SSLONGNAV: BENCHMARKING REPEATED COMPLETION OF STRUCTURED SUBTASKS IN LONG-HORIZON INSTANCE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In long-horizon navigation, success rates for individual visits do not reveal how reliably agents complete successive requests involving multiple object instances. We introduce Service-Structured Long-Horizon Instance Navigation (SSLongNav), which groups ordered instance visits into *structured subtasks* representing the navigation requirements of service-inspired requests, with shared service goals and per-visit intents. Whether a visit succeeds or fails, the agent proceeds to the next prescribed visit from its current pose. Built on synthetic HSSD and scanned HM3D scenes, SSLongNav comprises 450 episodes across 326 scenes, with 3,669 structured subtasks and 12,755 instance visits. Our primary metric, Strict Subtask Completion (SSC), measures the fraction of subtasks in which every prescribed visit succeeds. To establish a strong baseline on this benchmark, we further develop PEARS, which uses shared subtask context to guide search and combines selective evidence persistence, typed evidence arbitration, and risk-aware submission. On scene-disjoint evaluation splits, PEARS achieves SSC of 5.75% on HSSD and 13.94% on HM3D, exceeding the strongest benchmark-adapted competitors by 5.09 and 7.09 percentage points, respectively. Results reveal a substantial gap between partial progress and full completion, while documenting complete multi-visit requests after earlier failures in the same run.

1 INTRODUCTION

Long-horizon navigation studies how embodied agents pursue successive goals within an environment, often using experience accumulated during earlier execution (Wani et al., 2020; Krantz et al., 2023; Khanna et al., 2024b). In service settings, a single request may require visiting several designated object instances (Zhang et al., 2024), and its navigation requirements are fulfilled only when all required visits succeed. An agent may therefore reach many individual targets while leaving requests partially fulfilled. Evaluating long-horizon navigation for sustained service thus requires examining how consistently agents complete entire requests throughout an ongoing run.

MultiON studies map memory through navigation to ordered object-goal sequences (Wani et al., 2020). IVLN examines experience reuse across tours of instruction-following episodes (Krantz et al., 2023), while GOAT-Bench evaluates successive category-, language-, and image-specified goals with persistent scene memory (Khanna et al., 2024b). LHPR-VLN evaluates multi-stage navigation and SG3D-Nav ordered visits to task-relevant object instances, both assessing joint completion within a task (Song et al., 2025; Zhang et al., 2024). However, evaluating individual visits or a single multi-stage task does not reveal how many complete requests an agent fulfills as execution continues from the states left by earlier successes and failures.

For example, two successive requests to bring the blue cup to the dining table and the remote to the coffee table each involve an ordered pair of navigation visits (Figure 1, top). We represent each request's navigation requirements as a *structured subtask*, with a shared service goal and per-visit intents providing context for ordered instance visits that must all succeed. Execution continues through all remaining visits from the realized pose after success or failure, preserving later requests as separate completion opportunities. An aggregate visit success rate ignores request boundaries, whereas whole-episode success treats every partially completed request stream as a failure. Both

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.

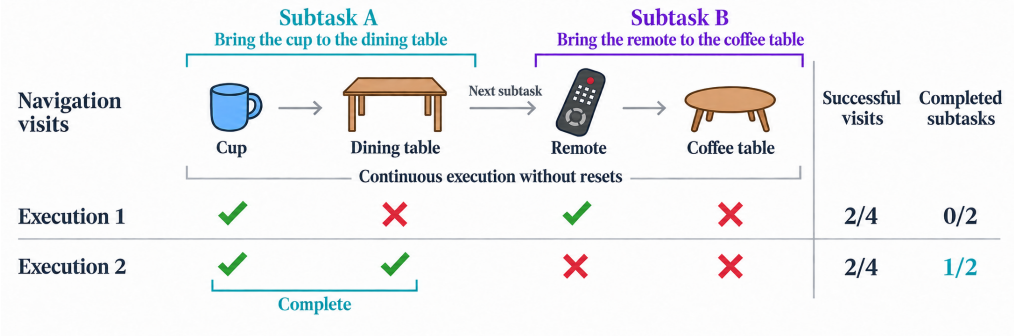


Figure 1: **Repeated strict subtask completion. Top:** Two service-inspired subtasks form an ordered visit sequence without physical resets. **Bottom:** Both executions succeed on two of four visits and fail overall, yet complete different numbers of subtasks, each requiring both visits to succeed.

executions in Figure 1 (bottom) succeed on two of four visits and fail overall, yet only one completes a structured subtask. The limitations of visit-level and whole-episode evaluation motivate a benchmark for repeated strict subtask completion during continuous navigation.

To this end, we introduce Service-Structured Long-Horizon Instance Navigation (SSLongNav), a benchmark built on synthetic HSSD and scanned HM3D scenes (Khanna et al., 2024a; Ramakrishnan et al., 2021). SSLongNav comprises 450 episodes across 326 scenes, with 3,669 structured subtasks and 12,755 instance visits. We construct episodes from visible, reachable object instances, organizing prescribed visits into separately scored subtasks with instance descriptions, shared service goals, and per-visit intents. Episodes vary in both the number of successive subtasks and the number of visits required within each subtask. The dataset also includes distinct instances of the same category and cross-subtask revisits, supporting the study of instance discrimination and evidence reuse alongside complete-request performance. Each prescribed visit has its own action budget. Our primary metric, Strict Subtask Completion (SSC), measures the fraction of subtasks in which all prescribed visits succeed. Complementary metrics measure visit success, partial progress within subtasks, and strict completion weighted by path efficiency.

We introduce Persistent Evidence Arbitration with Risk-Aware Submission (PEARS), a navigation framework for repeated completion of structured subtasks. PEARS uses shared service goals and per-visit intents to guide contextual retrieval and candidate verification, while related objects within the current subtask provide anchors for instance search. Selective persistence carries reusable scene evidence across visits and subtasks, refreshing target-specific queries and decision state as goals change. Typed evidence arbitration distinguishes active-target hypotheses from search hypotheses such as contextual anchors and unexplored regions, allowing both types to guide navigation while reserving STOP proposals for active-target hypotheses. Because an incorrect STOP can preclude completion of an entire subtask, risk-aware submission checks source-specific semantic, geometric, and reliability evidence before authorizing STOP, with rejected proposals prompting further observation or recovery. On scene-disjoint evaluation splits, PEARS achieves SSC of 5.75% on HSSD and 13.94% on HM3D, improving over the strongest benchmark-adapted competitor in each scene family by 5.09 and 7.09 percentage points, respectively. Even with these gains, most structured subtasks remain incomplete, highlighting the challenge of repeated request completion on SSLongNav.

Our contributions are threefold:

- We introduce SSLongNav, a benchmark comprising 450 episodes across 326 scenes from HSSD and HM3D, with 3,669 structured subtasks and 12,755 instance visits. It evaluates strict completion separately for successive service-structured subtasks during continuous navigation.
- We propose PEARS, which uses shared subtask context for instance search and combines selective evidence persistence, typed evidence arbitration, and risk-aware submission to support repeated subtask completion.
- Comparisons with adapted baselines and system ablations characterize PEARS's gains in strict completion. Analyses expose the gap between partial progress and full completion,

and document complete multi-visit requests after earlier failures during continuous execution.

2 Related Work

Single-goal embodied navigation studies category and open-vocabulary targets (Yokoyama et al., 2024a;b; Yin et al., 2025) and the grounding and verification of specific instances (Krantz et al., 2022; Barsellotti et al., 2024; Lei et al., 2024; Taioli et al., 2025; Jang & Kim, 2026). Vision-and-language navigation grounds route instructions in discrete and continuous environments (Anderson et al., 2018; Krantz et al., 2020).

Long-horizon navigation includes ordered category goals in MultiON (Wani et al., 2020), experience reuse across oracle-connected tours in IVLN (Krantz et al., 2023), and persistent multimodal goal streams in GOAT-Bench (Khanna et al., 2024b). LHPR-VLN (Song et al., 2025), SG3D (Zhang et al., 2024), and LH-AVLN (Chen et al., 2026) evaluate multi-stage tasks, sequential instance grounding, and spatial-audio multi-goal navigation, respectively. LMEE (Wang et al., 2026a) and LENL (Wang et al., 2026b) study memory-based exploration and continual navigation learning. SS-LongNav evaluates strict completion separately for successive service-structured requests within a continuous run (Section 4.3).

