WHICH TO VISIT, WHEN TO STOP: AUTONOMOUS MULTI-OBJECT SEARCH WITH INTERACTIVE 3D MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing embodied navigation benchmarks support diverse goal modalities and long-horizon tasks, but typically prescribe goal sequences or task lengths, leaving two central decisions underexplored: *which to visit* and *when to stop*. We introduce MOST-Bench, a benchmark for autonomous search in embodied navigation, where agents must select target instances, organize visits, track progress, and determine when the search is complete. MOST-Bench comprises two complementary tasks: Many-Object Navigation (ManyON), which requires visiting a specified number of distinct matching instances, and All-Object Navigation (AllON), which requires finding all matching instances and deciding when to stop without knowing the target count. The benchmark contains 750 episodes across 36 HM3D scenes, covering five task–modality combinations with 2–10 targets per episode. We also introduce MOST-Nav, a training-free baseline that combines vision-language model decision-making with interactive 3D memory reconstructed from RGB observations. We evaluate seven training-based and training-free frameworks on task completion, reporting accuracy, and target coverage. The best-performing method achieves an overall success rate below 18%, highlighting the difficulty of autonomous search and motivating further research on coordinating exploration, instance-level memory, and search termination. The project page is available at `https://most-bench.netlify.app/`.

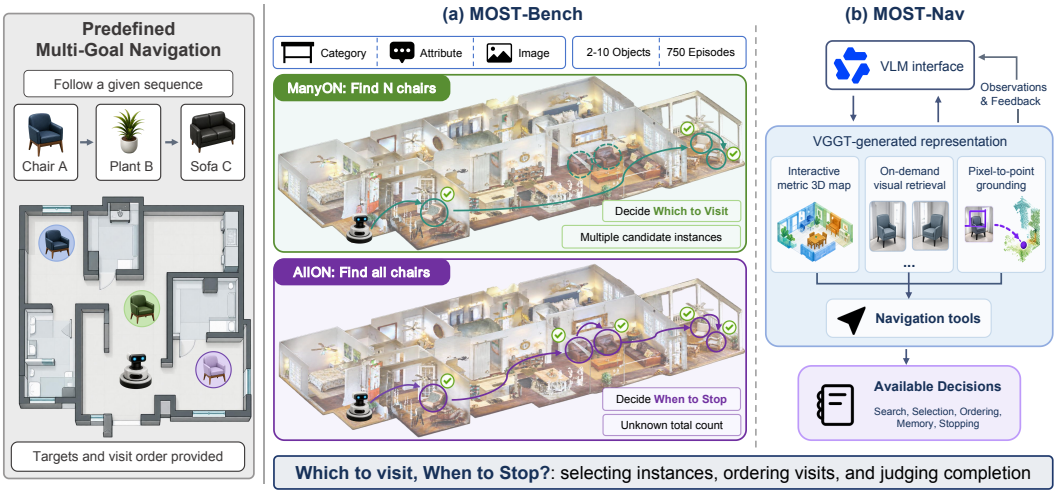


Figure 1: **MOST-Bench and MOST-Nav.** ManyON requires a known quota and allows the agent to choose the visit order and, when additional candidates exist, the target subset. AllON requires exhaustive search with a hidden target count and an explicit completion signal. MOST-Nav couples a frozen VLM with interactive 3D memory and navigation tools.

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.

1 INTRODUCTION

Embodied navigation is a foundational capability for robots performing everyday tasks such as household organization and warehouse order picking. These tasks require agents to interpret diverse goal specifications, including images, object categories, and language descriptions, and to plan and execute navigation actions accordingly (Zhang et al., 2026; Chu et al., 2026).

Object navigation is a central research problem in embodied navigation and has been explored in multiple directions. Building on single-goal navigation (Savva et al., 2019; Batra et al., 2020), subsequent work requires agents to sequentially reach multiple goals specified through different modalities (Wani et al., 2020; Khanna et al., 2024), follow more fine-grained navigation instructions, or retain environmental memory over extended task horizons (Song et al., 2025; Raychaudhuri et al., 2025). These settings evaluate multimodal understanding, instruction-based planning, and memory reuse across longer tasks. However, they typically prescribe goal sequences or task lengths, largely composing navigation subtasks whose objectives are externally specified. Consequently, they place limited emphasis on two task-level decisions: *which to visit* and *when to stop*.

For example, an instruction such as "find a chair, then find a wooden table" specifies the navigation goals and their order in advance. Although the agent must determine when each navigation subtask succeeds, it need not organize the overall goal sequence or judge whether an open-ended search is complete. In contrast, requests such as "find three black chairs" or "find all clothes in the house" place greater emphasis on the agent's decision-making ability, requiring it to manage the search autonomously. *Which to visit* concerns selecting the next target instance and revising visit order as new evidence becomes available. *When to stop* concerns determining whether the overall search requirement has been satisfied, particularly when the target count is unknown. Together, these decisions demand spatial memory that distinguishes visited instances from new discoveries and relates local observations to overall task progress.

We refer to navigation that requires agents to autonomously select instances, organize visits, and determine search completion in an initially unknown environment as *autonomous search*. Unlike executing an externally prescribed goal sequence, autonomous search requires agents to construct and revise their plans based on accumulated spatial evidence and task progress. This shifts the evaluation focus from completing individual navigation goals to managing a search process.

We introduce MOST-Bench to evaluate autonomous search through two complementary tasks (Figure 1). In *Many-Object Navigation* (ManyON), agents must locate and visit a specified number of distinct matching instances within a step budget. In *All-Object Navigation* (AllON), the target count is withheld, requiring agents to search an unknown environment for all matching instances and determine when to stop. Both tasks support category and attribute-based language goals, while image goals are used only in ManyON to specify a fixed set of reference instances. Thus, agents select matching instances for category and attribute goals, and organize visits among designated instances for image goals. MOST-Bench contains 750 episodes across 36 HM3D scenes, with balanced coverage of the five valid task–modality combinations and target counts from 2 to 10.

We further present MOST-Nav, a training-free reference baseline with interactive 3D memory. MOST-Nav reconstructs the environment from RGB streams using VGGT-SLAM (Wang et al., 2025; Maggio et al., 2025). Through an interface, a vision-language model (VLM) queries rendered map views, retrieves registered RGB observations, and requests memory updates to ground selected visual targets in 3D. The VLM makes task-level decisions, while the navigation system maintains the map and plans navigation actions. This interaction provides spatial evidence for choosing subsequent targets, tracking previous visits, and assessing whether further search is needed.

Using MOST-Bench, we evaluate seven representative training-based and training-free frameworks, including MOST-Nav, reporting task completion and analyzing search progress. Success rate measures whether an episode satisfies its task-specific completion criteria; precision measures the correctness of reported instances; recall measures coverage of the target requirements; and F1 summarizes precision and recall. Our analysis compares completion and reporting quality across tasks and goal modalities, and examines how coverage and task completion vary with the required number of targets. The results show that autonomous search remains challenging for all evaluated methods, leaving substantial room for improvement. These findings motivate further research into agents that coordinate exploration, instance-level memory, and search termination.

Our contributions are threefold:

- We formulate autonomous search around two task-level decisions—which to visit and when to stop—and introduce ManyON and AllON to evaluate navigation with specified and undisclosed target counts.
- We establish MOST-Bench, comprising 750 episodes across 36 HM3D scenes and five task–modality combinations, with a unified evaluation protocol for task completion, reporting accuracy, and target coverage.
- We introduce MOST-Nav, a training-free baseline coupling VLM decision-making with interactive 3D memory, and evaluate seven representative frameworks to characterize current capabilities and remaining challenges in autonomous search.

2 Related Work

2.1 Embodied Navigation Benchmarks

Embodied object navigation benchmarks developed along two directions. The first broadens goal categories and input modalities, from closed-set categories (Batra et al., 2020; Zhang et al., 2025b) to open-vocabulary categories (Yadav et al., 2023; Yokoyama et al., 2024b; Busch et al., 2025; Xue et al., 2026) and goal modalities, including language and images (Khanna et al., 2024; Raychaudhuri et al., 2025; Chen et al., 2020; Paul et al., 2022). The second extends task horizons through sequential multi-goal navigation, where agents receive each subsequent goal after reporting the last one, and retain environmental memory to complete prolonged navigation tasks (Wani et al., 2020; Song et al., 2025), evaluating memory reuse and navigation efficiency. However, these settings largely emphasize execution rather than autonomous search decisions. MOST-Bench complements this by requiring agents to determine the subset and order of target visits and when search is exhaustive. It requires agents to report distinct matching instances rather than complete an externally specified sequence of subgoals. Table 1 compares benchmark properties, including attribute and image goals, cross-category composition, and exhaustive search with a hidden target count.

Table 1: Task properties of representative object navigation benchmarks. *Free ordering*: the agent freely chooses the order in which to visit targets. *Exhaustive completion*: the agent must determine when all matching instances have been found, with the target count withheld.

