e0982cbc81401df3430ee1ff780dc7a2-Paper-Conference.pdf.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pp. 667–676, 2017. doi: 10.1109/3DV.2017.00081.

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, 2020.

Devendra Singh Chaplot, Murtaza Dalal, Saurabh Gupta, Jitendra Malik, and Russ R Salakhutdinov. Seal: Self-supervised embodied active learning using exploration and 3d consistency. *Advances in neural information processing systems*, 34:13086–13098, 2021.

Junting Chen, Guohao Li, Suryansh Kumar, Bernard Ghanem, and Fisher Yu. How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers. *arXiv preprint arXiv:2305.16925*, 2023.

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, 2023. doi: 10.1109/CVPR52729.2023.02219.

Maoguo Gao, Zejun Zhu, Zhiming Sun, Zhengwei Ma, Longze Yuan, Zhongjing Ma, Zhigang Gao, Jinhui Zhang, and Suli Zou. Drive-nav: Directional reasoning, inspection, and verification for efficient open-vocabulary navigation. *IEEE Robotics and Automation Letters*, 11(10):11158–11165, 2026. doi: 10.1109/LRA.2026.3723310.

Mobin Habibpour and Fatemeh Afghah. History-augmented vision-language models for frontier-based zero-shot object navigation. In *2025 59th Asilomar Conference on Signals, Systems, and Computers*, pp. 280–284, 2025a. doi: 10.1109/IEEECONF67917.2025.11443839.

Mobin Habibpour and Fatemeh Afghah. Think, remember, navigate: Zero-shot object-goal navigation with vlm-powered reasoning. *arXiv preprint arXiv:2511.08942*, 2025b.

Zihe Ji, Huangxuan Lin, and Yue Gao. Dynavlm: Zero-shot vision-language navigation system with dynamic viewpoints and self-refining graph memory. *arXiv preprint arXiv:2506.15096*, 2025.

Glenn Jocher, Jing Qiu, Mengyu Liu, Shuai Lyu, Fatih Cagatay Akyon, and Muhammet Esat Kalfaoglu. Ultralytics yolo26: unified real-time end-to-end vision models. *arXiv preprint arXiv:2606.03748*, 2026.

Yuxuan Kuang, Hai Lin, and Meng Jiang. OpenFMNav: Towards open-set zero-shot object navigation via vision-language foundation models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 338–351, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.24. URL https://aclanthology.org/2024.findings-naacl.24/.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/long25b.html.

Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32340–32352. Curran Associates, Inc., 2022. doi: 10.52202/068431-2343. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/d0b8f0c8f79d3a621af945cafb669f4b-Paper-Conference.pdf.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, Quan Vuong, Tingnan Zhang, Tsang-Wei Edward Lee, Kuang-Huei Lee, Peng Xu, Sean Kirmani, Yuke Zhu, Andy Zeng, Karol Hausman, Nicolas Heess, Chelsea Finn, Sergey Levine, and Brian Ichter. PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37321–37341. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/nasiriany24a.html.

Dujun Nie, Xianda Guo, Yiqun Duan, Ruijun Zhang, and Long Chen. Wmnav: Integrating vision-language models into world models for object goal navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2392–2399. IEEE, 2025.

Esteban Padilla, Boyang Sun, Marc Pollefeys, and Hermann Blum. Openfrontier: General navigation with visual-language grounded frontiers. In *Robotics: Science and Systems (RSS)*, 2026.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile Gervet, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 15306–15336, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/430894999584d0bd358611e2ecf00b15-Paper-Conference.pdf.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=-v4OuqNs5P.

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, 2022. doi: 10.1109/CVPR52688.2022.01832.

Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. Rf-detr: Real-time detection transformer. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2511.09554.

Jincheng Wang, Lingfan Bao, Tong Yang, Diego Martinez Plasencia, Jianhao Jiao, and Dimitrios Kanoulas. Sand-planner: Sample-efficient diffusion planner in b-spline space for robust local navigation. In *Robotics: Science and Systems (RSS)*, 2026.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. *Robotics: Science and Systems*, 2024.

Feng Wu, Wei Zuo, Wenliang Yang, Jun Xiao, Yang Liu, and Xinhua Zeng. Rememnav: A rethinking and memory-augmented framework for zero-shot object navigation. *arXiv preprint arXiv:2603.26788*, 2026.

Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. VoroNav: Voronoi-based zero-shot object navigation with large language model. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53757–53775. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wu24u.html.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4927–4936, 2023. doi: 10.1109/CVPR52729.2023.00477.