3 Task Definition and Evaluation Protocol

Task structure. SSLongNav episodes start from a single pose in one scene and contain M ordered structured subtasks, each representing a service-inspired request's navigation requirements. Subtask m specifies a shared service goal and $L_m \geq 1$ ordered instance visits, each with a description and intent. At each visit, the agent must reach the designated instance's vicinity and issue STOP. Instances may recur across subtasks. Appendix A.2 gives formal definitions; Section 4 details construction.

Continuous execution. The agent receives RGB-D, simulator localization, the active instance's stable ID and public specification, and the current subtask's shared context and ordered target information. IDs serve as identity tokens; target positions, valid viewpoints, and evaluator success labels remain private. Each visit ends on STOP, timeout at 500 actions, or ABANDON. If visits remain, execution advances from the realized pose without resetting. A failed visit precludes strict completion of its subtask; all remaining visits still proceed in the prescribed order, and later subtasks are attempted and scored separately. Appendix A gives the full interface and termination rules.

Hierarchical metrics. Visit-level success measures individual navigation performance but does not indicate whether all visits required by a request succeed. Whole-episode success requires every request to succeed and cannot distinguish how many requests are completed in unsuccessful episodes. We therefore introduce a hierarchical evaluation framework covering strict subtask completion, partial progress, visit success, and path efficiency, with Strict Subtask Completion (SSC) as the primary metric. For one episode, let $S_{m,l} \in \{0, 1\}$ indicate whether visit (m, l) ends with STOP within $1\,\mathrm{m}$ evaluation distance of its designated instance's private scoring set (valid viewpoints and stored goal position; Appendix A.3). Let D_m sum reference distances recomputed from each visit's actual starting pose and P_m sum traveled distances for subtask m.

$$
\begin{aligned}
C_m &= \prod_{l=1}^{L_m} S_{m,l}, \qquad \mathrm{SSC} = \frac{1}{M}\sum_{m=1}^{M} C_m, \qquad \mathrm{SP} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{L_m}\sum_{l=1}^{L_m} S_{m,l}, \\
\text{SS-SPL} &= \frac{1}{M}\sum_{m=1}^{M} C_m \frac{D_m}{\max(P_m, D_m)}, \qquad \mathrm{IVC} = \frac{\sum_{m=1}^{M}\sum_{l=1}^{L_m} S_{m,l}}{\sum_{m=1}^{M} L_m}.
\end{aligned}
\tag{1}
$$

SSC measures the fraction of subtasks in which all prescribed visits succeed. Subtask Progress (SP) averages the fraction of successful visits within each subtask, Instance-Visit Completion (IVC) measures overall visit success, and Strict Subtask SPL (SS-SPL) weights strict completion by path efficiency. SS-SPL measures path efficiency relative to visit starts induced by the agent's realized execution. Evaluation-split results weight all subtasks equally for SSC, SP, and SS-SPL, and all visits equally for IVC.

4 SSLONGNAV: BENCHMARK CONSTRUCTION AND CHARACTERIZATION

4.1 CONSTRUCTION OF SERVICE-STRUCTURED EPISODES

Source-specific visibility and reachability checks identify navigable targets in HSSD and HM3D (Khanna et al., 2024a; Ramakrishnan et al., 2021), each with a stable identity and valid viewpoints (Figure 2). For both scene families, we use Qwen3.5-9B (Qwen Team, 2026) on rendered target views to generate instance descriptions that capture appearance and local context. To form subtasks, we focus on local service activities, such as organizing a desk area, whose navigation requirements involve multiple objects within a shared local area. We use spatial locality as a prior for forming instance groups from the relative positions of their navigable viewpoints. Groups are ordered from the episode start, and a spatial visit order is prescribed within each group (Figure 2(b)). Pairing a singleton with a nearby instance that also appears in another subtask introduces cross-subtask revisits. The resulting fixed sequence provides the same ordered instance requests to all evaluated agents.

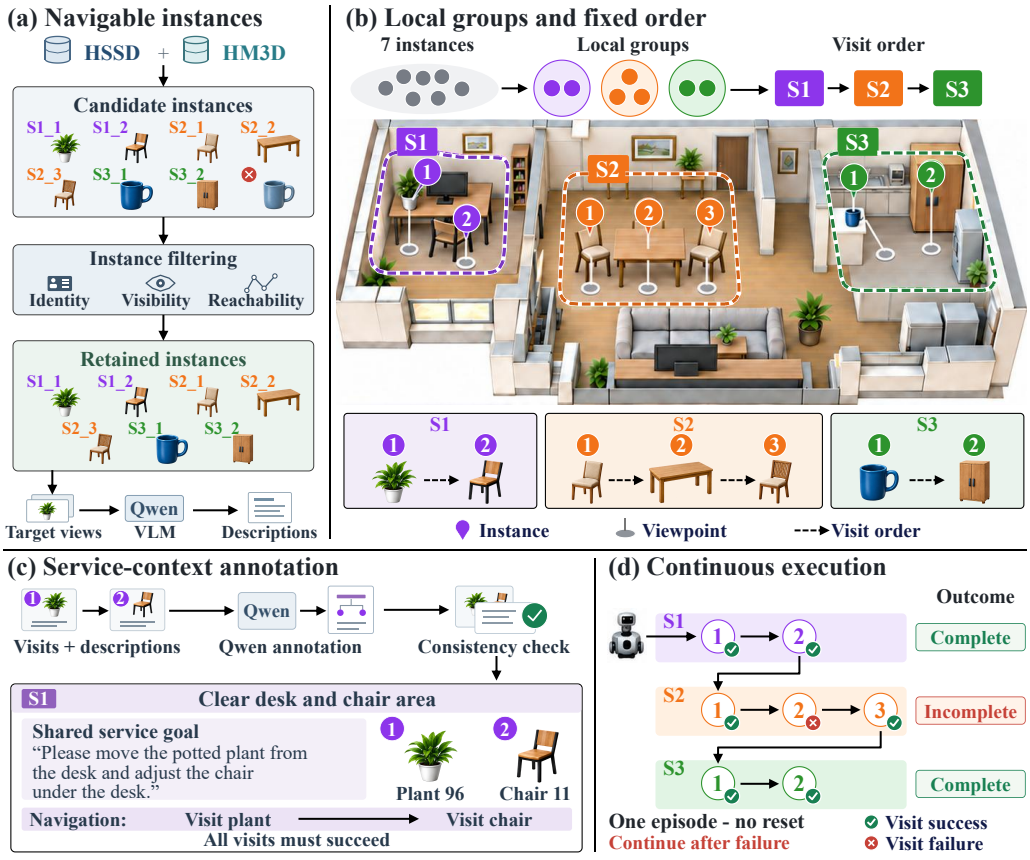


Figure 2: SSLongNav construction: (a) grounded navigable instances, (b) spatial grouping and visit ordering, (c) shared service goals and aligned visit intents, and (d) continuous execution with separate subtask outcomes. The S1 goal and visit sequence are from the benchmark; the layout, other groups, viewpoints, and outcomes are schematic. Dashed arrows indicate visit order.

After establishing group membership and order, we condition Qwen on each multi-visit group's instance categories and descriptions to generate a shared service-inspired goal linking its visits to a common activity, with one intent specifying each visit's role (Figure 2(c)). Multi-view inspection checks instance grounding and descriptions, while manual review checks that all visits fit the shared goal and each intent matches its assigned instance. For example, the plant–chair visits in *Clear desk and chair area* capture the navigation component of moving a plant from a desk and adjusting the chair beneath it. We concatenate the subtasks into an episode with one initial pose and sepa-

rate completion boundaries (Figure 2(d)), following the continuous-execution protocol in Section 3. Appendices B and B.7 provide the filtering, grouping, annotation, and review details.