Benchmark	Multi-obj.	Attr.	Image	Attr. comb.	Free ordering	Exhaustive completion
ObjectNav (Batra et al., 2020)	✗	✗	✗	✗	✗	✗
HM3D-OVON (Yokoyama et al., 2024b)	✗	✗	✗	✗	✗	✗
MultiON (Wani et al., 2020)	✓	✗	✗	✗	✗	✗
OneMap (Busch et al., 2025)	✓	✗	✗	✗	✗	✗
GOAT-Bench (Khanna et al., 2024)	✓	✓	✓	✗	✗	✗
LaMoN (Raychaudhuri et al., 2025)	✓	✓	✗	✓	✗	✗
MOST-Bench (Ours)	✓	✓	✓	✓	✓	✓

2.2 Vision-Language Models for Embodied Navigation

Vision-language models increasingly guide embodied navigation. Training-based methods learn spatial reasoning and navigation from extensive trajectory supervision (Zhang et al., 2024; 2025a; Zhu et al., 2025; Wei et al., 2026). Training-free methods augment pretrained VLMs with semantic maps and scene graphs for localization and exploration (Huang et al., 2023; Yu et al., 2023; Yokoyama et al., 2024a; Yin et al., 2024; Huang et al., 2026; Chen et al., 2026b; Maggio et al., 2026), or visual memories for observation retrieval (Yang et al., 2025; Hu et al., 2026; Ge et al., 2026). Modular agentic systems coordinate reasoning, memory, grounding, perception, and navigation through structured evidence and execution feedback (Zhou et al., 2026; Chen et al., 2026a). However, these methods primarily address single or sequential goals, leaving instance selection, deduplicated reporting, and termination under hidden target counts underexplored. MOST-Nav addresses these challenges through an interactive 3D memory that tracks spatial coverage and individual instances. VGGT-based views (Wang et al., 2025; Maggio et al., 2025), anchored RGB observations, and pixel-to-point grounding jointly support instance selection and deduplicated reporting in ManyON, while informing exploration or FINISH decisions under hidden target counts in AllON.

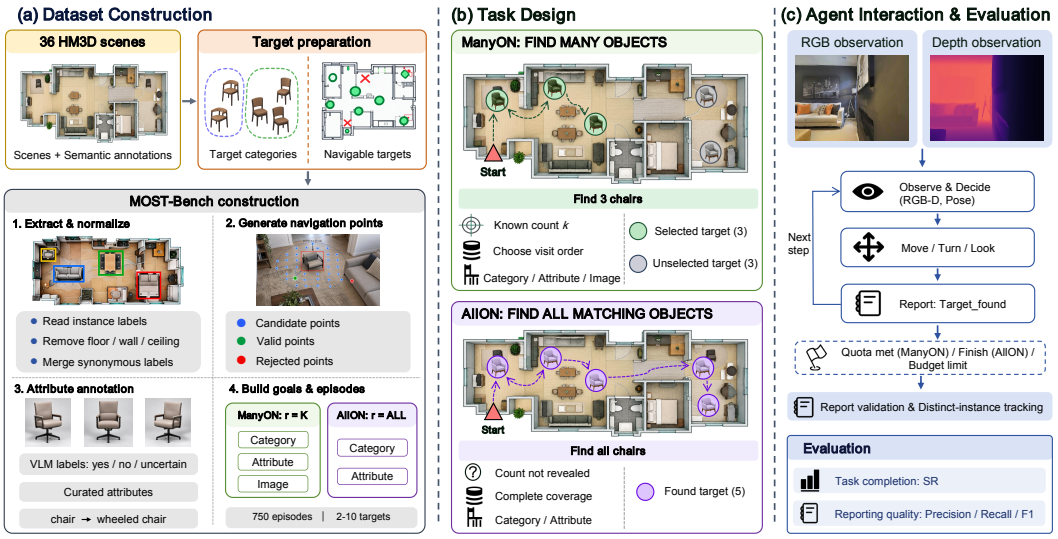


Figure 2: **Overview of MOST-Bench.** (a) Dataset construction across five task–modality combinations. (b) ManyON specifies a target count; AllON requires finding all matches and deciding when to stop. (c) Interaction and evaluation of task completion, reporting accuracy, and target coverage.

3 MOST-BENCH

MOST-Bench evaluates multi-object search, requiring agents to decide which instances to visit, in what order, and when to stop as illustrated in Figure 2. Many-Object Navigation (ManyON) requires finding a specified number of distinct matching instances, while All-Object Navigation (AllON) requires exhaustive search and an explicit completion signal without knowing the target count.

3.1 TASK FORMULATION

An episode consists of an initial pose in a scene s and an instruction $I = (d, c, r)$, where d is the agent-visible goal description, c is the evaluation criterion and determines which instances qualify, and r specifies the target requirement. Let $\mathcal{G}_s(c)$ be the corresponding ground-truth target set, and $M = |\mathcal{G}_s(c)|$. We use three goal forms: category specifications, attribute descriptions, and sets of marked reference images. Image goals appear only in ManyON and identify a fixed set of instances.

At each interaction step, the agent receives an egocentric RGB observation with depth and pose available. The agent navigates from an initial pose and reports targets using TARGET_FOUND. A report receives credit if the instance matches c, has not been credited previously, and the agent is within 0.25 m horizontally of one of its navigation points. Let $\mathcal{R}_t$ denote the distinct instances credited by step t. The standard interaction budget is 800 steps; details are given in Appendix B.

ManyON: meeting a known quota. The instruction specifies a quota $r \leq M$. The episode terminates when the quota is met, after r total report attempts, or upon budget exhaustion, and succeeds if $|\mathcal{R}_t| \geq r$ within budget. When $M > r$, the agent chooses a target subset and visit order. Invalid or redundant reports count toward termination, not the success quota.

AllON: exhaustive search with a hidden count. The instruction requests all matching instances ($r = \text{ALL}$), while M remains hidden. Success requires $\mathcal{R}_t = \mathcal{G}_s(c)$ followed by an explicit FINISH within the budget. This action is available only in AllON and requires the agent to judge whether any matching instances remain undiscovered.

3.2 DATASET CONSTRUCTION

We construct MOST-Bench from 36 scenes in the HM3D validation split (Ramakrishnan et al., 2021). We first extract instance identities and category labels from semantic annotations. Struc-

tural or unsuitable categories for navigation, such as floors, are removed, and synonymous labels are merged into canonical categories. For each remaining instance, we sample nearby floor positions and project them onto the navigation mesh to obtain valid navigation points. Instances without at least one valid navigation point are discarded. To prevent AllON tasks from containing inaccessible targets, a scene–category pair is retained only if all of its annotated instances survive this filtering. To support attribute-based goals, we curate visually verifiable attributes and use a VLM to label rendered instance views. The resulting inventory contains 3,327 navigable instances across 47 canonical categories.

For category and attribute goals, a language model verbalizes the target criterion c and count requirement r into an agent-facing description d. The generated description changes only how the task is presented; ground-truth target membership remains determined by c. Image goals use individually marked reference images as d and identify a fixed set of target instances. Initial poses are sampled from the navigation mesh and paired with these instructions to form episodes. Reachability from each initial pose to the relevant target navigation points is verified on the navigation mesh.

The evaluation set contains 750 episodes: 450 ManyON episodes with category, attribute, or image goals, and 300 AllON episodes with category or attribute goals. Each of the five task–goal groups contains 150 episodes. Required target counts range from 2 to 10. We stratify episodes into easy (2–4), medium (5–7), and hard (8–10) tiers, each containing 150 ManyON and 100 AllON episodes. Among ManyON episodes with category or attribute goals, 180 have more matching candidates than the quota ($M > r$) and 120 require the full matching set ($M = r$). The 150 image episodes use fixed reference sets. The evaluation therefore includes both target-subset selection and fixed-set coverage, while leaving the visit order unconstrained in all settings.

3.3 Evaluation Metrics

We evaluate task completion and target-reporting quality using four metrics.

Task completion. Success rate (SR) measures the fraction of episodes that satisfy the task-specific completion rule. For N episodes, let $S_i \in \{0, 1\}$ indicate whether episode i succeeds:

$$\mathrm{SR} = \frac{1}{N} \sum_{i=1}^{N} S_i. \tag{1}$$

ManyON succeeds upon meeting the required quota within the interaction budget, whereas AllON requires finding all matching instances and explicitly issuing FINISH.

Target-reporting quality. Precision (P) measures report accuracy, recall (R) measures coverage of the target requirement, and F1 summarizes their balance. Let n_i denote the required quota in ManyON or the full matching-set size in AllON. Let TP_i count distinct credited instances and FP_i count invalid or redundant reports. For each episode, we compute

$$P_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}, \qquad R_i = \min\left(1, \frac{\mathrm{TP}_i}{n_i}\right), \qquad F1_i = \frac{2P_iR_i}{P_i + R_i}. \tag{2}$$

We set $P_i = 0$ when no report is made and $F1_i = 0$ when $P_i + R_i = 0$, then average each metric equally across episodes. Full required coverage ($R_i = 1$) satisfies ManyON; AllON additionally requires an explicit FINISH.

4 Baseline Method: MOST-Nav

We introduce MOST-Nav as the primary baseline for MOST-Bench, as illustrated in Figure 3. MOST-Nav combines a frozen VLM with interactive 3D memory to address target selection, visit planning, instance deduplication, and completion judgment. Its design provides a simple reference for autonomous multi-object navigation, with limited task-specific engineering rather than extensive performance optimization. An online VGGT-based reconstruction registers RGB observations and target instances in a shared geometric frame. The representation system handles pose estimation, map maintenance, and low-level navigation, while the VLM retrieves visual evidence and makes task-level decisions. No task-specific training or fine-tuning is required.