Fan Yang, Per Frivik, David Hoeller, Chen Wang, Cesar Cadena, and Marco Hutter. Spatially-enhanced recurrent memory for long-range mapless navigation via end-to-end reinforcement learning. *The International Journal of Robotics Research*, 45(11):1682–1706, 2026. doi: 10.1177/02783649251401926. URL https://doi.org/10.1177/02783649251401926.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

Xin Ye and Yezhou Yang. Efficient robotic object search via hiem: Hierarchical policy learning with intrinsic-extrinsic modeling. *IEEE Robotics and Automation Letters*, 6(3):4425–4432, 2021. doi: 10.1109/LRA.2021.3068906.

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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 5285–5307. Curran Associates, Inc., 2024. doi: 10.52202/079017-0171. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/098491b37deebbe6c007e69815729e09-Paper-Conference.pdf.

Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19057–19066, 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 *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42–48, 2024. doi: 10.1109/ICRA57147.2024.10610712.

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, 2023. doi: 10.1109/IROS55552.2023.10342512.

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. *Robotics: Science and Systems*, 2025a.

Lingfeng Zhang, Yuecheng Liu, Zhanguang Zhang, Matin Aghaei, Yaochen Hu, Hongjian Gu, Mohammad Ali Alomrani, David Gamaliel Arcos Bravo, Raika Karimi, Atia Hamidizadeh, et al. Mem2ego: Empowering vision-language models with global-to-ego memory for long-horizon embodied navigation. *arXiv preprint arXiv:2502.14254*, 2025b.

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, 2025c. doi: 10.1109/LRA.2025.3606388.

Sixian Zhang, Xinyao Yu, Xinhang Song, Yiyao Wang, and Shuqiang Jiang. Function-centric bayesian network for zero-shot object goal navigation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19535–19545, 2025d. doi: 10.1109/ICCV51701.2025.01816.

Jiamin Zheng, Jingwen Yu, Guangcheng Chen, and Hong Zhang. Enhancing glass surface reconstruction via depth prior for robot navigation. *arXiv preprint arXiv:2604.18336*, 2026.

Linqing Zhong, Chen Gao, Zihan Ding, Yue Liao, Huimin Ma, Shifeng Zhang, Xu Zhou, and Si Liu. Topv-nav: Unlocking the top-view spatial reasoning potential of mllm for zero-shot object navigation. *arXiv preprint arXiv:2411.16425*, 2024.

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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42829–42842. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zhou23r.html.

A Benchmark Evaluation Setup

Evaluation Protocol and Implementation. We follow the official protocol without modification, including 640×480 RGB-D observations, a 79° horizontal field of view, and a 500-step limit. The agent operates in the discrete action space $\{\textsc{Stop}, \textsc{MoveForward}, \textsc{TurnLeft}, \textsc{TurnRight}, \textsc{LookUp}, \textsc{LookDown}\}$, with a forward step size of $0.25\,\mathrm{m}$ and a rotation increment of 30°. We adopt the low-level planning and target-approach modules following ApexNav (Zhang et al., 2025c). For target perception, we use RF-DETR (Robinson et al., 2026) and SAM3 (Carion et al., 2026). The main experiments use Gemini 3.5 Flash for vision-language reasoning. All simulation experiments are run on a desktop equipped with a single NVIDIA GeForce RTX 4090 GPU.

Benchmarks. We evaluate MarvisNav on three Habitat (Puig et al., 2024) ObjectNav benchmarks constructed from real-world 3D scans: HM3D-Semantics v0.1, HM3D-Semantics v0.2, and Matterport3D (MP3D) (Ramakrishnan et al., 2021; Yadav et al., 2023; Chang et al., 2017). HM3D v0.1, used in the Habitat 2022 ObjectNav Challenge, contains 2,000 validation episodes across 20 scenes and six target categories. HM3D v0.2 expands and corrects the semantic annotations, with 1,000 validation episodes across 36 scenes and the same six categories. MP3D provides 2,195 validation episodes across 11 scenes and 21 categories. In total, we evaluate on 5,195 episodes.