4.2 Dataset Scale and Structural Characteristics

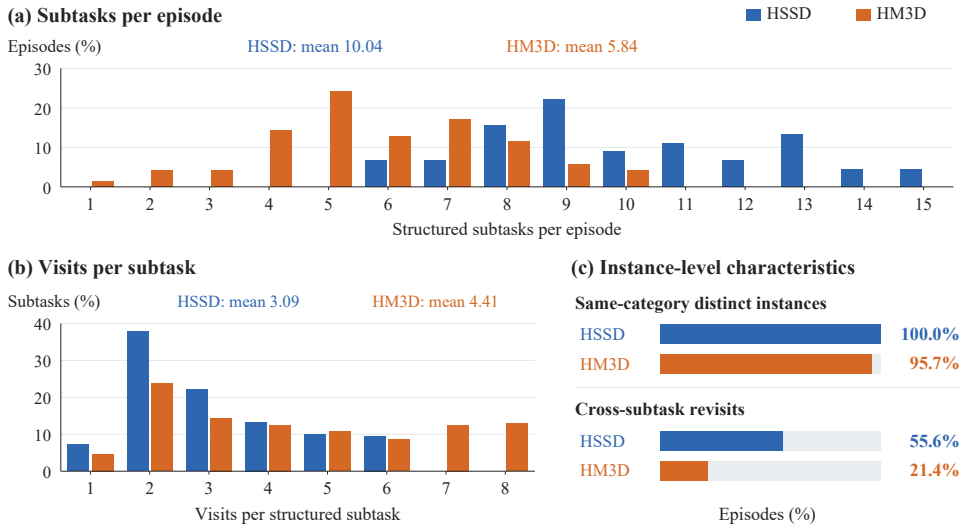


Figure 3: Structure of the SSLongNav evaluation split: HSSD (45 episodes, 452 subtasks) and HM3D (70 episodes, 409 subtasks). Percentages are normalized within each scene family: (a) episodes by subtask count, (b) subtasks by visit count, and (c) episodes with same-category distinct instances or cross-subtask revisits. Full background tracks in (c) denote 100%.

SSLongNav contains 450 episodes across 326 scenes, with 3,669 structured subtasks and 12,755 prescribed instance visits, including revisits. The 335 training and 115 evaluation episodes are scene-disjoint within each source. Across both splits, retained targets cover 28 HSSD categories and 350 HM3D source-defined category labels.

Across all splits, episodes average 8.15 structured subtasks and 28.34 instance visits; 94.8% of subtasks require multiple visits. On average, HSSD evaluation episodes contain more subtasks, whereas HM3D evaluation subtasks contain more visits (Figure 3(a,b)), emphasizing repeated completion across requests and sustained success within requests, respectively. Nearly all evaluation episodes request distinct instances of the same category, highlighting instance discrimination. Cross-subtask revisits support using earlier observations to guide later searches (Figure 3(c)). Appendix B reports full statistics (Table 5); Appendix B.2 defines counting conventions.

Table 1: Joint completion requirements and execution after failure in representative navigation benchmarks.

Benchmark	**Completion structure**		**Execution continuity**
	Visits required for joint success	Repeated multi-visit groups per run	After an unsuccessful visit
MultiON	All goals in the episode	—	Episode ends on incorrect FOUND or timeout
IVLN	—	—	Oracle completion and repositioning
GOAT-Bench	—	—	Next goal after timeout; no reset
LHPR-VLN	All stages of one task	—	Not specified
SG3D-Nav	All steps of one task	—	Not specified
SSLongNav	**All visits in each structured subtask**	**Yes**	All remaining visits and later subtasks; **current pose retained**

Repeated groups are separately scored multi-visit units within one run. GOAT-Bench uses "subtask" to denote a single goal. —: no corresponding multi-visit level; goal-level metrics may exist. Not specified: after-failure pose and advancement are unreported in the cited protocol. Details: Appendix B.3.

4.3 Comparison with Existing Benchmarks

Table 1 compares joint-success units and execution continuity. MultiON (Wani et al., 2020) scores all episode goals jointly; LHPR-VLN (Song et al., 2025) and SG3D-Nav (Zhang et al., 2024) score all stages or steps of one task. GOAT-Bench (Khanna et al., 2024b) scores individual goals and continues after timeout without a reset, while IVLN (Krantz et al., 2023) links episodes through oracle completion and repositioning. SSLongNav separately scores successive service-structured groups of instance visits. Execution continues from the realized pose after success or failure through all remaining visits, including those within failed subtasks. Earlier execution therefore shapes both the starting conditions and available evidence for later requests. Appendix B.3 provides detailed comparisons.

5 PEARS: A Reference Navigation Framework

PEARS uses the active instance specification and shared subtask context to guide evidence acquisition. At each observation, it updates reusable scene evidence and target-specific hypotheses, then selects navigation, scanning, or a STOP proposal. STOP proposals undergo source-specific validation. Selective persistence, typed evidence arbitration, and risk-aware submission connect this cycle across visits and subtasks (Figure 4).

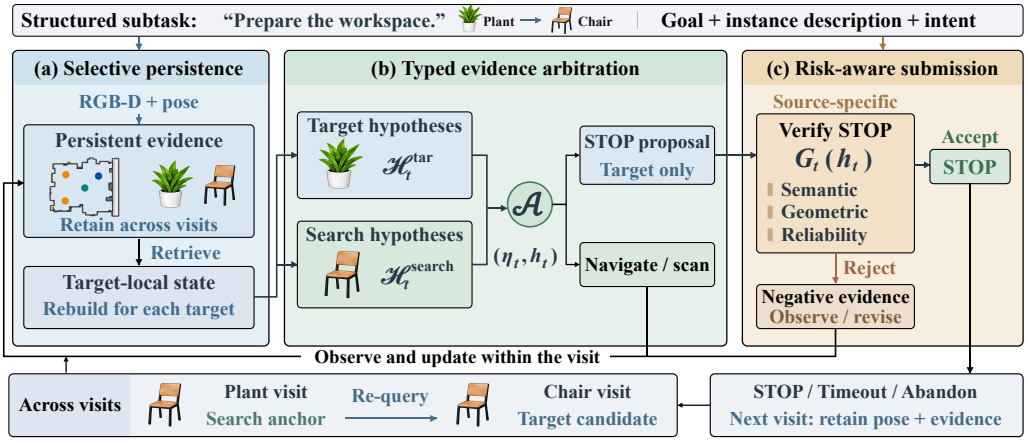


Figure 4: Decision cycle of PEARS. Arbitration $\mathcal{A}$ uses target and search hypotheses; only active-target hypotheses can propose STOP, validated by $G_t(h_t)$. Rejected proposals guide recovery; visit transitions preserve pose and reusable evidence.

Selective Persistence. PEARS retains spatial maps, visual features, grounded-instance records, and execution history across visits. At each target switch, it rebuilds queries, candidates, beliefs, and local plans around the new instance specification, allowing accumulated observations to support the next visit. Instance retrieval uses the active category and description. Shared service goals and per-visit intents condition contextual retrieval and verification, while categories of other objects in the current subtask provide cues for retrieving search anchors. In a plant–chair subtask, an observed chair can guide the search for the plant. When the chair becomes the active target, the same stored observations support retrieval and grounding under its instance specification. Appendix C.2 details the state lifecycle, memory updates, and retrieval rules.

Typed Evidence Arbitration. PEARS represents active-target hypotheses $\mathcal{H}_t^{\text{tar}}$ and search hypotheses $\mathcal{H}_t^{\text{search}}$ in a unified typed belief. Target hypotheses draw on detections or recalled objects; search hypotheses include contextual anchors and unexplored regions. For candidate i, navigation and STOP utilities are

$$\begin{aligned} U_i^{\text{nav}} &= w_{\text{tar}} p_i^{\text{tar}} + w_{\text{id}} p_i^{\text{id}} + w_{\text{vis}} p_i^{\text{vis}} + w_I I_i - w_C C_i - w_s R_i^{\text{stop}} - w_l R_i^{\text{loop}}, \\ U_i^{\text{stop}} &= v_p p_i^{\text{stop}} - v_s R_i^{\text{stop}} - v_l R_i^{\text{loop}}. \end{aligned} \tag{2}$$

Here, $p_i^{\mathrm{tar}}, p_i^{\mathrm{id}}, p_i^{\mathrm{vis}}, p_i^{\mathrm{stop}}$ score target relevance, identity, visibility, and STOP support; $I_i, C_i, R_i^{\mathrm{stop}}, R_i^{\mathrm{loop}}$ encode information gain, path cost, false-STOP risk, and loop risk. These are heuristic scores, with fixed positive weights $w_\cdot, v_\cdot$. A nearby target with sufficient STOP support can propose STOP; uncertain evidence can trigger scanning. Continued search prioritizes target navigation, region navigation, then exploration. At step t, the arbiter $\mathcal{A}$ applies these priorities and candidate-role constraints to select candidate h_t and a navigation, scanning, or STOP-proposal intent η_t; a_t is the executed action and G_t the submission gate:

$$\begin{aligned} (\eta_t, h_t) &= \mathcal{A}\big(\mathcal{H}_t^{\mathrm{tar}}, \mathcal{H}_t^{\mathrm{search}}\big), \\ a_t = \mathrm{STOP} &\Longrightarrow \eta_t = \mathrm{StopProp} \wedge h_t \in \mathcal{H}_t^{\mathrm{tar}} \wedge G_t(h_t) = 1. \end{aligned} \tag{3}$$

Appendix C.3 specifies score construction, coefficient values, thresholds, and controller priorities.