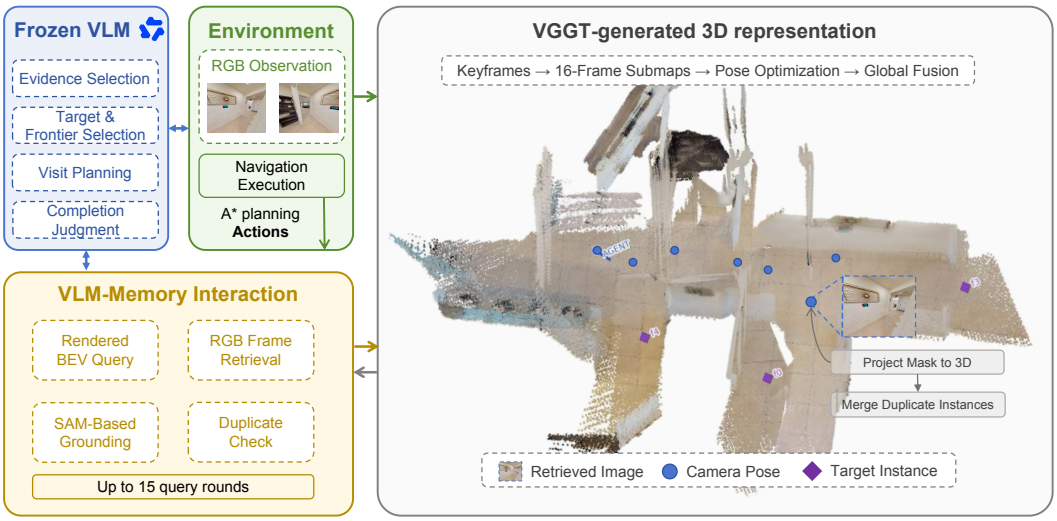


Figure 3: **Overview of MOST-Nav**: a navigation framework with VLM–3D memory interaction.

4.1 VGGT-BASED 3D MEMORY

A VGGT-SLAM server (Wang et al., 2025; Maggio et al., 2025) processes the egocentric RGB stream online. Keyframes are selected using optical-flow disparity, with a maximum-interval fallback for motion in textureless regions, and grouped into 16-frame submaps. Each completed submap triggers a VGGT forward pass followed by factor-graph optimization. Retrieval-based loop closure further refines previously estimated poses. Upon each submap commit, optimized keyframe poses and dense per-frame point maps are fused into a global point cloud and a gravity-aligned occupancy grid for A* path planning and frontier extraction.

The memory preserves both spatial context and visual evidence. The geometric map captures free space and exploration frontiers, while registered target locations encode instance-level spatial relations. Associated RGB observations remain accessible by frame ID, allowing the VLM to retrieve relevant evidence without loading the full observation history into its context. For spatial reasoning, the system renders a bird's-eye view annotated with persistent identifiers for frontiers and target instances, together with the agent's current pose and latest observation pose. The VLM inspects this view to reason about the layout and retrieves individual RGB frames for detailed visual checks.

4.2 VLM–MEMORY INTERACTION

MOST-Nav alternates between VLM decisions and navigation execution. Once the VLM selects a target instance or frontier, the representation system plans and executes a path while updating the reconstruction. Replanning and collision recovery are handled within the navigation system. Arrival, navigation failure, scan completion, or completion of a requested action triggers the next decision cycle. At each cycle, the VLM receives the remaining step budget, the previous execution outcome, and the current spatial state. It may perform up to 15 rounds of map queries before selecting an action, which is checked against the current map state and action schema.

The query–update interface supports evidence retrieval, 3D target grounding, and instance identity checks. To ground a target in a current or retrieved frame, the VLM requests candidate masks from SAM (Kirillov et al., 2023) and selects the corresponding mask. The representation system then uses the frame's VGGT point map and optimized pose to project the selected region into the global reconstruction and create an instance record for navigation. If the estimated location is close to an existing instance, the VLM compares their RGB observations to determine whether they correspond to the same physical object. Matching observations are linked to a shared instance record. This mechanism supports deduplicated target reporting while restricting semantic processing to observations selected on demand.

Table 2: Task completion and reporting quality (%). ManyON includes category, attribute, and image goals, while AllON includes category and attribute goals. **Bold** and underlined values indicate the best and second-best results, respectively.

Method	ManyON (450)				AllON (300)				Overall (750)			
	SR	P	R	F1	SR	P	R	F1	SR	P	R	F1
NaVid (Zhang et al., 2024)	6.22	19.30	25.51	21.40	0.00	0.57	30.91	0.83	3.73	11.81	27.67	13.17
UniNaVid (Zhang et al., 2025a)	7.33	20.27	27.18	22.58	0.00	0.23	30.55	0.41	4.40	12.25	28.53	13.71
InternVLA-N1 (Wei et al., 2026)	3.33	15.63	21.32	17.54	0.00	0.37	27.37	0.71	2.00	9.53	23.74	10.81
MTU3D (Zhu et al., 2025)	9.78	32.65	36.17	32.98	0.00	12.23	38.23	12.74	5.87	24.48	36.99	24.88
3D-Mem (Yang et al., 2025)	19.56	48.51	50.26	47.63	6.33	23.00	**56.05**	25.42	14.27	38.30	**52.58**	38.74
SCOPE (Wang et al., 2026)	**20.67**	51.19	**51.35**	**49.49**	6.67	29.65	52.69	30.91	15.07	42.57	51.88	42.06
MOST-Nav (Ours)	17.33	**55.26**	49.02	49.40	**18.33**	**48.27**	54.16	**46.99**	**17.73**	**52.46**	51.08	**48.44**

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmark settings. We evaluate all methods on the same 750 episodes from 36 HM3D validation scenes, following the construction in Section 3.2. The evaluation set comprises 450 ManyON episodes with category, attribute, or image goals and 300 AllON episodes with category or attribute goals, yielding five task–goal groups of 150 episodes each. Required target counts range from 2 to 10 and define three difficulty tiers: easy (2–4), medium (5–7), and hard (8–10). ManyON provides a target quota, whereas AllON withholds the target count and requires an explicit completion decision. All methods share the same reporting criterion: a distinct matching instance is credited when the agent is within $0.25\,\mathrm{m}$ horizontally of one of its navigation points. We report SR, precision, recall, and F1 as defined in Section 3.3, averaging each metric equally across episodes. Thus, ManyON and AllON contribute 60% and 40% to the overall results, respectively. Full interaction and aggregation details appear in Appendices B and C.3.

Method settings. We compare four trained policies—NaVid (Zhang et al., 2024), UniNaVid (Zhang et al., 2025a), InternVLA-N1 (Wei et al., 2026), and MTU3D (Zhu et al., 2025)—using existing checkpoints without further training on MOST-Bench. InternVLA-N1 uses the DualVLN V2 configuration. We also evaluate three training-free agents: 3D-Mem (Yang et al., 2025), SCOPE (Wang et al., 2026), and MOST-Nav, all using `Qwen3.8-Flash` as their VLM. Each method retains its native sensory inputs: NaVid, UniNaVid, InternVLA-N1, and MOST-Nav use RGB observations, while MTU3D, 3D-Mem, and SCOPE additionally use depth and pose. Baseline adapters preserve pretrained components and core navigation mechanisms while adapting task inputs and reporting and termination semantics to ManyON/AllON. Agent memory persists across search rounds, without access to ground-truth target annotations, hidden counts, or report-correctness feedback. The trained policies and MOST-Nav operate under an 800-step budget counted at the primitive-action level. For 3D-Mem and SCOPE, a navigation step instead executes a trajectory segment through a Habitat NavMesh path planner; we allocate 80 steps, following their shortened GOAT-style evaluation budgets (50 rather than 500 steps per subtask). All search rounds share one episode-level budget. Method-specific adaptation and control details are provided in Appendix C.1.

5.2 PERFORMANCE ACROSS TASKS AND GOAL FORMS

Target discovery does not guarantee task completion. Table 2 reveals different system strengths under known-quota and exhaustive search. MOST-Nav achieves the highest overall SR (17.73%), precision (52.46%), and F1 (48.44%), outperforming SCOPE by 2.66, 9.89, and 6.38 percentage points, respectively. However, this advantage is concentrated in AllON: MOST-Nav raises SR from 6.67% to 18.33% and F1 from 30.91% to 46.99%, whereas SCOPE leads on ManyON SR (20.67% versus 17.33%) with nearly identical F1 (49.49% versus 49.40%). The distinction between discovery and completion is especially evident in AllON. Although 3D-Mem achieves the highest recall (56.05%), its SR is only 6.33%; MOST-Nav achieves slightly lower recall (54.16%) but substantially higher precision (48.27% versus 23.00%) and SR (18.33%). Thus, higher average target coverage alone does not imply more successful episodes under a criterion requiring exhaustive coverage and an explicit FINISH. Similarly, the four trained policies obtain nonzero recall but zero AllON SR,

compared with 3.33–9.78% SR on ManyON. These results expose a gap between partial discovery and full completion, although aggregate metrics cannot distinguish failures in coverage, instance tracking, and stopping judgment.