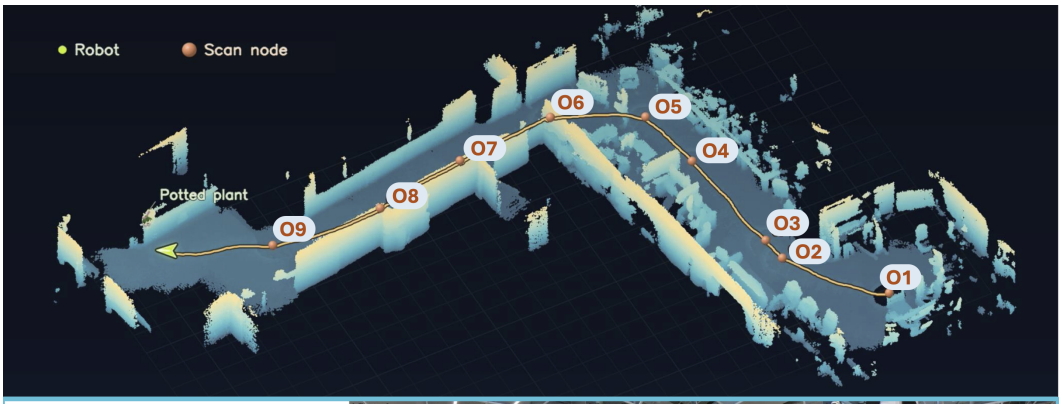
B Real-World Deployment

Robot platform and navigation setup. We deploy MarvisNav on a Unitree Go2 to demonstrate closed-loop object search beyond simulation. An Odin1 sensing module provides RGB-D observations and pose estimates, with the camera mounted $0.45\,\mathrm{m}$ above the ground. We limit the depth observation range to $4\,\mathrm{m}$ for both mapping and local navigation. All MarvisNav components except VLM inference run onboard an NVIDIA Jetson AGX Orin with Ubuntu 20.04 and ROS Noetic. Target detection and instance segmentation use YOLO26m-seg (Jocher et al., 2026), while Gemini-3.5-Flash is accessed through a remote API for high-level route selection. We retain MarvisNav's topological memory and visual decision pipeline, replacing the classical map-based planner with goal-conditioned visual navigation (Yang et al., 2026; Wang et al., 2026). Specifically, we use the pretrained SRU (Yang et al., 2026) policy without retraining or fine-tuning to navigate toward the metric goals specified by the VLM-selected route identifiers.

Deployment protocol and demonstrations. We deploy MarvisNav on a Unitree Go2 across diverse real-world environments, including a large office, a home, and outdoor settings, with runs covering daytime and nighttime conditions. Each run starts with an empty map and only a target category, without prior knowledge of the scene layout or target location. The robot performs online mapping, route selection, and navigation without human intervention. Full indoor and outdoor deployment videos are shown in the supplementary material.

Figure 5 presents a representative indoor run in which the robot searches for a *potted plant in the reception* in a large open-plan office. The robot travels $39.2\,\mathrm{m}$ and records a mapped area of $238.6\,\mathrm{m}^2$. The run includes nine high-level decision and scan sites, O1–O9, with one VLM query per site. At O1, MarvisNav captures six egocentric views for panoramic coverage of the surroundings before route selection. The VLM combines the instruction's reception-area cue with visible office and corridor features to guide context-aware route selection. During exploration, reaching the selected node in the State-Aware RVG triggers the next scan and decision. The plant first becomes visible at O9, after which the robot approaches the target and stops automatically upon arrival.

The point-cloud overview at the top of Fig. 5 shows the actual robot trajectory, while Fig. 6 presents the final State-Aware RVG. Both maps are constructed from observations collected during navigation. The State-Aware RVG represents observed free space through junctions, boundary nodes, and connecting paths, with node states capturing local exploration progress. Some projected path segments on egocentric images appear discontinuous because of depth noise, occlusion filtering, and the $4\,\mathrm{m}$ observation range. The robot nevertheless completes the search, demonstrating closed-loop route selection and execution despite these local projection gaps. The trajectory overlaid in red on the State-Aware RVG shows how the selected visual route choices translate into physical navigation.



O1 **B**

target_visible: no **target_in_view: none**
chosen_view: view 1

scene: Robot laboratory/testing hall, with the front-right direction leading toward an open office or reception area.

why: No plant is currently visible. On the right side of View 1, extending toward View 6, there is an open passage leading deeper into the space, where furniture resembling office desks or a white reception counter is faintly visible. B is a cyan junction located along the main route toward this area, making it a suitable point from which to inspect the reception region.



O2 **D**

target_visible: no **target_in_view: none**
chosen_view: view 1

scene: Office/laboratory corridor, with a long hallway ahead leading toward a more open office area.

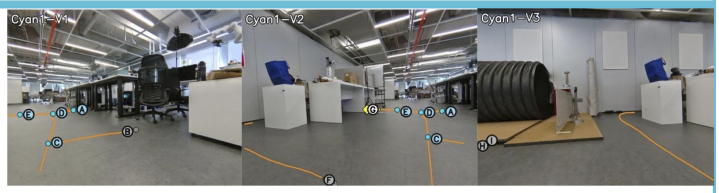


why: The target is a potted plant in a reception area. The agent is currently in a laboratory/R&D-style office environment, where the corridor ahead appears to lead toward a more open area or reception/entrance space. D is the nearest cyan junction along the forward route and provides a useful position for inspecting the long corridor ahead and its side branches.

O3 **E**

target_visible: no **target_in_view: none**
chosen_view: view 2

scene: Open laboratory/workspace area in an office building, currently beside a broad passage with long tables and displayed items.