Risk-Aware Submission and Recovery. The gate G_t assesses support for submitting the active target at the current pose using source-specific semantic, geometric, and reliability checks. Semantic support includes vision-language verification of current or stored views against the instance description and service context. Local candidates draw on current target evidence and local geometry. Recalled-instance proposals draw on record trust and applicable stored-view or full-goal evidence. Accepted proposals authorize STOP. Rejected proposals update negative evidence and cooldowns to guide further observation, navigation, and candidate revision; repeated rejection can suppress a candidate. Each issued STOP ends the visit and is scored separately by the evaluator. STOP, timeout, or abandonment under budget or search-exhaustion rules advances the prescribed sequence while preserving the realized pose and reusable evidence. Appendix C.4 specifies the submission gates and recovery rules; Appendix C.5 reports domain-specific settings.

6 Experiments

Our experiments examine complete-request performance, sustained success, and completion after earlier failures on SSLongNav. We compare adapted navigation baselines and relate system ablations to PEARS's selective persistence, typed evidence arbitration, and risk-aware submission.

Table 2: Method profiles and scene-disjoint results on SSLongNav. SSC/SP/IVC: %; SS-SPL: $\times 100$. All metrics: higher is better. Bold: best per domain. †: camera-pitch actions retained.

		HSSD				HM3D			
Method	**Original task**	**SSC**	SS-SPL	SP	IVC	**SSC**	SS-SPL	SP	IVC
UniGoal-LT (CVPR 2025)	Universal zero-shot goal navigation	0.44	0.41	4.40	5.29	2.93	1.15	9.04	8.53
Context-Nav-LT (CVPR 2026)	Text-goal instance navigation	0.66	0.24	3.04	3.29	0.98	0.05	8.86	9.36
MemoryExplorer-LT (CVPR 2026)	Long-term embodied exploration	0.44	0.16	4.06	4.94	2.20	1.15	9.08	9.09
GOAT SenseAct-NN-LT† (CVPR 2024)	Multimodal lifelong navigation	0.44	0.44	2.13	2.58	6.85	5.32	18.18	20.33
PEARS (Ours)	Structured-subtask navigation	**5.75**	**3.63**	**18.45**	**20.03**	**13.94**	**7.16**	**42.35**	**44.71**

6.1 Evaluation Setup

Evaluation uses scene-disjoint splits of 45 HSSD episodes (452 subtasks, 1,398 visits) and 70 HM3D episodes (409 subtasks, 1,805 visits). Termination, scoring, and aggregation follow Section 3, with SSC as the primary metric.

Baselines cover universal goal navigation with UniGoal (Yin et al., 2025), instance grounding with Context-Nav (Jang & Kim, 2026), memory-based exploration with MemoryExplorer (Wang et al., 2026a), and lifelong navigation with GOAT SenseAct-NN (Khanna et al., 2024b) (Table 2). The

"-LT" adaptations share a target-transition interface and evaluator, with historical state retained according to each method's implementation. GOAT retains its trained camera-pitch actions (†); other methods use the standard five-action interface. PEARS requires no SSLongNav-specific training or fine-tuning. Appendix D.1 details navigation mechanisms, model backends, and adapters.

6.2 Complete-Request Performance

Table 2 shows that PEARS achieves 5.75% SSC on HSSD and 13.94% on HM3D, exceeding the strongest SSC baselines, Context-Nav-LT and GOAT SenseAct-NN-LT, by 5.09 and 7.09 percentage points, respectively. PEARS also attains the highest SP, IVC, and SS-SPL point estimates in both domains. The four paired SSC intervals against Context-Nav-LT and GOAT lie above zero (Appendix D.2), supporting gains in complete-request performance under the benchmark adaptations.

Substantial partial progress remains outside completed requests: PEARS reaches 18.45% and 42.35% SP, respectively, while most subtasks remain incomplete. SP and SSC weight the same subtasks, so their gap measures progress contributed by incomplete requests. Repartitioning the frozen visit streams also preserves PEARS's highest mean SSC under all three tested random schemes (Appendix D.5).

6.3 Analysis of Structured-Subtask Completion

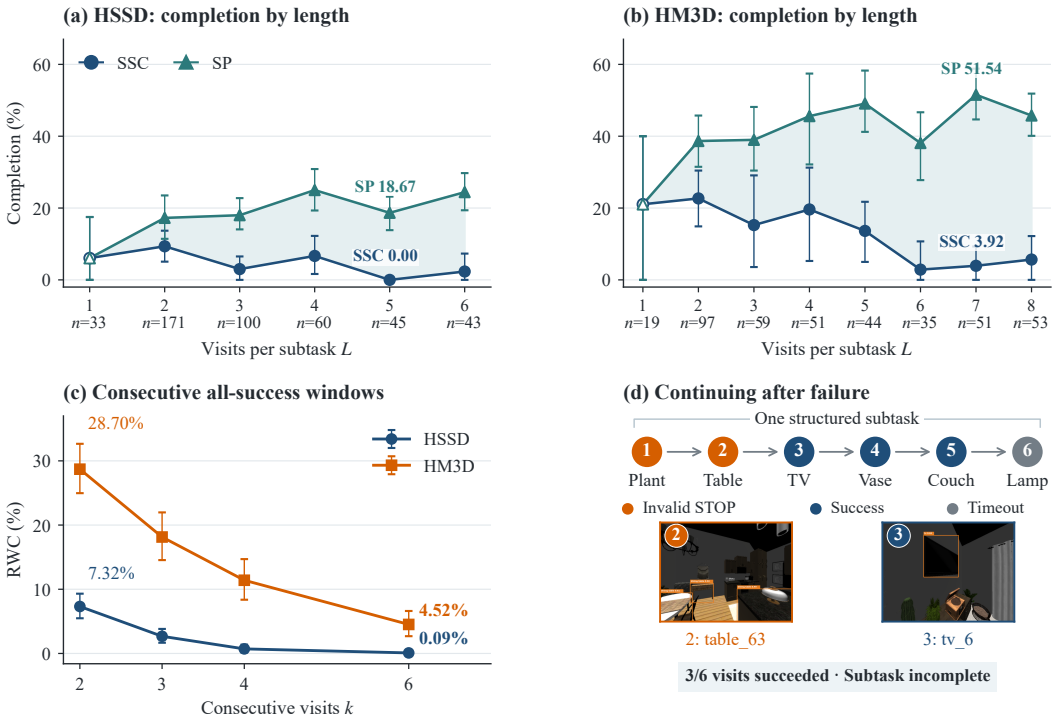


Figure 5: Challenges of repeated subtask completion. (a–b) PEARS SP and SSC by subtask length; n counts subtasks and bars show scene-clustered descriptive 95% bootstrap intervals. Singleton service text is templated (hollow $L = 1$ markers); multi-visit text is Qwen-generated. (c) RWC_k is the fraction of consecutive k-visit windows within episodes with all visits successful; bars show the supplied lower and upper bounds. (d) Outcomes of a six-visit HSSD subtask and views from visits 2–3 illustrate continuation after failure (full trace in Appendix F).

Completing groups and sustaining success. Figure 5(a–b) localizes the progress–completion gap within multi-visit requests. PEARS completes none of the 45 five-visit HSSD subtasks despite 18.67% SP; the 51 seven-visit HM3D subtasks reach 51.54% SP but only 3.92% SSC. These strata differ in targets and context as well as length (Appendix D.3). Rolling-window completion (RWC_k) measures the fraction of contiguous k-visit windows within episodes with every visit successful. Only 0.09% of six-visit windows succeed on HSSD and 4.52% on HM3D (Figure 5(c)). Thus,

sustained success remains difficult even when assessed without prescribed subtask boundaries; exact window counts appear in Appendix D.4.