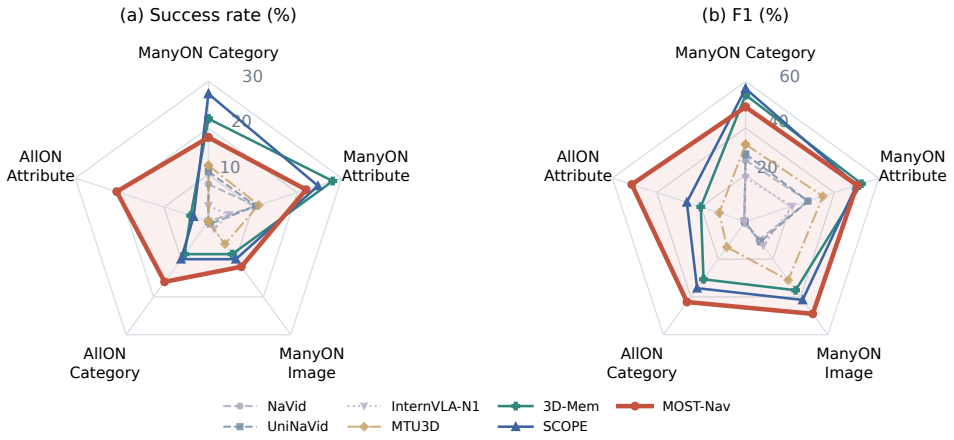


Figure 4: SR and F1 across five task–goal groups (150 episodes each). Image goals occur only in ManyON. Radial scales: 0–30% for SR and 0–60% for F1.

Performance across goal forms is task dependent. Figure 4 shows that no system consistently leads across the three ManyON goal forms: SCOPE achieves the highest F1 on category goals (56.91%), 3D-Mem on attribute goals (52.23%), and MOST-Nav on image goals (48.97%). MOST-Nav also leads SR on image goals, demonstrating its strongest relative performance within ManyON when reference images specify a fixed instance set. In contrast, its AllON advantage holds across both category and attribute goals, with the highest SR and F1 in each group. This consistency suggests that the overall AllON gain is not confined to a single goal input type. MOST-Nav obtains higher AllON F1 on attribute goals than on category goals (51.28% versus 42.70%), but this comparison does not isolate the effect of attribute conditioning because the episode sets differ. Table 5 reports the corresponding precision and recall.

5.3 Scaling with the Number of Required Targets

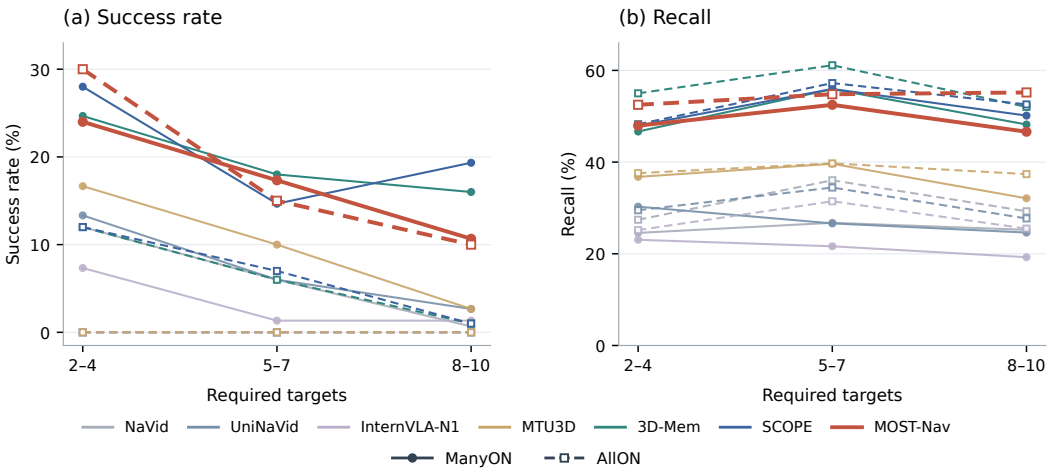


Figure 5: Task completion (SR) and fractional target coverage (recall) across target-count tiers. Both panels show ManyON (solid lines, circles) and AllON (dashed lines, hollow squares).

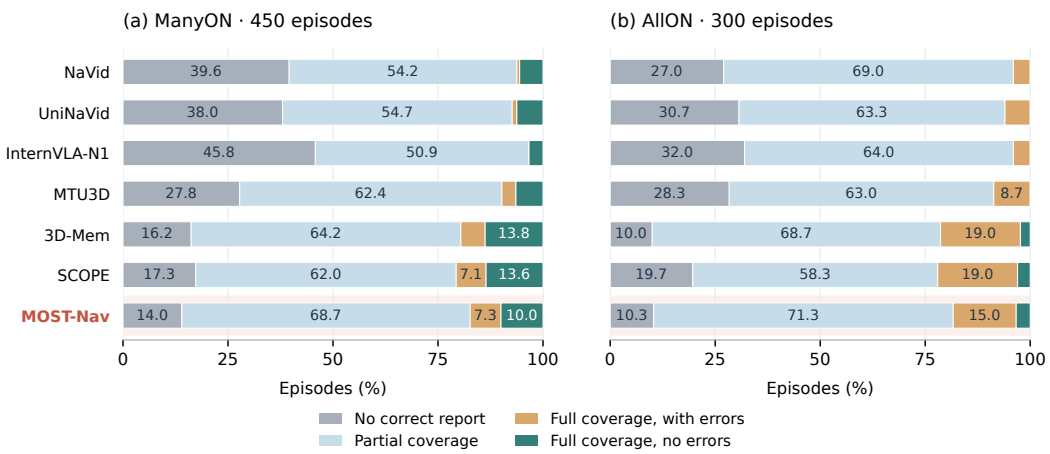


Figure 6: Distribution of reporting outcomes across tasks. Episodes are grouped into no correct report ($R = 0$), partial coverage ($0 < R < 1$), full coverage with invalid or redundant reports ($R = 1$, $P < 1$), and error-free full coverage ($P = R = 1$).

For target-count scaling, Figure 5 pairs SR with recall to compare task success with the fraction of required targets reported. Recall makes this coverage comparison directly, without combining it with report accuracy. MOST-Nav's ManyON SR decreases from 24.00% to 17.33% and 10.67% across the three tiers, while recall remains between 46.62% and 52.49%. Its AllON SR decreases from 30.00% to 15.00% and 10.00%, despite recall of 52.50%, 54.80%, and 55.19%. SCOPE shows a similar pattern on AllON: SR falls from 12.00% to 1.00%, while hard-tier recall remains 52.58%.

Comparable fractional coverage can thus accompany lower task success when more targets are required. These trends describe performance across count tiers; they do not isolate the effects of exploration, recognition, or reporting errors. Table 6 supplies precision and F1 for the same groups, complementing the SR/recall curves.

5.4 Partial and Full Target Coverage

Figure 6 uses per-episode precision and recall to characterize reporting outcomes within each task. Partial coverage accounts for 62.00–68.67% of ManyON episodes and 58.33–71.33% of AllON episodes for 3D-Mem, SCOPE, and MOST-Nav. Most episodes therefore include correct reports but fall short of the full target requirement, identifying incomplete coverage as a common failure mode. On ManyON, MOST-Nav has the lowest proportion of episodes with no correct report (14.00%), yet its larger partial-coverage share does not translate into a higher quota completion rate.

On AllON, SCOPE achieves full required coverage in 22.00% of episodes but error-free full coverage in only 3.00%. MOST-Nav reaches 18.33% and 3.33%, respectively. Thus, most full-coverage episodes still contain invalid or redundant reports. Full coverage also differs from task success: SCOPE's coverage rate exceeds its SR of 6.67%, indicating that many episodes find all required instances without issuing the required FINISH. These results distinguish three evaluation dimensions: covering all required targets, reporting them accurately, and explicitly declaring completion.

6 Conclusion

We introduced MOST-Bench for autonomous search, requiring agents to select instances, plan visits, maintain instance-level memory, and decide when to stop. Its complementary ManyON and AllON tasks cover specified and undisclosed target counts across diverse goal modalities. We also proposed MOST-Nav, a training-free baseline combining VLM decision-making with interactive 3D memory, and evaluated seven representative frameworks under a unified protocol. Results highlight persistent challenges in reliable search completion, motivating better coordination of exploration and instance tracking under partial observability.

AI USE STATEMENT

In this work, generative AI tools were used during benchmark construction to annotate visually verifiable object attributes from rendered views and to generate natural-language paraphrases of benchmark queries, as described in Appendices A.2 and A.3. Generative AI was also used to draft and polish the manuscript. Uncertain attribute annotations were manually verified. The authors take responsibility for the resulting benchmark artifacts and the final manuscript.

REPRODUCIBILITY STATEMENT

The paper includes detailed descriptions of the methods and experiments to facilitate reproducibility. The code and supplementary resources are available at `https://most-bench.netlify.app/`.

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. *arXiv preprint arXiv:2006.13171*, 2020.

Finn Lukas Busch, Timon Homberger, Jesús Ortega-Peimbert, Quantao Yang, and Olov Andersson. One map to find them all: Real-time open-vocabulary mapping for zero-shot multi-object navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 14835–14842, 2025.

Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. SoundSpaces: Audio-visual navigation in 3D environments. In *Proceedings of the 16th European Conference on Computer Vision*, pp. 17–36, 2020.