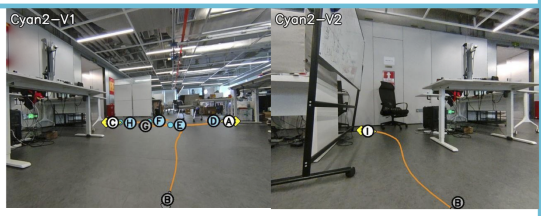
why: The target is a potted plant in a reception area. The agent is currently inside a laboratory/office workspace. E follows the passage toward a deeper open area or exit direction and connects to further junctions ahead, making it a suitable observation point for progressing toward and searching for a reception lobby.

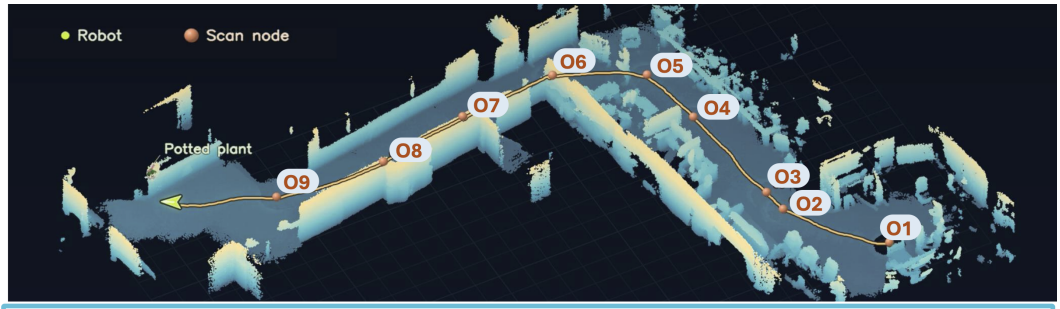
O4 **E**

target_visible: no **target_in_view: none** **chosen_view: view 1**

scene: Open office workspace with multiple desks, whiteboard partitions, and junctions leading to other parts of the office.

why: The target is a potted plant in a reception area. The agent is currently inside an open office. E is a key cyan junction ahead that connects to multiple routes extending left, right, and deeper into the workspace (e.g., C, H, F, D, and A). Moving to E first provides a better view of these open passages and helps identify a route toward the reception area.

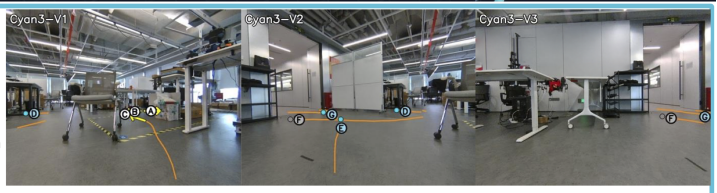




O5 **G**

target_visible: no **target_in_view: none**
chosen_view: view 2

scene: Open office/laboratory area with a doorway leading to an external corridor.



why: The agent is currently inside an office/laboratory area, and the target is a potted plant in a reception area. Views 2 and 3 show a doorway leading to an external corridor. G is located just beyond the doorway and serves as a key junction for leaving the current room and exploring routes toward the reception.

O6 **C**

target_visible: no **target_in_view: none**
chosen_view: view 2

scene: Office-building corridor and entrance transition area.



why: The target, a potted plant in the reception area, is not visible in the current view. The corridor ahead leads toward a more open area, and C is a cyan junction located further along the route. It serves as a suitable waypoint for gathering information about the connected branches and potential routes toward the reception area.

O7 **A**

target_visible: no **target_in_view: none**
chosen_view: view 1

scene: Office-building corridor near a restroom entrance.

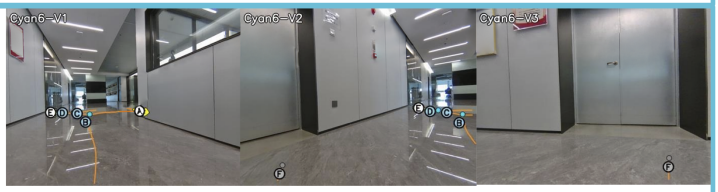


why: The target is a potted plant in the reception area. View 3 clearly leads into a restroom, which is unlikely to contain the reception area. Views 1 and 2 continue along the main office corridor and are more likely to lead toward the lobby. A is a cyan junction on the main corridor ahead, making it a suitable waypoint for moving forward and observing possible routes toward the reception area.

O8 **B**

target_visible: no **target_in_view: none**
chosen_view: view 1

scene: Office-building corridor leading toward a lobby/reception area.



why: A more open space resembling a lobby or reception area is visible further ahead. B is a cyan junction at the corridor intersection leading toward this area, making it a suitable waypoint for moving forward and determining the exact location of the reception area and the potted plant.

O9 **H**

target_visible: YES **target_in_view: none**
chosen_view: view 1

scene: Office-building corridor leading toward a lobby/reception area.



why: A more open space resembling a lobby or reception area is visible further ahead. B is a cyan junction at the corridor intersection leading toward this area, making it a suitable waypoint for moving forward and determining the exact location of the reception area and the potted plant.