Completing later requests after failure. Figure 5(d) shows three successful visits after two invalid STOPs within an incomplete six-visit request. Beyond this partial progress, PEARS completes 19 HSSD and 37 HM3D multi-visit subtasks after an earlier subtask in the same episode has failed (Appendix D.4). These later completions demonstrate the value of retaining separate request outcomes throughout continuous execution: an earlier incomplete request does not erase later completed ones. Appendix F presents qualitative examples from separate reruns.

6.4 Ablation Study

Table 3 examines HM3D configurations associated with PEARS's three mechanisms. These system-level interventions can alter coupled control paths; Appendix D.6 specifies their scope.

Table 3: PEARS configurations on the HM3D evaluation split (%). SSC measures complete subtasks. IVC, Invalid STOP, Timeout, and Abandon are percentages of all visits and sum to 100% up to rounding.

Variant	**SSC** ↑	IVC ↑	Invalid STOP	Timeout	Abandon
Full configuration	**13.94**	**44.71**	28.31	6.04	20.94
No instance memory	10.76	31.25	26.76	16.73	25.26
No visual memory	11.25	37.78	27.59	6.87	27.76
No structured belief	7.09	33.91	21.99	2.66	41.44
Relaxed STOP verification	2.69	4.71	9.36	23.43	62.49
No belief + relaxed verification	7.58	23.93	73.30	0.78	1.99

Selective persistence. Removing instance or visual memory lowers SSC from 13.94% to 10.76% or 11.25%, respectively, and reduces IVC. More timeouts without instance memory and more abandonment without visual memory are consistent with their respective roles in retrieving grounded candidates and guiding continued search (Section 5). These ablations support reusable evidence at the system level; they do not isolate service-context conditioning.

Typed arbitration and continued pursuit. Without structured belief, SSC falls to 7.09% and abandonment rises from 20.94% to 41.44%, despite fewer invalid STOPs. This links the candidate-control configuration to maintaining pursuit as well as authorizing submissions. The switch removes the belief controller, direct gates, hypotheses, frontier utility, and structured target memory together, so it does not isolate candidate typing. Appendix E provides a narrower diagnostic of candidate roles.

Risk-aware submission and recovery. With structured belief retained, relaxed verification lowers SSC to 2.69% and IVC to 4.71%, with more timeout (23.43%) and abandonment (62.49%). Removing structured belief as well shifts the dominant outcome to invalid STOP (73.30%). Fewer invalid STOPs among all visits can therefore accompany poorer completion when fewer submissions occur. These contrasting outcomes motivate considering submission checks together with candidate revision and recovery, as in Section 5. The full configuration achieves the highest SSC among the tested variants.

7 Conclusion

We introduced SSLongNav, a benchmark that separately scores successive service-structured requests during continuous instance navigation. Its hierarchical evaluation distinguishes complete requests from partial progress while retaining outcomes after earlier failures. PEARS provides a reference framework combining selective persistence, typed evidence arbitration, and risk-aware submission. It improves strict completion over adapted baselines in both domains, while system ablations relate evidence reuse and candidate control to completion and terminal behavior. Results reveal substantial progress within incomplete requests and document complete multi-visit requests after earlier failures in the same run. Despite these gains, low strict completion rates in both domains highlight the challenge of SSLongNav and motivate further research on reliably completing successive navigation requests.

AI USE STATEMENT

We used OpenAI Codex to discuss manuscript organization and writing plans, and to polish the language for clarity and fluency. It also assisted in refining the wording and presentation of the experimental analysis, including descriptions of the reported results and the organization of the accompanying discussion.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018.

Luca Barsellotti, Roberto Bigazzi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Personalized instance-based navigation toward user-specific objects in realistic environments. In *Advances in Neural Information Processing Systems*, volume 37, pp. 11228–11250, 2024.

Rufeng Chen, Yue Chang, Zili Shao, Zhaofan Zhang, Li Chen, Hechang Chen, Hui Xiong, and Sihong Xie. LH-AVLN: A benchmark for long-horizon audio-visual-language navigation, 2026. arXiv preprint arXiv:2607.03920.

Won Shik Jang and Ue-Hwan Kim. Context-Nav: Context-driven exploration and viewpoint-aware 3D spatial reasoning for instance navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9626–9636, 2026.

Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. Habitat Synthetic Scenes Dataset (HSSD-200): An analysis of 3D scene scale and realism tradeoffs for ObjectGoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16384–16393, 2024a.

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, 2024b.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Proceedings of the European Conference on Computer Vision*, pp. 104–120, 2020.

Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances, 2022. arXiv preprint arXiv:2211.15876.

Jacob Krantz, Shurjo Banerjee, Wang Zhu, Jason Corso, Peter Anderson, Stefan Lee, and Jesse Thomason. Iterative vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14921–14930, 2023.

Xiaohan Lei, Min Wang, Wengang Zhou, Li Li, and Houqiang Li. Instance-aware exploration-verification-exploitation for instance ImageGoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16329–16339, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John 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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

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.

Francesco Taioli, Edoardo Zorzi, Gianni Franchi, Alberto Castellini, Alessandro Farinelli, Marco Cristani, and Yiming Wang. Collaborative instance object navigation: Leveraging uncertainty-awareness to minimize human-agent dialogues. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18781–18792, 2025.

Sen Wang, Bangwei Liu, Zhenkun Gao, Lizhuang Ma, Xuhong Wang, Yuan Xie, and Xin Tan. Explore with long-term memory: A benchmark and multimodal LLM-based reinforcement learning framework for embodied exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37098–37108, 2026a.

Xudong Wang, Jiahua Dong, Baichen Liu, Qi Lyu, Lianqing Liu, and Zhi Han. Lifelong embodied navigation learning. In *International Conference on Learning Representations*, 2026b.

Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. MultiON: Benchmarking semantic map memory using multi-object navigation. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9700–9712, 2020.

Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. UniGoal: Towards universal zero-shot goal-oriented navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19057–19066, 2025.

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.

Zhuofan Zhang, Ziyu Zhu, Junhao Li, Pengxiang Li, Tianxu Wang, Tengyu Liu, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Siyuan Huang, and Qing Li. Task-oriented sequential grounding and navigation in 3D scenes, 2024. arXiv preprint arXiv:2408.04034.

APPENDIX

A DETAILED TASK AND EVALUATION PROTOCOL

This section specifies the observation interface, episode representation, and evaluation-set aggregation for repeated strict subtask completion during continuous execution.

A.1 OBSERVATION AND CONTROL INTERFACE

The evaluator provides 640×480 RGB-D observations with a 120° horizontal field of view, a maximum depth of $5\,\mathrm{m}$, and simulator localization. The agent and sensor height are $1.25\,\mathrm{m}$. The navigation radius is $0.18\,\mathrm{m}$ for HSSD and $0.17\,\mathrm{m}$ for HM3D. The five standard actions are MOVEFORWARD, TURNLEFT, TURNRIGHT, STOP, and ABANDON. MOVEFORWARD advances $0.25\,\mathrm{m}$, and TURNLEFT and TURNRIGHT rotate by 30°.

Each requested visit has a budget of 500 actions. The episode-level cap is at least 60,000 actions, so a visit-level failure does not normally terminate the remaining stream. A successful STOP, invalid STOP, timeout, or ABANDON advances only the active visit. The Habitat task and simulator are initialized once per episode, and neither the simulator nor the agent pose is reset at a visit or subtask boundary.

Persistence refers to physical and information state within an episode. The protocol permits history-dependent policies and evaluates the navigation component of each request.

The public goal interface includes the active stable object ID, category, instance description, step intent, navigation hint, subtask title, service goal, and ordered subtask context. The stable ID is an identity token rather than a geometric pointer. Goal changes are observable transition events. Target positions, valid viewpoints, semantic masks, the scene-level target map, and evaluator success labels remain private to construction and scoring.