Yang Chen, Lirong Che, Zhenyu Huang, Wenbo Fu, Chuang Wang, Xu Cao, Daqi Liu, Yuzhe Yang, Jian Su, and Lan-Zhe Guo. HarnessVLN: Unifying training-free embodied navigation through an agent harness. *arXiv preprint arXiv:2609.15195*, 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. *arXiv preprint arXiv:2608.28279*, 2026b.

Zedong Chu, Shichao Xie, Xiaolong Wu, Yanfen Shen, Minghua Luo, Zhengbo Wang, Fei Liu, Xiaoxu Leng, Junjun Hu, Mingyang Yin, et al. ABot-N0: Technical report on the VLA foundation model for versatile embodied navigation. *arXiv preprint arXiv:2602.11598*, 2026.

Zuhao Ge, Xiaosong Jia, Chao Wu, Yuchen Zhou, Zuxuan Wu, and Yu-Gang Jiang. EvoMemNav: Efficient self-evolving fine-grained memory for zero-shot embodied navigation. *arXiv preprint arXiv:2606.03509*, 2026.

Junjun Hu, Xinda Xue, Botao Ren, Minghua Luo, Jintao Chen, Haochen Bai, Liangliang You, and Mu Xu. AstraNav-Memory: Contexts compression for long memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8097–8109, 2026.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 10608–10615, 2023.

Xun Huang, Shijia Zhao, Yunxiang Wang, Xin Lu, Wanfa Zhang, Rongsheng Qu, Weixin Li, Yunhong Wang, and Chenglu Wen. MSGNav: Unleashing the power of multi-modal 3D scene graph for zero-shot embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37154–37163, 2026.

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A benchmark for multi-modal lifelong navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16373–16383, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. In *Advances in Neural Information Processing Systems*, pp. 129839–129867, 2025.

Dominic Maggio, Nicolas Gorlo, and Luca Carlone. FOUND-IT: Foundation-model-first task-driven 3D scene graphs with granularity on demand. *arXiv preprint arXiv:2605.25371*, 2026.

Sudipta Paul, Amit Roy-Chowdhury, and Anoop Cherian. AVLEN: Audio-visual-language embodied navigation in 3D environments. In *Advances in Neural Information Processing Systems*, pp. 6236–6249, 2022.

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*, 2021.

Sonia Raychaudhuri, Enrico Cancelli, Tommaso Campari, Lamberto Ballan, Manolis Savva, and Angel X. Chang. MLFM: Multi-layered feature maps for richer language understanding in zero-shot semantic navigation. *arXiv preprint arXiv:2507.07299*, 2025.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied AI research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347, 2019.

Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12078–12088, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025.

Ningnan Wang, Weihuang Chen, Liming Chen, Haoxuan Ji, Zhongyu Guo, Xuchong Zhang, and Hongbin Sun. Expand your SCOPE: Semantic cognition over potential-based exploration for embodied visual navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18620–18628, 2026.

Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. MultiON: Benchmarking semantic map memory using multi-object navigation. In *Advances in Neural Information Processing Systems*, pp. 9700–9712, 2020.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground Slow, Move Fast: A dual-system foundation model for generalizable vision-and-language navigation. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Xinda Xue, Junjun Hu, Minghua Luo, Shichao Xie, Jintao Chen, Zixun Xie, Kuichen Quan, Wei Guo, Zedong Chu, Mu Xu, et al. OmniNav: A unified framework for prospective exploration and visual-language navigation. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4927–4936, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17294–17303, 2025.

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 *Advances in Neural Information Processing Systems*, pp. 5285–5307, 2024.

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

Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. HM3D-OVON: A dataset and benchmark for open-vocabulary object goal navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5543–5550, 2024b.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3MVN: Leveraging large language models for visual target navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3554–3560, 2023.

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

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.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, Yuxin Fan, Wenjun Li, Zhibo Chen, Fei Gao, Qi Wu, Zhizheng Zhang, and He Wang. Embodied navigation foundation model. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

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, 2025b.

Jian Zhou, Xunyi Zhao, Gengze Zhou, Zerui Li, Sihao Lin, Jiajun Liu, and Qi Wu. Embodied agents take control: Minimal-interface zero-shot agents rival industrial-scale policies in vision-and-language navigation. *arXiv preprint arXiv:2607.26148*, 2026.

Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, Siyuan Huang, and Qing Li. Move to understand a 3D scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8120–8132, 2025.

CONTENTS OF THIS APPENDIX

A MOST-Bench Construction Details **13**
A.1 Source Data and Filtering 13
A.2 Attribute Annotation 14
A.3 Instruction Generation and Episode Sampling 14
A.4 Dataset Composition 14

B Interaction Protocol **15**
B.1 Observations and Sensing 15
B.2 Actions and Report Validation 15
B.3 Termination and Success 15

C Experimental Details and Additional Results **16**
C.1 Method Adaptation 16
C.2 Per-Method Input Requirements 17
C.3 Evaluation Settings and Aggregation 18
C.4 Complementary Outcome Breakdowns 18
C.5 Robustness of Success Rate to Excessive Reporting 19

D Simulation Environment Visualization **20**

E Prompts **20**

SUPPLEMENTARY MATERIAL

This appendix details benchmark construction (Appendix A), interaction protocol (Appendix B), experimental settings and additional results (Appendix C), visualization of the dataset environment (Appendix D), and model prompts used in construction and evaluation (Appendix E).

A MOST-BENCH CONSTRUCTION DETAILS

This section expands Section 3 with the full construction pipeline: source data, navigability filtering, attribute annotation, instruction and episode generation, and dataset composition.

A.1 SOURCE DATA AND FILTERING

We build MOST-Bench on the 36 scenes of the HM3D v0.2 validation split (Ramakrishnan et al., 2021), using its instance-level semantic annotations as the source of object identities and category labels. We apply three curation steps to the raw label set: (i) structural or unsuitable categories are discarded (e.g., *floor*, *ceiling*, *wall*), since they do not correspond to discrete objects an agent can be asked to visit; (ii) synonymous labels are merged into a single canonical category (e.g., *TV* and *television→tv*), applied uniformly across all scenes; and (iii) for each retained category we enumerate its instances per scene, preserving the instance identities used for evaluation.

An instance is usable only if the agent can physically stand next to it. We formalize this through *navigation points*: navigable floor positions in the immediate vicinity of a target instance. For each instance we lay a dense horizontal grid with $0.25\,\mathrm{m}$ spacing over the region within $2\,\mathrm{m}$ of the instance's axis-aligned bounding box, yielding candidate floor positions around the object. Each

candidate is snapped onto the Habitat scene navmesh, i.e., the set of floor positions reachable by the embodied agent. A snapped point is retained as a valid navigation point only if the horizontal snap displacement is $\leq 0.25\,\mathrm{m}$, the downward vertical displacement is $\leq 2.0\,\mathrm{m}$ (allowing the candidate to project down onto the walkable floor, but not to jump up onto a different surface), and the snapped position is itself navigable on the navmesh. An instance with no valid navigation point is discarded. A category is retained as a target specification c in a scene only if all of its instances survive this filtering. This all-or-nothing rule is essential for AllON; it guarantees that the hidden ground-truth set $\mathcal{G}_s(c)$ contains no inaccessible instance, so exhaustive search is always physically achievable.

A.2 Attribute Annotation

HM3D provides only coarse category labels (e.g., *chair*, *cup*), which cannot support attribute-rich goals such as "a painting depicting a landscape." We therefore enrich a subset of categories with fine-grained, visually verifiable attributes. For selected categories we manually curate a small set of visually verifiable, category-discriminative attributes by inspecting rendered instances (e.g., *wooden*, *wheeled*, or *purple* for chairs); attributes that cannot be judged reliably from renderings are excluded during curation. Each instance of a category with curated attributes is then labeled by a judge VLM (`gpt-5.6-terra`), which inspects rendered close-up views of the instance from up to three directions and answers *yes*/*no*/*uncertain* for each candidate attribute. The exact judging prompt is given in Appendix E. A *yes* label creates a new fine-grained category from a raw category (e.g., *chair*→*wheeled chair*); *uncertain* labels are then verified manually, and *no* labels do not create a positive category. After this step the inventory contains 3,327 navigable instances across 47 canonical categories. We treat each such (fine-grained or coarse) category, as well as multi-category compositions, as a valid target specification c. Membership in the ground-truth set $\mathcal{G}_s(c)$ is determined solely by c and is independent of how the goal is later verbalized.

A.3 Instruction Generation and Episode Sampling

Let $M_{s,c}$ denote the number of instances matching specification c in scene s. We generate instructions from every (s, c) with $M_{s,c} \geq 2$: one AllON instruction with $r = \mathrm{ALL}$ (target count hidden from the agent), and up to three ManyON instructions with required counts sampled from $r \in \{2, 3, \ldots, M_{s,c}\}$. The instruction pool additionally includes AllON queries whose specification c spans multiple categories.

For category and attribute goals, the pair (c, r) is passed to an LLM (`gpt-5.6-terra`), which produces three paraphrases of the agent-facing description d; one paraphrase is sampled per episode. The paraphrasing affects only presentation—the ground-truth target membership is fixed by c and r. The generation prompt is given in Appendix E. Image goals appear only in the ManyON track: here d is a set of individually marked reference images, and c is the corresponding fixed set of reference instances. Images are captured from the same navigation points that define the success criterion at agent eye height ($1.5\,\mathrm{m}$), so the photo depicts a view the agent could actually attain. From the semantic mask we reject views whose target covers under 2% of the frame (too small to recognize) or over 50% (usually not intact in the image), and then rank survivors by proximity to a 12% ideal coverage and keep the best.