Figure 5: **Real-world deployment of MarvisNav on a Unitree Go2.** The robot searches for a *potted plant in the reception*. The point-cloud overview shows the executed trajectory and decision sites O1–O9. The accompanying panels present the annotated egocentric views, selected route identifiers, and original VLM responses. Route identifiers are local to each decision.

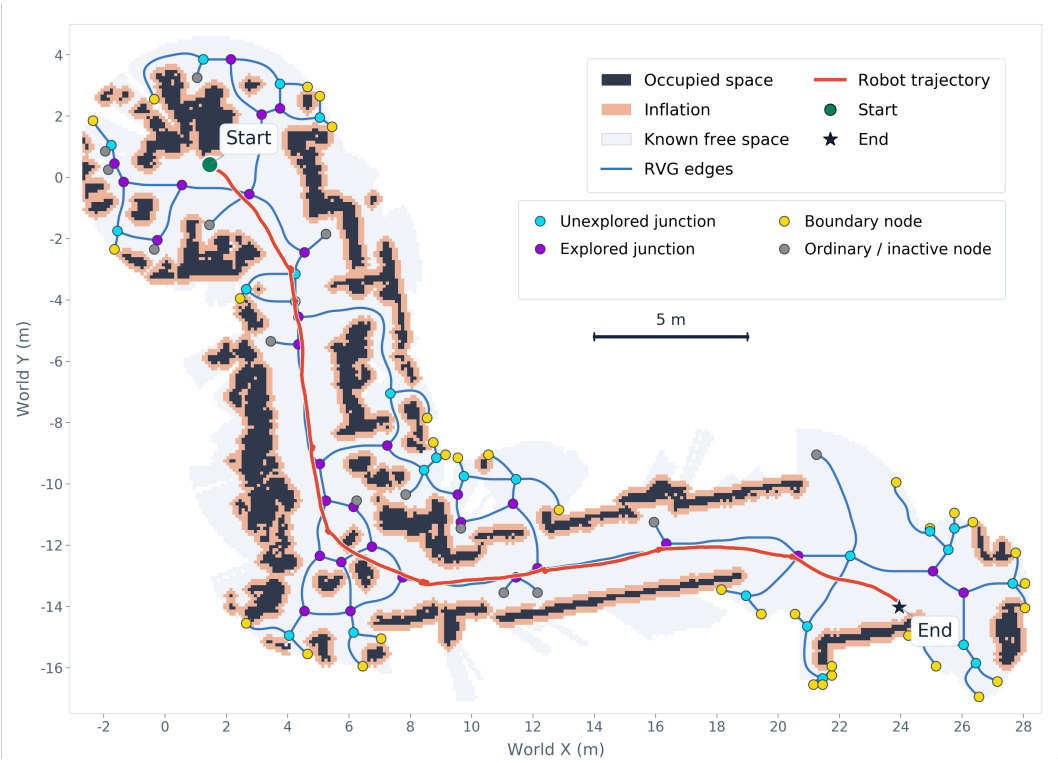


Figure 6: **State-Aware RVG and executed trajectory in the representative indoor deployment.** Starting from an empty map, MarvisNav incrementally constructs the RVG and updates node exploration states from online RGB-D observations. The final map shows the resulting topology, exploration states, and robot trajectory from start to end.

C Scene Reconstruction for Visualization

As shown in Fig. 7, raw scan-derived visualizations of HM3D scenes can be visually cluttered, making their underlying spatial layout difficult to parse at a glance. To facilitate interpretation of our method, we independently reconstructed the layout of one HM3D scene using Pascal Editor, following its original layout and relative spatial proportions. This is an actual scene used in our evaluation, rather than a toy environment created solely for illustration. As illustrated in the figure, we present the original top-down view, the reconstructed 3D model, and its rendered top-down view. Importantly, this reconstruction is used only for visualization, and evaluations are performed in the original HM3D environment. We will release the reconstructed 3D model as a reusable asset for future work, supporting visualization and qualitative analysis in the corresponding HM3D scene.



(a) Original HM3D top-down view.

(b) Reconstructed 3D model.

(c) Top-down rendering of the reconstructed model.

Figure 7: Scene reconstruction used to create method illustrations. The reconstruction follows the layout and relative spatial proportions of HM3D scene `00877-4ok3usBNeis`.