A.2 EPISODE REPRESENTATION

Both scene sources use the same task hierarchy. An episode $e = (\mathcal{S}, x_0, \mathcal{U})$ contains a scene $\mathcal{S}$, an initial pose x_0, and ordered subtasks $\mathcal{U} = (u_1, \ldots, u_M)$. Each $u_m = (c_m, r_{m,1}, \ldots, r_{m,L_m})$ groups $L_m \geq 1$ ordered instance visits under service context c_m specifying a shared goal. A visit $r_{m,l} = (i_{m,l}, q_{m,l})$ pairs a designated physical instance $i_{m,l}$ with public information $q_{m,l}$, including its instance description and visit intent. Subtask and visit orders are prescribed; an instance may be requested again in a later subtask. Stable instance IDs associate each visit with its description and private scoring geometry. Table 4 summarizes the information at each level and its visibility to the agent.

Table 4: Episode fields and their role. Stored target geometry is used only by construction and scoring, not by the navigation policy.

Level	Representative fields	Runtime visibility
Episode	episode identity, scene identity, start pose, ordered subtasks	ID and initial observations are public; start pose initializes the simulator
Subtask	title, service goal, ordered object IDs, ordered steps	Active subtask title, goal, and ordered context are public
Visit step	object ID, category, description, step intent, navigation hint	Public for the active instance visit
Ground truth	target position/rotation, valid viewpoints, semantic-mask and coverage metadata	Hidden from the policy; retained for validation and scoring

A.3 Hierarchical Metrics and Scoring Details

Visit success and strict completion. Let $\mathcal{E}$ denote the evaluation episodes and let $\mathbb{U}_{\mathcal{E}} = \biguplus_{e \in \mathcal{E}} \mathcal{U}_e$ collect their structured subtasks. Each subtask remains a separate evaluation unit even if it revisits an instance. Write $M = |\mathbb{U}_{\mathcal{E}}|$, let L_u be the number of prescribed visits in subtask u, and let $N = \sum_u L_u$ be the total number of visits. A visit indicator $S_{u,l}$ is one only when the agent submits STOP for the current designated instance at evaluation distance at most $1.0\,\mathrm{m}$; entering its success region alone is insufficient. Otherwise $S_{u,l} = 0$, including an invalid STOP, timeout, or abandonment. Strict completion is $C_u = \prod_{l=1}^{L_u} S_{u,l}$: every required visit must succeed in the prescribed execution order.

Metrics and aggregation. The four evaluation-set metrics are

$$\mathrm{SSC} = \frac{1}{M} \sum_{u \in \mathbb{U}_{\mathcal{E}}} C_u, \qquad \mathrm{SP} = \frac{1}{M} \sum_{u \in \mathbb{U}_{\mathcal{E}}} \frac{1}{L_u} \sum_{l=1}^{L_u} S_{u,l},$$

$$\mathrm{IVC} = \frac{1}{N} \sum_{u \in \mathbb{U}_{\mathcal{E}}} \sum_{l=1}^{L_u} S_{u,l}, \quad \mathrm{SS\text{-}SPL} = \frac{1}{M} \sum_{u \in \mathbb{U}_{\mathcal{E}}} C_u \frac{D_u}{\max(P_u, D_u)}.$$

SSC, the primary metric, measures the fraction of service-inspired requests whose navigation requirements are fully met. SP measures mean within-subtask progress, preserving successful visits in incomplete requests. IVC measures the fraction of all requested instance visits that succeed. SS-SPL weights strict completion by path efficiency, with D_u and P_u defined below. SSC, SP, and SS-SPL give every subtask equal weight; IVC gives every visit equal weight. Aggregation is directly over the evaluation set, not an equal-weight average of episode scores. All four quantities lie in $[0, 1]$; tables express them as percentages or, for SS-SPL, equivalently as $\times 100$.

Evaluation distance and path efficiency. The evaluator deduplicates the current target's valid viewpoints and stored goal position to form $\mathcal{G}_{u,l}$. Its distance is

$$d(x, \mathcal{G}_{u,l}) = \min_{p \in \mathcal{G}_{u,l}:\, d_{\mathrm{geo}}(x,p) < \infty} d_{\mathrm{geo}}(x, p).$$

Only when Habitat returns no finite geodesic for any member of the target set does the evaluator use the minimum Euclidean distance. For SS-SPL, $D_{u,l}$ is recomputed at the beginning of each visit from the actual pose left by the preceding outcome, and $P_{u,l}$ is the distance traveled during that visit. Then $D_u = \sum_l D_{u,l}$ and $P_u = \sum_l P_{u,l}$. The reference therefore sums conditional shortest distances along the realized continuous execution; it is neither a jointly optimized tour from the episode start nor a construction-time route summary. An incomplete subtask contributes zero to SS-SPL, as does a subtask with any unavailable or non-finite visit-start reference distance. Otherwise, for a completely successful subtask with $D_u \le 10^{-6}$, its efficiency ratio is set to one.

Reading the metrics together. Consider two subtasks with outcomes $(1, 1) \mid (1, 0, 0, 0)$, where 1 denotes success. SSC is $1/2 = 50\%$, SP is $(1 + 1/4)/2 = 62.5\%$, and IVC is $3/6 = 50\%$. The four-visit subtask has 25% progress and is incomplete; its successful visit contributes to SP and IVC, but not SSC. SP gives the two subtasks equal weight, whereas the longer subtask accounts for four of IVC's six visits. More generally, SP minus SSC is the progress contributed by incomplete subtasks averaged over all subtasks; subtracting SSC from IVC instead mixes statistical units. For single-visit subtasks, progress and strict completion coincide. A case's complete/incomplete label reports C_u, while SSC aggregates these indicators over the evaluation set.

B Benchmark Construction Details

Construction turns scene instances into a workload with explicit completion units. Navigable instances provide grounded targets; local grouping and a fixed order define which visits belong together; shared service text links those visits; and concatenation preserves each group's boundary within one continuous episode. The statistics below account for this structure, followed by the construction rules and annotation interface.

B.1 BENCHMARK STATISTICS

Table 5 gives the source- and split-wise accounting underlying the aggregate scale reported in the main paper. "Unique" counts distinct target instances within each episode and is then summed across episodes; "Visits" counts the ordered target stream and therefore includes cross-subtask revisits. "Sub./Ep." and "Visits/Ep." are split-level per-episode means.

Category coverage is computed as the union of category labels in the retained ordered visits across training and evaluation: 28 for HSSD and 350 for HM3D. HM3D retains the category labels supplied by the source resources, counting differently named labels separately even when they are synonymous. HM3D contains 261 category labels in training and 153 in evaluation; the union counts labels shared by the two splits once.

Table 5: SSLongNav statistics by scene family and split.

Split	Scenes	Ep.	Subtasks	Unique	Visits	Sub./Ep.	Visits/Ep.
HSSD train	117	120	1,153	3,534	3,664	9.61	30.53
HSSD eval	43	45	452	1,345	1,398	10.04	31.07
HM3D train	131	215	1,655	5,732	5,888	7.70	27.39
HM3D eval	35	70	409	1,784	1,805	5.84	25.79
Train total	248	335	2,808	9,266	9,552	8.38	28.51
Eval total	78	115	861	3,129	3,203	7.49	27.85
All total	326	450	3,669	12,395	12,755	8.15	28.34

B.2 EVALUATION STRUCTURE AND COUNTING CONVENTIONS

Figure 3 is computed directly from the evaluation episode annotations, without filtering by a policy's outcomes. HSSD contributes 45 episodes and 452 subtasks; HM3D contributes 70 episodes and 409 subtasks. Episodes are identified by the pair of scene ID and episode ID, so repeated episode IDs in different scenes are not merged. Every ordered step counts as one instance visit, including a visit to an instance requested in an earlier subtask. The ordered step IDs are checked against each subtask's stored instance sequence and the scene's target catalog before aggregation.

Table 6: Integer counts underlying the structural distributions in Figure 3. The upper block counts episodes; the lower block counts subtasks. Zero denotes an absent length in the corresponding evaluation split.