An episode pairs an instruction with an initial pose. Initial poses are sampled on the navmesh at a geodesic distance of 3–$100\,\mathrm{m}$ from the reference route to the targets, excluding trivially close and disconnected start locations. For every episode we verify, via shortest-path queries on the navmesh, that all relevant target navigation points are reachable from the sampled initial pose; episodes failing this check are resampled or discarded.

A.4 Dataset Composition

Figure 7 summarizes the composition of the evaluation set. The 750 episodes form five task–goal groups of 150 episodes each (inner ring): ManyON with category, attribute, or image goals, and AllON with category or attribute goals. The outer ring shows the raw query types behind each group. In particular, 91 AllON episodes are based on multi-condition compositions (`all_mixed`) that jointly specify several categories or attribute constraints in a single instruction. A composition is assigned to AllON–Attribute if it contains at least one attribute constraint, and to AllON–Category

otherwise. The right panel further reports the pool of distinct target instances behind each group: category goals draw on the largest pool (1,235 and 884 instances for ManyON and AllON, respectively), attribute goals on a mid-sized pool (632 and 623), and image goals on the fixed reference sets of their episodes (326). Instances may appear in multiple groups, so the per-group counts sum to more than the 3,327-instance inventory.

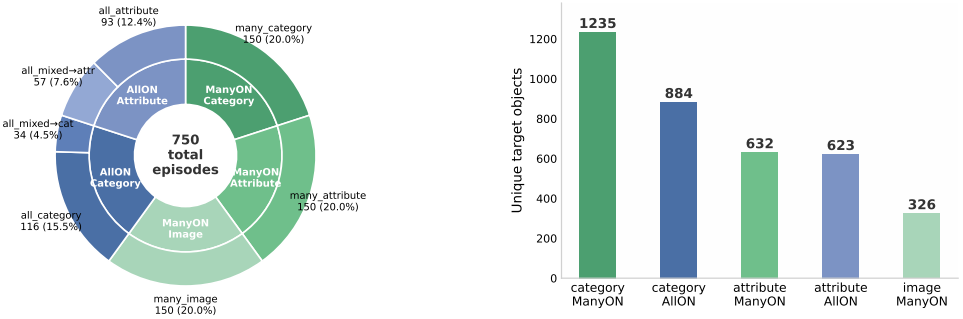


Figure 7: Left: composition of the evaluation set. Right: distinct target instances covered by each task–goal group.

B Interaction Protocol

B.1 Observations and Sensing

At each step the agent receives an egocentric observation from a sensor suite mounted at a height of $1.5\,\mathrm{m}$: an RGB frame of 1280×1280 pixels, a registered depth map, and the agent's pose (3D position and heading yaw). Both the RGB and depth sensors use a 120° horizontal field of view. In addition to the sensory inputs, the agent is told the remaining step budget and the outcome of its last action (e.g., success or collision). The instruction d is provided only once, at the beginning of the episode, and is not repeated; the agent must retain the goal in its own memory or context for the duration of the episode.

B.2 Actions and Report Validation

The standard action space consists of seven discrete actions: MOVE_FORWARD ($0.25\,\mathrm{m}$), TURN_LEFT and TURN_RIGHT (30°), LOOK_UP and LOOK_DOWN (30°), TARGET_FOUND, and FINISH, the last being available only in AllON. Forward motion uses navmesh sliding: if the straight-line displacement is blocked, the agent slides along the obstructing surface rather than aborting the step, so a colliding move still consumes one step and may produce a partial displacement. Episodes run under an 800-step budget.

The agent reports nearby targets using TARGET_FOUND. Each report is evaluated against all ground-truth target instances matching the episode criterion c. Every previously uncredited matching instance whose nearest precomputed navigation point lies within $0.25\,\mathrm{m}$ horizontally of the agent is credited. Thus, a single report action may credit multiple distinct instances.

A report that credits no new matching instance counts as one false positive. A report that credits one or more new matching instances contributes those instances as true positives and does not additionally count as a false positive. Consequently, true positives count credited instances, whereas false positives count unsuccessful report actions; the number of report actions need not equal $\mathrm{TP} + \mathrm{FP}$. For ManyON, credited instances are capped at the required quota when computing the metrics.

B.3 Termination and Success

In ManyON, the episode terminates when the quota $|R_t| \geq r$ is met, the agent has issued r reports, or the budget is exhausted, and succeeds if $|R_t| \geq r$ at termination. In AllON, the episode ends when

the agent issues FINISH or the budget is exhausted; success additionally requires that all matching instances have been credited at that point. A FINISH issued with uncovered targets simply results in episode failure—it carries no extra penalty beyond the resulting failure.

C Experimental Details and Additional Results

C.1 Method Adaptation

MOST-Bench uses a GOAT-style agent API (Khanna et al., 2024) extended with multi-object action semantics. Except for MOST-Nav, none of the evaluated methods natively support the ManyON/AllON interaction pattern: they are built either for single-goal navigation (ObjectNav-style, reaching one instance and stopping) or for externally sequenced multi-subtask navigation (GOAT-style, where each subgoal is revealed one at a time and the environment advances to the next given goal after a subtask-stop). Neither setting requires the agent to (i) decide which matching instance to visit and in what order, (ii) track distinct instances discovered under a single instruction so they are not double-counted, or (iii) decide when an exhaustive search is complete under a hidden target count. Running these methods on MOST-Bench therefore requires an adaptation layer that separates reporting an instance from terminating the entire search.

The adaptation preserves each method's pretrained models and navigation mechanisms, while modifying task inputs, report/termination semantics, and the interface to benchmark actions. No method is trained or fine-tuned on MOST-Bench. The adapter never exposes ground-truth target identities or locations, hidden target counts, per-report correctness, duplicate/redundant judgments, or the number of remaining targets. Report history is carried across rounds only as unverified agent-side memory. We organize the interface into three layers: (1) a shared *task-semantics* layer describing the goal and the ManyON/AllON requirement; (2) a *report-and-termination* layer maintaining the agent's own reports and implementing completion decisions; and (3) a method-specific *native control* layer. All rounds within an episode retain the method's visual-history or scene-memory state and share one episode-level action budget.

Training-based methods. We evaluate UniNaVid, NaVid, InternVLA-N1 DualVLN V2, and MTU3D using their existing checkpoints. For UniNaVid, NaVid, and DualVLN, the navigation instruction is augmented with the task requirement, the number of reports already issued, and up to 16 recent report poses. A native `stop`/`STOP` prediction is interpreted as reporting one candidate at the current position, rather than ending the episode. After a report, pending navigation actions are cleared and the agent continues searching with its visual history retained. Report poses provide only soft de-duplication hints: repeated or incorrect reports are neither rejected nor compensated with additional reporting opportunities.

The three methods retain their respective prediction mechanisms. UniNaVid uses its video-history encoding and online visual-token merging to predict discrete actions. NaVid retains its full observed-frame history and native distance/angle outputs, which are converted into benchmark forward and turning actions. DualVLN retains its S2 waypoint/arrow predictions and S1 trajectory-generation pipeline; the adapter executes short action sequences with synchronous replanning and preserves the native look-down interaction, followed by an explicit camera-restoration action. Thus, pretrained prediction components are retained, while action discretization, execution scheduling, and camera control are adapted to the benchmark API. For image goals, all K reference images are supplied jointly at each model decision, without assigning a search order. UniNaVid and NaVid encode them as separate reference-image token blocks, distinct from navigation history, while DualVLN includes them jointly in its multimodal input.

MTU3D retains its RGB-D scene/object memory, learned object-versus-frontier selection, frontier exploration, and geodesic follower. Navigating to a model-selected object destination produces one report; selecting a frontier continues exploration without reporting. Scene memory, visited frontiers, and decision state persist across rounds. Text goals receive compact task instructions and unverified previously selected candidate positions, subject to the native text encoder's context limit. For image goals, the adapter extends the single-image interface to jointly accept K independently encoded CLIP image features. This branch replaces the text input with image features and therefore does not receive textual report-history hints. MTU3D has no native FINISH head: for non-image AllON, the adapter terminates when the learned exploration probability exceeds 0.5 and no unvisited frontier

remains. Otherwise, search continues subject to the episode budget. MTU3D also retains its native navigation-map and geodesic-follower access; this geometric access is distinct from the withheld target annotations and is part of its evaluation setting.

For training-based methods, a public quota triggers termination after r reports for category/attribute ManyON or K reports for an image-goal set. These counters measure issued reports, including incorrect or repeated ones, rather than verified discoveries. For non-image AllON, their completion decision is model-generated, whereas MTU3D uses the frontier-based rule described above.

Training-free methods. Training-free agents (3D-Mem, SCOPE) retain their native sequential-search interface, in which a high-level VLM selects a snapshot/candidate or a frontier to explore next. We organize the instruction into search rounds that reuse the same persistent scene, memory, and map. A snapshot/candidate selection is mapped to a report, and a frontier selection continues native exploration. From the second round onward, the agent's own prior reports are appended as soft de-duplication hints (Prompt 3); these are advisory only—the adapter does not filter candidates, reject repeats, grant compensatory rounds, or return correctness feedback. For AllON only, the high-level choice space is extended with an explicit FINISH option (Prompt 4). For these training-free adapters, ManyON and image goals provide no FINISH, terminating when the report quota or fixed image-set size is reached, or when the budget is exhausted.