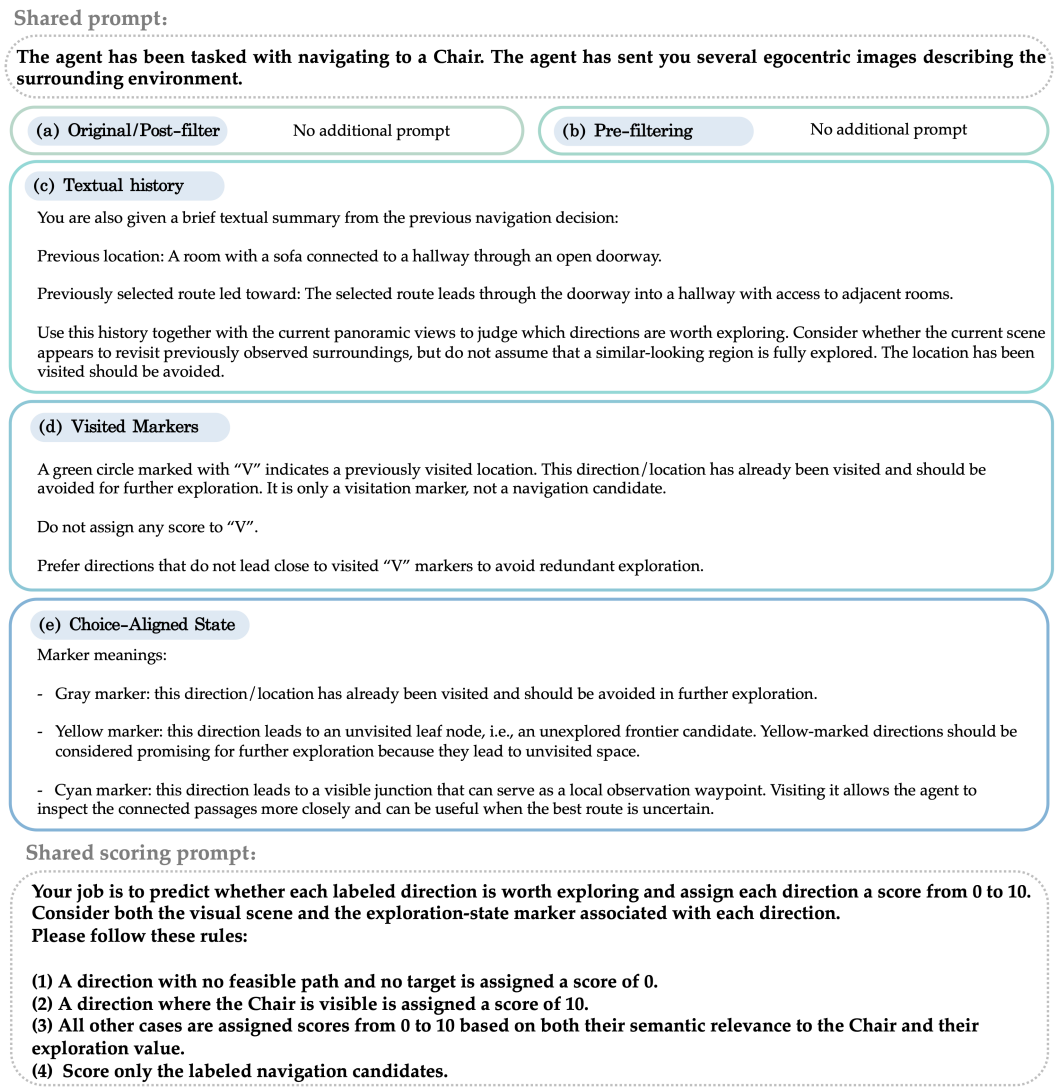


Figure 8: **Prompt templates for the motivation study.** All variants share the same task and scoring instructions and differ only in how exploration memory is provided.

D Details of the Motivating Study

D.1 Setup and memory representations

We conducted the motivating study of how memory representation affects the evaluation of the VLM route. Across variants, we fix the navigation scene, target category (*Chair*), underlying egocentric observation, candidate geometry, image resolution, base prompt, and model configuration, making only the changes required by each memory representation. For a simple and visually transparent comparison, we omit the full RVG paths and retain only the candidate markers and memory cues. We compare five representations. **(a) Original / Post-filter** provides the reference VLM scores without exploration memory; the same raw scores are used before memory is applied by a post-filtering rule. **(b) Pre-filtering** removes the explored candidate **A** before VLM evaluation, while **B**, **C**, and **D** retain their original identities and locations. **(c) Textual History** provides a VLM-generated summary of the previous decision as linguistic memory. For this diagnostic study, we use only the immediately preceding history entry, which describes the location just visited and is directly relevant to the current decision. This provides the minimal setting in which the VLM must associate textual memory with the corresponding visual choice, without introducing additional effects from history length. Our full

simulation ablation instead uses the most recent ten entries. **(d) Visited Markers** places a separate green V marker beside the explored candidate **A**. **(e) Choice-Aligned State** attaches the exploration state directly to each candidate, with gray denoting an explored endpoint, yellow an unexplored endpoint, and cyan an incompletely explored junction.