Subtasks/episode	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HSSD	0	0	0	0	0	3	3	7	10	4	5	3	6	2	2
HM3D	1	3	3	10	17	9	12	8	4	3	0	0	0	0	0

Visits/subtask	1	2	3	4	5	6	7	8
HSSD	33	171	100	60	45	43	0	0
HM3D	19	97	59	51	44	35	51	53

The subtasks-per-episode histogram is normalized by the number of episodes in each source; the visits-per-subtask histogram is normalized by the number of subtasks. In particular, the evaluation split has 809 multi-visit and 52 singleton subtasks, whereas the 94.8% multi-visit figure in the main scale description refers to all splits (3,477 of 3,669). The evaluation means are $452/45 \approx 10.04$ and $409/70 \approx 5.84$ subtasks per episode, and $1,398/452 \approx 3.09$ and $1,805/409 \approx 4.41$ visits per subtask, after rounding.

An episode has *same-category distinct instances* if at least two requested object IDs share a category; this holds in 45/45 HSSD and 67/70 HM3D episodes. A *cross-subtask revisit* requires the same object ID to occur in two different subtasks, and holds in 25/45 and 15/70 episodes. These are episode-level indicators, not fractions of visits, and do not imply disjoint success regions (Appendix B.8). Additional workload summaries are 45/45 and 69/70 episodes with at least two subtasks, 45/45 and 56/70 with at least 20 visits, and 44/45 and 42/70 with at least 30 visits. These counts use the episode and subtask denominators specified above.

B.3 SOURCE-NATIVE SCALE AND TASK ORGANIZATION

Table 7 situates the repeated completion level among existing task organizations. It retains each source's own unit—episode, tour, task, trajectory, or step—and reports the navigation subset of SG3D. The comparison identifies the completion units and execution links supported by each task organization.

Table 7: Native reported scales and task organizations of representative long-horizon embodied navigation benchmarks.

Benchmark	Scenes	Native reported scale	Evaluation organization
MultiON (Wani et al., 2020)	90	50k training episodes/scene per m-ON setting	Episode → 1–3 ordered goals
IVLN (Krantz et al., 2023)	80	375 IR2R tours 414 IR2R-CE tours	Tour → up to 100 one-goal episodes Oracle repositioning between episodes
GOAT-Bench (Khanna et al., 2024b)	181	725k training episodes 360 validation episodes	Episode → 5–10 goals Continues after success or timeout
LHPR-VLN (Song et al., 2025)	216	3,260 tasks	Task → 2–4 stages
LMEE (Wang et al., 2026a)	181	1,982 task trajectories 9,708 navigation goals	Trajectory → navigation goals → memory QA
LENL (Wang et al., 2026b)	18	18 lifelong tasks 3 instruction styles	Learning task → separate episodes Only learned knowledge persists
SG3D-Nav (Zhang et al., 2024)	181	2,868 navigation episodes 12,385 object steps	Episode → grounded object steps
SSLongNav	**326**	**450 episodes; 3,669 subtasks 12,755 visits**	**Episode → strict subtasks → grounded-instance visits**

An IVLN tour connects one-goal episodes, while an SG3D-Nav episode contains the object steps of one task. SSLongNav introduces a repeated request-completion level inside the episode: each structured subtask can be completed independently of whether earlier requests succeeded.

Comparison scope and target specifications. In Table 1, "Visits required for joint success" specifies the distinct goal-visiting steps that must all succeed together. Intermediate waypoints along a single instruction-following route do not constitute separate goal visits. "Repeated multi-visit groups per run" means that multiple such groups are separately scored within the same run; it does not count a sequence of individual single-goal tasks or an average over independently initialized task episodes. SSLongNav permits singleton subtasks as well, but 94.8% of its subtasks contain multiple visits. MultiON specifies category goals, IVLN supplies route instructions, and GOAT-Bench mixes category, language, and image goals. LHPR-VLN uses language-grounded object targets; SG3D-Nav and SSLongNav associate each visiting step with a designated object instance. These target specifications complement the completion-structure comparison.

Completion and transition rules. MultiON measures goal progress and whole-sequence success; an incorrect FOUND action or exhausted episode budget terminates the episode (Wani et al., 2020). IVLN instead connects navigation episodes with oracle phases: after the agent stops or exhausts its budget, the oracle completes the route when needed and leads it to the next episode's start, while observations remain available (Krantz et al., 2023). GOAT-Bench calls each individual goal a subtask and advances to the next goal after success or timeout without resetting the scene (Khanna et al., 2024b). Its category goals accept category-level specifications, whereas language and image goals identify instances. Each is scored as a single goal, so the native term "subtask" does not denote the multi-visit completion level compared in Table 1.

LHPR-VLN provides stage-wise ISR and conditional task scores, and its standard task-level SR requires all navigation stages to succeed in order (Song et al., 2025). SG3D-Nav likewise reports step success and task success, with the latter requiring every object step in the task episode to succeed (Zhang et al., 2024). SSLongNav applies strict completion repeatedly to separate requests

in a shared run. The cited LHPR-VLN and SG3D-Nav task descriptions do not explicitly specify the after-failure pose and advancement rule; Table 1 therefore marks this rule as "Not specified" rather than treating an unspecified rule as an absent capability. LMEE adds memory question answering to navigation trajectories, while LENL studies continued learning across task distributions; both remain in the broader comparison because they address complementary meanings of long-horizon experience (Wang et al., 2026a;b).

Evaluation consequence. Together, the completion and transition rules give each structured subtask a separate strict outcome while preserving the execution history shared by successive requests. SSC measures complete requests, and SP and IVC retain partial progress at their respective aggregation levels (Appendix A.3).

B.4 HSSD Instance and Viewpoint Construction

The HSSD benchmark uses a 28-category target vocabulary selected from HSSD's broader scene semantics; 28 is the benchmark target-vocabulary size, not the size of the full HSSD ontology. The HSSD builder reconstructs the navigation mesh with a $1.25\,\mathrm{m}$-high, $0.18\,\mathrm{m}$-radius agent and includes static objects. For each semantic instance, it samples 24 azimuths at horizontal radii 0.8, 1.2, 1.8, and $2.5\,\mathrm{m}$. Candidate positions are snapped to the navigation mesh, rejected when their vertical offset exceeds $1.5\,\mathrm{m}$ or their horizontal distance is below $0.35\,\mathrm{m}$, and retained only when a finite geodesic path exists. At most 24 geometric candidates per instance are sent to rendering.

Exact-instance visibility is then checked with a 640×480, 120°-HFOV semantic camera at $1.25\,\mathrm{m}$. The builder temporarily assigns a distinct semantic ID to every candidate object and retains a viewpoint only if the intended instance occupies at least 50 semantic-mask pixels. Viewpoints are ranked using visibility and geodesic distance, and at most five are stored per instance.

Floors are clustered from valid viewpoint heights with a $1.0\,\mathrm{m}$ threshold, and a floor must contain at least 15 valid instances. Only the navigation island containing the largest number of retained targets is used. Category-aware sampling first favors rare categories and then cycles across categories with a four-instance round-robin quota before filling any remaining slots from the complete valid pool. The quota is therefore a diversity preference rather than a hard per-category cap. At most 30 distinct instances are selected. The builder permits up to eight target-selection retries and samples a start pose that has a finite path to every selected target. Each valid floor produces one episode; hence a multi-floor scene can produce multiple episodes. Companion repair occurs later and can make the final ordered-visit count exceed the 30-instance limit.

Table 8 separates the effects of geometric and visual validation. Geometry removes only 84 of 16,481 raw training instances and 24 of 7,165 evaluation instances, whereas exact-instance visibility is the more selective stage, retaining 14,469 and 6,262 instances. Episode-level sampling then selects 3,534 training and 1,345 evaluation instances, from which multiple stored viewpoints remain available for scoring and description. These counts document construction attrition rather than agent performance.

Table 8: HSSD construction attrition. The first four numeric columns count instances; only the last counts stored viewpoints. Selected instances are episode-level distinct targets before companion revisits.

Split	Raw instances	Instances with geometric viewpoints	Visible instances	Selected instances	Stored viewpoints
Train	16,481	16,397	14,469	3,534	17,516
Eval	7,165	7,141	6,262	1,345	6,662

B.5 HM3D Filtering and Sequential Rollout

HM3D construction first aligns scene keys and integrates target catalogs from HM3D-OVON and GOAT-style resources. When both resources supply the same category entry, the OVON-side entry takes precedence. Only instances with a resolvable object ID, nonempty category and description, position, and candidate viewpoint proceed to episode construction. Evaluation construction maintains a primary goal stream and a supplementary seen-goal stream through floor selection and reachability filtering, then combines their filtered episodes before forming subtasks.