Table 3 summarizes the task-level search and termination rules, including the early-termination option available to the three action-generating training-based methods. Multi-category AllON queries are presented as a single joint instruction rather than split into per-category subtasks. Image goals present all K reference images jointly without imposing a search or report order. Navigation, camera-control, reporting, and termination actions consume the same episode-level budget; reporting never resets it. Adaptation prompts for task semantics, per-round instructions, report history, and the AllON stopping criterion are provided in Appendix E. The adaptation layer code is also released as part of our benchmark project.

Table 3: Adaptation of MOST-Bench tasks to native single-goal/GOAT-style interfaces. All search rounds share one episode-level budget and persistent agent state. Quotas count issued reports, not verified successes. No hidden states are provided.

Task	Count given	Search / termination
ManyON (category/attribute)	Quota r	At most one report per round; terminate at r reports, r credits or budget exhaustion.[†]
ManyON (image)	K ref. images	Joint image-goal set; terminate at K reports, K credits or budget exhaustion.[†]
AllON (category/attribute)	Hidden	At most one report per round; terminate through model-issued FINISH, MTU3D's frontier-based rule, or budget exhaustion.
AllON (multi-category)	Hidden	A single joint instruction; otherwise the same reporting and termination rules as above.

[†] UniNaVid, NaVid, and DualVLN may emit a native finish prediction, which is interpreted as method-side early abandonment rather than the benchmark FINISH action. The image-goal setting has a public set size K, irrespective of its internal dataset label.

C.2 Per-Method Input Requirements

The six external methods differ substantially in what sensor inputs they natively consume (Table 4). The adaptation layer feeds each method only its native subset of the episode observation stream (Appendix B): video-based monocular policies (NaVid, UniNaVid, and InternVLA-N1 in its RGB-only DualVLN V2 configuration) receive the RGB stream and the goal text, while methods built on online 3D reconstruction or mapping (MTU3D, 3D-Mem, SCOPE) additionally receive depth and pose. All inputs come from the standard MOST-Bench sensor suite; no method receives privileged information, and methods that do not consume depth or pose simply leave those channels unused.

Table 4: Native input requirements of the evaluated methods. A ✓indicates the input is consumed by the method's native pipeline; a ✗indicates the channel is left unused.

Method	Type	RGB	Depth	Pose
NaVid (Zhang et al., 2024)	trained policy	✓	✗	✗
UniNaVid (Zhang et al., 2025a)	trained policy	✓	✗	✗
InternVLA-N1 DualVLN V2 (Wei et al., 2026)	trained policy	✓	✗	✗
MTU3D (Zhu et al., 2025)	trained policy	✓	✓	✓
3D-Mem (Yang et al., 2025)	training-free agent	✓	✓	✓
SCOPE (Wang et al., 2026)	training-free agent	✓	✓	✓
MOST-Nav (Ours)	training-free agent	✓	✗	✗

C.3 Evaluation Settings and Aggregation

All training-free agents use `Qwen3.8-Flash` as their VLM. Trained policies use existing checkpoints without additional training on MOST-Bench, and InternVLA-N1 uses the DualVLN V2 configuration. For step budgets, action-level policies (NaVid, UniNaVid, InternVLA-N1, MTU3D) and MOST-Nav operate under the standard 800-step budget, while 3D-Mem and SCOPE—whose steps are planner-executed trajectory segments rather than primitive actions—are evaluated under an 80-step budget. This budget follows these methods' own GOAT-style evaluation protocol, which likewise uses shortened per-subtask budgets (e.g., 50 steps per subtask rather than 500 steps), so their step allocation on MOST-Bench is consistent with how they were originally benchmarked. All methods share the same $0.25\,\mathrm{m}$ horizontal-distance criterion for target reports.

All four metrics use equal episode weights over 750 episodes per method. An invalid or redundant report earns no new true-positive credit. Target-count tiers are determined by the required quota in ManyON and the full matching-set size in AllON.

Table 5: Precision and recall by task and goal form (%; 150 episodes per group), complementing the SR/F1 profiles in Figure 4. Bold marks the highest displayed value in each column.

Method	ManyON						AllON			
	Category		Attribute		Image		Category		Attribute	
	P	R	P	R	P	R	P	R	P	R
NaVid	23.60	31.63	25.68	33.06	8.62	11.85	0.66	28.35	0.47	33.47
UniNaVid	26.40	33.83	24.77	34.66	9.64	13.05	0.13	28.03	0.32	33.07
InternVLA-N1	16.87	22.62	18.48	25.90	11.55	15.43	0.43	25.79	0.32	28.96
MTU3D	33.03	35.89	36.56	36.88	28.36	35.74	14.20	36.95	10.26	39.50
3D-Mem	57.95	54.68	50.88	**56.94**	36.69	39.17	29.07	**56.59**	16.92	55.51
SCOPE	**61.98**	**56.58**	52.08	52.52	39.52	**44.94**	34.76	53.65	24.53	51.72
MOST-Nav	50.05	50.70	**52.12**	51.88	**63.60**	44.48	**42.40**	50.70	**54.14**	**57.62**

Table 6 reports precision and F1 for the target-count tiers in Figure 5, which shows SR and recall. Each tier contains 150 ManyON and 100 AllON episodes. MOST-Nav's AllON precision increases from 36.29% on easy episodes to 57.71% on hard episodes, while SR decreases. Higher report accuracy therefore does not imply more frequent task completion across these tiers.

C.4 Complementary Outcome Breakdowns

Table 5 reports precision and recall for the five task and goal groups, complementing the SR/F1 profiles in Figure 4. On ManyON with image goals, MOST-Nav has higher precision than SCOPE (63.60% versus 39.52%) with similar recall (44.48% versus 44.94%), consistent with its higher F1.

For successful episodes in AllON, we additionally measure the distance traveled between the last credited target and the explicit `FINISH` as an indicator of stopping efficiency. The mean/median

Table 6: Precision and F1 by task and required-target count (%), complementing the SR/recall curves in Figure 5. Bold marks the highest displayed value per column.

Method	ManyON						AllON					
	2–4		5–7		8–10		2–4		5–7		8–10	
	P	F1	P	F1	P	F1	P	F1	P	F1	P	F1
NaVid	22.56	23.38	19.90	22.38	15.45	18.43	0.62	0.77	0.51	0.81	0.57	0.91
UniNaVid	28.33	29.13	18.48	21.41	13.99	17.21	0.08	0.16	0.15	0.29	0.45	0.77
InternVLA-N1	21.11	21.92	14.80	17.22	10.99	13.47	0.21	0.40	0.39	0.76	0.53	0.97
MTU3D	34.97	35.49	35.68	36.26	27.29	27.20	8.42	9.67	12.09	12.26	16.18	16.28
3D-Mem	48.36	46.50	51.45	52.44	45.71	43.95	23.02	25.12	24.20	27.81	21.77	23.32
SCOPE	49.71	**48.00**	54.55	**53.39**	49.31	47.06	23.83	27.46	26.95	30.64	38.16	34.64
MOST-Nav	**50.51**	47.99	**59.07**	53.07	**56.19**	**47.14**	**36.29**	**39.56**	**50.81**	**49.17**	**57.71**	**52.24**

distances are 29.67/26.77 m for 3D-Mem, 29.06/28.47 m for SCOPE, and 36.71/32.52 m for MOST-Nav. However, these methods succeed on largely different episode subsets with only three episodes in their intersection. These numbers therefore describe per-method behavior rather than establishing a ranking of stopping efficiency. For future work with higher success rates and larger overlap, post-coverage travel distance can serve as a useful metric of stopping capability.

C.5 Robustness of Success Rate to Excessive Reporting

The two tasks differ in how success responds to excessive reporting. In ManyON, the r-report allowance makes indiscriminate reporting self-defeating. Invalid or redundant reports consume the allowance without contributing to the quota, directly lowering the chance of success. In AllON, however, reports within the budget are unlimited and the agent decides when to issue FINISH, so an agent could in principle inflate its success rate by issuing TARGET_FOUND at every encountered instance, accumulating false-positive reports that are penalized only in precision.

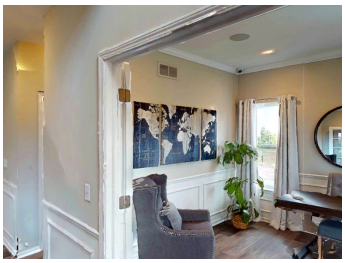
In practice, we observe no evidence of such behavior: the three methods with the highest overall SR (MOST-Nav, SCOPE, and 3D-Mem) are also the three highest-precision methods in the same order, and the best-SR method (MOST-Nav) also achieves the best precision in AllON (Table 2). To guard against such strategies in future work, we define $\mathrm{SR@P}_\tau$, in which an AllON episode counts as successful only if it satisfies the completion rule and its episode-level precision is at least τ. We do not adopt it in this work, as no evaluated method exhibits the behavior it is designed to detect, while the jointly reported precision already exposes it.



(a) Scene-level 3D rendering: example I.

(b) Scene-level 3D rendering: example II.

Figure 8: Scene-level 3D renderings of two MOST-Bench environments. These visualizations summarize the global scene structure for presentation and are not provided to evaluated agents.