D.2 Prompt and evaluation protocol

We adopt the base scoring prompt from WMNav (Nie et al., 2025), as it provides an existing VLM-based navigation prompt for evaluating candidate directions from observations and does not encode exploration memory itself. This gives a common reference prompt across all representations and avoids tailoring the base instruction to our proposed memory interface. All variants use the same scoring instruction and differ only in the minimal representation-specific information required by each interface. The complete prompts are provided in Fig.8. The VLM scores each displayed candidate from 0 to 10, with higher scores indicating a stronger preference for further exploration. *Textual History* adds the history summary, while *Visited Markers* and *Choice-Aligned State* add only the corresponding marker definitions. We evaluate Gemini-3.5-Flash and Qwen3.5-Plus with temperature 0 and thinking disabled. Each representation is queried independently 10 times per model. Repeated calls can still produce different outputs, particularly for Qwen3.5-Plus, so Fig. 2 reports the mean and standard deviation of the candidate scores across runs. These scores are used only to make changes in VLM preference visible in this motivation study and are not part of the MarvisNav navigation pipeline. The complete raw responses will be provided along with the code.

E Navigation and Ablation Prompts

For the full prompt used in navigation experiments, we replace the shared scoring block in Fig. 8 with the navigation decision prompt shown in Fig. 9. This prompt instructs the VLM to select a route candidate and report target visibility. The memory-representation ablations follow the same modular prompt structure as Appendix D, with memory descriptions and marker definitions specific to each representation.

F Failure Case Analysis

To better understand the remaining limitations of MarvisNav, we analyze unsuccessful episodes on HM3D v0.1, HM3D v0.2, and MP3D using the recorded termination reasons as shown in Fig.10. Despite different overall success rates, the failure composition is broadly consistent across benchmarks. False-positive target detections are the dominant failure mode, accounting for 64.9%, 59.6%, and 57.7% of failures on HM3D v0.1, HM3D v0.2, and MP3D, respectively. Step-budget exhaustion is the second most common outcome (23.0%–33.5%), while robot-stuck failures remain relatively infrequent (6.9%–12.4%). The prevalence of false positives indicates that target perception remains an important bottleneck beyond high-level route selection. Step-budget exhaustion can result from inefficient exploration, including missed target detections, or simply from the scale of the environment. The latter is particularly relevant on MP3D, where increasing the maximum step budget may alleviate some failures. Robot-stuck cases are largely caused by imperfections in the scanned benchmark geometry, such as holes or discontinuities that disrupt low-level navigation. Overall, these results suggest that improving target perception and recovery from long-horizon failures are promising directions for further improvement in future work.

As HM3D v0.2 has become an common evaluation benchmark for ZSON methods (Zhang et al., 2025c; Padilla et al., 2026), the quality of its evaluation episodes directly affects cross-method comparison. To support more reliable evaluation, we manually inspect all $1{,}000$ episodes in the HM3D v0.2 evaluation split and identify 52 episodes with benchmark-related issues. These cases fall into three main categories: incomplete scene reconstruction, including large holes, missing geometry, severely degraded small objects, or geometry that makes targets unreachable; inconsistent target annotations, where the annotated category does not match the visible object; and episodes in which no valid target instance exists in the agent's reachable region. The representative cases are shown in Fig 11. Such defects can turn benchmark limitations into apparent navigation failures, and these 52 episodes therefore impose an evaluation ceiling unrelated to navigation policy quality. We provide

Shared navigation decision prompt:

First inspect all views and determine whether the target, **{Object}**, is visible. A partially visible or distant **{Object}** still counts as visible. If **{Object}** appears only in a painting, photograph, wallpaper, or screen, it does **not** count as the physical target.

If the target is visible: **(1)** Select the marker that brings the agent **closest to the physical {Object} while remaining reachable**. **(2)** Do not simply choose a marker that visually overlaps **{Object}** or nearby furniture. Such a marker may correspond to a waypoint behind the object or across an obstacle. Instead, choose the nearest accessible waypoint that approaches **{Object}** without overshooting it.

If the target is not visible: **(1)** Use the scene context and candidate waypoints to select the most promising unexplored direction. Infer the room type and where each direction is likely to lead. **(2)** Do not choose stairs or entrances leading to another floor; **{Object}** is on the current floor. **(3)** If a cyan_junction must be passed before reaching a farther candidate, select the junction first.

Select exactly one letter. If no direction is useful, output **NONE**. A candidate may be permanently excluded only when the corresponding area is clearly a fully observed dead end, contains no **{Object}**, and has no remaining doorway, corridor, passage, or unexplored region. If there is any uncertainty, do not exclude it.

Output Format

Output exactly the following five lines first:

<A-Z | NONE>
target_visible: yes | no
target_in_view: view N, <location> | none
chosen_view: N | none
END_DECISION

N refers to the image order starting from 1.

After END_DECISION, provide:

scene: <brief scene description>
why: <brief reason for the choice>

Do not repeat or modify the decision fields after END_DECISION.