(a) Initial egocentric observation.



(b) Candidate-object view.



(c) Candidate-object view.



(d) Candidate-object view.

Figure 9: Egocentric RGB observations from MOST-Bench episodes. The examples cover different task–goal combinations and show the visual diversity, clutter, occlusion, and viewpoint variation encountered by the agent.

D Simulation Environment Visualization

This section provides qualitative examples of how MOST-Bench episodes appear in the simulator. Figure 8 presents two scene-level 3D renderings of the environments underlying the benchmark. The renderings highlight the multi-room layouts, object density, and spatial diversity of the simulation scenes. These global views are included only for visualization: during evaluation, agents receive only egocentric sensor observations and never have access to a top-down map, full-scene rendering, or privileged global geometry.

Figure 9 shows four egocentric RGB observations spanning different task and goal settings, including the initial view of an image-goal episode and candidate-object views from category, attribute, and image goals. These views illustrate the egocentric visual input available to the agent. Targets are embedded in furnished, cluttered indoor scenes and may be partially occluded, viewed at an angle, or surrounded by visually similar distractors.

E Prompts

This section collects model prompts used in this work. Prompt 1 is the attribute judge that turns coarse HM3D categories into fine-grained attribute groups during construction (Appendix A.2); Prompt 2 is the instruction generator that verbalizes each query program (c, r) into agent-facing descriptions d (Appendix A.3); Prompts 3 and 4 are the adaptation prompts used at evaluation time (Appendix C.1), giving the shared task-semantics and per-round instructions and the AllON stopping controller used by agents with a high-level VLM decision step. The image track also uses individually marked reference images. Placeholders in braces (e.g., {category}) or angle brackets (e.g., <TARGET_LABEL>) are filled per query/episode at generation time.

Prompt 1: attribute judging prompt (`gpt-5.6-terra`)

```
You are a fine-grained visual attribute judge for an embodied
↪  navigation benchmark.

Input:
```

```
You will see multiple viewpoints of the SAME target object. Each
↪  viewpoint contains two images:
- RGB: the original color image.
- overlay: the same view with the target object highlighted in red.

Rules:
- Use the overlay only to identify which object is the target. Judge
↪  its appearance from the RGB image; do not let the red highlight
↪  affect your judgment of color, material, or other appearance.
- Ignore all other objects, the background, and room layout.
- Some viewpoints may be render-failed or corrupted (blurry, black, or
↪  broken geometry). Prefer the clearer viewpoints and ignore the
↪  broken ones.

Target:
- Category: {category}
- Attribute: {attribute}
- Attribute definition: {description}

Question:
Does the highlighted target object have the attribute "{attribute}"?

Answer:
- "yes": the target has the attribute.
- "no": the target does not have the attribute.
- "uncertain": none of the viewpoints is clear enough to judge (too
↪  unclear, too small, heavily occluded, or broken). Do not guess.

Return JSON only:
{"answer": "yes"|"no"|"uncertain"}
```

Prompt 1 is run with `temperature`$= 0$ and enforced JSON output. The model receives up to three viewpoints, each provided as an RGB image plus an overlay in which the target instance is highlighted in red.

Prompt 2: language-goal generation prompt (`gpt-5.6-terra`)

```
You are writing natural-language navigation instructions for an
↪  embodied object navigation benchmark.

The instruction must describe what object(s) the agent should find
↪  based only on the query program.
Do not mention object IDs, query IDs, JSON fields, metadata, or
↪  implementation details.
Use natural English instructions a person might give to a robot.

Query metadata:
- target_mode: {target_mode}
- target_count: {target_count}
- query_program: {query_program}

Field meanings:
- "category" means the target object category.
- "attribute" means a fine-grained visual or content label that
↪  narrows the category (for example "people", "landscape",
↪  "animal"). When present, the instruction must refer to both the
↪  category and this attribute.
  {attribute_definition_line}
- "category" with op "in" and a list of values means the target set is
↪  the union of those categories, and target_mode is always "all":
↪  the instruction must ask for every object of every listed
↪  category, for example "Find all the cups and all the tables." Do
↪  not turn it into "cups or tables": the agent has to find both
↪  kinds, not pick one.
```

```
- "category_attribute" with op "in" and a list of [category,
↪  attribute] pairs works the same way, but each member of the union
↪  carries its own attribute, for example
↪  [["chair","leather"],["picture","people"]] becomes "Find all the
↪  leather chairs and all the pictures that contain people."

HARD RULE: keep the "category" word exactly as it appears in the query
↪  program; do not replace it with a synonym or reword it. For
↪  example, if the category is "couch", write "couch", not "sofa"; if
↪  it is "tv", write "tv", not "television". This rule applies only
↪  to the category word, not to the attribute.

When phrasing the attribute, use natural language rather than raw
↪  identifiers, for example "black_and_white" -> "in black and
↪  white", "has_glass" -> "with glass", "only_tablecloth" -> "with
↪  nothing on it except a tablecloth".

Target mode guidance:
- If target_mode is "many", the instruction must explicitly ask for
↪  exactly target_count distinct objects matching the category (and
↪  attribute if present). Use the number in natural language, for
↪  example "two" or "3"; do not use vague words like "some",
↪  "several", or "multiple".
- If target_mode is "all", the instruction should ask for all objects
↪  of the category (with the attribute constraint if one is present).

Return strict JSON only:
{
  "natural_language": [
    "..."
  ]
}

Generate exactly {num_descriptions} diverse paraphrases.
```

Prompt 2 is run with `temperature`$= 0.8$, producing 3 paraphrases per query program from which one is sampled per episode. The call uses the system message "*You generate concise benchmark navigation instructions. Return JSON only.*" together with the user prompt above. The query metadata fields (`target_mode`, `target_count`, `query_program`) encode the specification c and requirement r; `target_mode` is one of `many`/`all`, {`num_descriptions`} is set to 3.

Prompt 3: task semantics, instruction, and report history

```
[Initial goal line, shown only in round 1, one of:]
Goal (object category name): "<TARGET_LABEL>".
Goal (language instruction): "<INSTRUCTION>".
Goal (reference images): "<K> individually marked reference images are
↪  attached."

[Initial mode line, shown only in round 1:]
Mode: many. Find <K> distinct physical instances of the matching
↪  target. Each physical instance may be reported at most once; do
↪  not report the same physical instance more than once.
Mode: all. Find every matching target instance. Each physical instance
↪  may be reported at most once; do not report the same physical
↪  instance more than once.

[Per-round instruction, one of:]
- many:
Search round <ROUND_INDEX> of <K>. In this round, search for and
↪  report at most one matching physical instance.
- all:
```

```
In this round, search for and report at most one candidate matching
↪  physical instance. The total number of matching instances is
↪  hidden. Do not infer a target count from the number of search
↪  rounds or reports.
- many-image:
<K> reference images are provided together as one image-goal set.
↪  Search for the pictured physical instances jointly; image order
↪  does not define search or report order. In this round, search for
↪  and report at most one candidate corresponding to any reference
↪  image. The episode requires exactly one report per reference
↪  image.

[Report history, appended from the second round onward when reports
↪  exist:]
Previous reports (agent-side memory only; unverified):
- Report 1: agent_object_id=<AGENT_OBJECT_ID>,
↪  agent_label=<AGENT_LABEL>, approximate_position=<POSITION>
- Report 2: ...
This history contains no ground-truth correctness, success, failure,
↪  or duplicate feedback. Use it only to avoid intentionally
↪  reporting the same agent object id or physical instance again.
```

Prompt 4: AllON episode-level stopping controller (system + user)

```
[System]
You are the episode-level stopping controller for an indoor search
↪  task.

Decide whether the agent should continue exploring for another
↪  distinct physical instance matching the task, or finish the
↪  episode.

Choose Continue when further exploration is warranted because another
↪  matching instance may reasonably remain.

Choose Finish when the explored evidence is sufficient and you believe
↪  no additional distinct matching instance remains.

You are responsible for the episode-level stopping decision. If you
↪  choose Continue and action budget remains, the adapter must start
↪  another search round. The episode must not end merely because the
↪  number of search rounds or reports has reached the hidden
↪  ground-truth target count. Normal termination is allowed only when
↪  you choose Finish or when the episode action budget is exhausted.

Use only the agent's own observations, exploration history, and report
↪  history. Do not assume or infer a hidden target count. The
↪  presence of a frontier does not automatically require Continue,
↪  and the absence of a frontier does not automatically require
↪  Finish.

The report history records what the agent reported. It contains no
↪  ground-truth correctness, success, failure, or duplicate feedback.

Output exactly one word:
Continue
or
Finish

[User]
Original task:
{original_instruction}
```

```
Most recent search-round outcome:
{round_outcome}

Agent report history:
{report_history}

Recent exploration summary:
{exploration_summary}

Current scene evidence:
- Current scene-graph summary: {scene_graph_summary}
- Current frontier count: {frontier_count}
- Native steps used in the most recent search round: {native_steps}

Current egocentric observations and frontier observations are attached
↪  when available. Based only on this evidence, decide whether
↪  another search round is worthwhile.

If you answer Continue and action budget remains, another search round
↪  will be run. Continue never ends the episode by itself.

Answer exactly one word: Continue or Finish.
```

Only a response whose first word is exactly `Finish` ends the episode; any other output is treated as `Continue`.