Figure 9: Shared navigation decision prompt

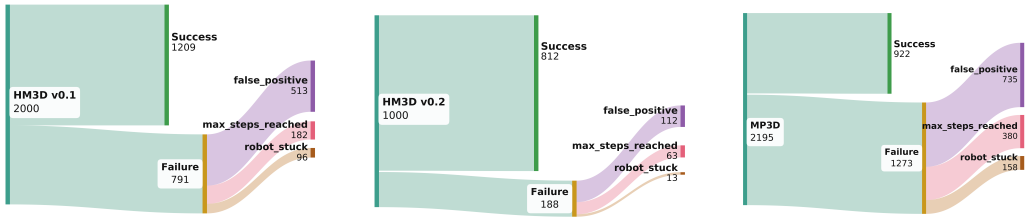


Figure 10: **Failure analysis across HM3D v0.1, HM3D v0.2, and MP3D.** Evaluation episodes are decomposed into successes and three failure outcomes: false-positive target detection, maximum step budget reached, and robot stuck. Numbers denote episode counts.

the corresponding scene and episode IDs together in Table 4 with their issue categories to support independent verification, reproducible comparison, and future benchmark curation by the community, and recommend excluding the verified invalid episodes when constructing a cleaned HM3D v0.2 evaluation split. More broadly, we hope this analysis supports more reliable ObjectNav evaluation by helping separate algorithmic failures from artifacts of benchmark construction. It also highlights the need for better-curated and more reliable benchmarks in future research.

G Future Work

Future work could extend the local exploration summaries in the State-Aware RVG to hierarchical summaries that combine node- and region-level progress while keeping memory aligned with current visual choices. In this work, we use RVG to construct memory-bearing visual route choices, but the same design could be adapted to other topological graphs. Future work could also extend the spatial representation to support multi-floor object search (Werby et al., 2024). Another direction

Table 4: **Benchmark issues identified in the HM3D v0.2 evaluation split.** We report the 52 identified episodes grouped by issue type and scene.

Issue Type	Scene ID	Episode IDs	Number of Episodes
No valid target in reachable region	7MXmsvcQjpJ	84, 89, 96, 109	4
Annotation error / omission	Dd4bFSTQ8gi	196, 197, 222	3
	bCPU9suPUw9	508, 511, 528	3
	jxsvVRusffK	538, 539, 558	3
	p53SfW6mjZe	732	1
	a8BtkwhxdRV	492, 499	2
	GLAQ4DNUx5U	228, 238, 243	3
	EF36g7L6Z9M	595, 610, 611	3
	mL8ThkuaVTM	673	1
	svBbv1Pavdk	841, 843, 846, 863	4
	BAbdmeyTvMZ	119	1
	cvZr5TUy5C5	566	1
	5cdEh9F2hJL	32, 44	2
	HY1NcmCgn3n	261, 262, 263, 264, 265, 266, 267, 268, 269	9
	yr17PDCnDDW	946, 948	2
Incomplete scene reconstruction	HY1NcmCgn3n	254, 255, 259, 260	4
	bCPU9suPUw9	520, 527	2
	k1cupFYWXJ6	644, 646, 648, 649	4
Total problematic episodes			**52**

is to improve geometric reliability around glass and reflective surfaces through uncertainty-aware depth estimation and multi-view verification (Zheng et al., 2026), providing more reliable geometry for RVG construction and the projection of visual route candidates into egocentric views. Finally, MarvisNav's candidate-selection interface motivates exploring lower-latency decision backends, including real-time multimodal interaction through the Gemini Live and visual extensions inspired by Jev's structured-decision design Almeida (2026). These advances may enable VLM-based ZSON to approach the responsiveness of end-to-end navigation policies without task-specific policy training.



(a) **k1cupFYWXJ6**, ep 649



(b) **Dd4bFSTQ8gi**, ep 197



(c) **7MXmsvcQjpJ**, ep 96

Figure 11: **Representative benchmark-related failure cases from HM3D v0.2.** Each row shows a representative problematic episode, including the egocentric observation, the corresponding depth/geometry view, and a top-down scene map. In the third column, red regions indicate the benchmark ground-truth object region. **(a) Incomplete scene reconstruction** (`k1cupFYWXJ6`, ep 649, target: *sofa*): missing or corrupted scene geometry around the target region makes the episode unreliable for evaluation. **(b) Annotation error / omission** (`Dd4bFSTQ8gi`, ep 197, target: *chair*): the agent successfully reaches a visible chair, but the benchmark ground-truth annotation is incomplete or inconsistent, so the episode is incorrectly counted as a false positive. **(c) No valid target in the reachable region** (`7MXmsvcQjpJ`, ep 96, target: *sofa*): the episode asks the agent to search for a sofa in a basement area where no valid target instance exists. These examples illustrate three representative benchmark defects that can cause apparent navigation failures unrelated to policy quality.