The initial builder uses the first candidate viewpoint as the representative target position and clusters floors with a $1.5\,\mathrm{m}$ height threshold. A floor must contain at least two targets. A same-floor pass removes goals whose representative viewpoint does not match the episode-start floor. The first reachability filter deduplicates object IDs, snaps the start and target viewpoints to the navigation mesh, requires them to share a connected island and a finite shortest path, and retains an episode only when at least 95% of its initial targets satisfy this test.

A second filter performs a construction-time sequential discrete rollout. The validator begins at the episode start, advances its state to the reached pose after a successful target, and restores the pre-attempt state after removing a failed target. It first follows a path to the direct viewpoint; if that fails, it searches reachable points within $2.5\,\mathrm{m}$ using $0.35\,\mathrm{m}$ radial increments and 16 angles per ring. The rollout uses $0.25\,\mathrm{m}$ forward steps, 15° turns, a $0.22\,\mathrm{m}$ waypoint threshold, and at most 80 actions per waypoint. Its camera is 640×480 with 120° HFOV at $1.25\,\mathrm{m}$. Because formal evaluation turns by 30°, this rollout is an executability filter, not an exact replay of the evaluated controller.

After sequential filtering, the two evaluation streams are consolidated scene by scene. Episodes whose start heights differ by less than $0.1\,\mathrm{m}$ are paired; their target lists are concatenated in primary-then-supplementary order, retaining the primary episode's start pose. Unmatched episodes from either stream are retained. The evaluation initial and filtering rows in Table 9 trace the primary stream, and the merged-episode row reports the combined inputs to subtask formation. Spatial grouping and conditional singleton repair then produce the ordered subtasks described in Appendix B.6.

For HM3D evaluation only, a fixed, language-independent selection step processes streams longer than 35 visits. It samples whole subtasks to approach 30 visits without splitting a subtask, producing 2–35 visits per final episode with a median of 30. This selection uses a threshold of 35 visits, a target length of 30, and sampling seed 20260527. Training retains all subtasks after formation. Companion revisits introduced by singleton repair account for the increase from target counts to ordered visits.

Table 9: HM3D stage counts. A dash indicates that subtasks have not yet been formed.

Stage	Episodes	Visits	Subtasks
Eval single-floor initial	76	1,536	–
Eval connectivity filter	65	1,342	–
Eval sequential rollout	65	1,333	–
Eval merged filtered episodes	70	2,614	–
Eval subtasks before sampling	70	2,650	629
Eval final	70	1,805	409
Train single-floor initial	268	7,243	–
Train connectivity filter	215	5,857	–
Train sequential rollout	215	5,732	–
Train final subtasks	215	5,888	1,655

B.6 Spatial Subtask Formation

Targets are deduplicated by object ID and processed in input task order using representative viewpoints. A target joins the nearest-centroid cluster for which both centroid and nearest-member Euclidean distances are at most $d_c = 2.5\,\mathrm{m}$; otherwise it starts a new cluster. Each singleton is merged into the nearest multi-instance cluster if its centroid distance is at most $1.25d_c$ and that cluster has fewer than eight members. Remaining singletons are paired within $1.2d_c$. If no multi-instance cluster exists, this merge-and-pair stage retains the initial clusters.

Clusters are ordered from the episode start by nearest centroid, updating the group-ordering anchor to each selected centroid. Within a cluster, nearest-neighbor ordering starts from the preceding emitted visit's endpoint, or the episode start for the first cluster. Ordered clusters are split into groups of at most six visits in HSSD or eight in HM3D, shortening a maximal chunk by one to avoid a singleton remainder. A singleton receives its nearest different instance from the episode target pool within $1.4d_c = 3.5\,\mathrm{m}$, if available; the pair is reordered from the current visit anchor. Otherwise it remains single. The visit anchor advances to the last instance of each emitted group.

Within an episode, identical ordered object-ID sequences are emitted only once. Companion insertion can reuse an instance from another subtask, creating cross-subtask revisits. The final dataset

contains 3,477 multi-visit and 192 singleton subtasks; companion visits account for the difference between episode-level unique instances and ordered visits. Shared service text is assigned to these fixed groups and sequences below.

B.7 Multi-View Validation and Service-Context Annotation

For HSSD instance descriptions, the pipeline selects up to three complementary views that balance visibility, maximal visible area, and surrounding context. RGB, bounding-box, mask-overlay, and crop variants are rendered, and the selected views are jointly supplied to local Qwen3.5-9B to produce a concise description that distinguishes the instance from same-category alternatives. All 4,879 selected HSSD instances have nonempty descriptions: 4,878 use the generated grounded brief and one was manually cleaned.

For HM3D instance descriptions, we render target images from candidate viewpoints and provide each image and its target category to Qwen3.5-9B to describe appearance and nearby context. We match the generated descriptions by scene and object ID and write them into the instance catalogs before episode construction.

HM3D multi-view validation follows target filtering and final subtask selection. For every retained physical instance, the renderer selects up to four reachable candidate viewpoints, preferring camera positions separated by at least $0.65\,\mathrm{m}$ and 30° around the target. Review panels pair these spatially distinct RGB views with the stable object ID, category, assigned description, target position, and available viewpoint diagnostics. The panel is accepted only when the views consistently identify the same reachable instance on the selected floor and the description remains valid across viewpoints. This validation does not expose privileged views to the evaluated policy.

Service semanticization begins after target membership, IDs, geometry, and both ordering levels have been fixed. The annotation model receives each ordered instance group with object IDs, categories, task categories, and descriptions capped at 50 words. It produces a subtask title, a shared service goal, and one intent per ordered visit. Table 10 specifies the annotation requirements and alignment checks. Text generation changes neither IDs nor geometry nor the visit sequence. All 3,477 multi-visit subtasks have Qwen-generated service text; the 192 singleton subtasks use deterministic templates. Length-one analyses therefore also differ in text source.

Table 10: Service-context annotation requirements. Quoted phrases are prompt excerpts; the final row specifies validation and alignment.

Component	Input, instruction, or validation
System prompt	"You are an expert home-service instruction writer."
User instruction	"Rewrite one cluster subtask into natural human household instruction." The goal must be a concise English imperative, omit raw instance IDs, and "Keep the same ordered instance chain semantics."
Annotation output	A subtask title, a shared service goal, and an ordered list of visit intents, with one intent per input instance.
Validation and alignment	Validate the title, service goal, and visit intents; cap title, goal, and intents at 18, 42, and 14 words. If valid intent count differs from the chain length, construct aligned intents from the cluster theme. Align intents with the prescribed visits and derive each navigation hint from its designated instance ID and category.

For a concrete HSSD example, the benchmark contains the subtask *Clear desk and chair area*, with service goal "Please move the potted plant from the desk and adjust the chair under the desk." Its fixed visits are `potted_plant_96` followed by `chair_11`. Their descriptions specify a plant beside a computer monitor on the desk and a black-frame, wooden-seat chair under that desk. The shared context connects the two visits, while evaluation scores only navigation to those instances; moving the plant and adjusting the chair are not executed or evaluated. This illustrates annotation of an existing ordered group, rather than generation of a manipulation plan from a user request.

For every selected HSSD and HM3D target, human reviewers inspect the multi-view package together with its object ID, category, description, target geometry, and subtask assignment. They reject inconsistent identity or category labels, views that do not support the assigned instance, floor or reachability mismatches, and descriptions that are inaccurate or insufficiently discriminative. Reviewers then check each generated subtask title, service goal, and step intent against the fixed or-

dered target sequence. Incorrect text is corrected or regenerated and passes a second manual check before finalization. The review checks that the shared goal connects the group, that each intent refers coherently to its assigned instance, and that the language preserves the fixed navigation requirements.

After semanticization and manual review, programmatic validation checks that every ordered object ID resolves to the episode catalog, every visit has the evaluator-required identity and geometry fields, and every selected unique instance appears in at least one subtask. The final benchmark has 100% coverage of the selected instances. Subtasks are then concatenated in their prescribed order with a single start pose, retaining their group boundaries for scoring. Geometry, semantic masks, generation warnings, and validation metadata are retained for construction and evaluation and excluded from the policy interface.

B.8 Grounded-Instance Separability Audit

Because success is defined against a 1-m region around a target's designated viewpoint set and stored goal position, distinct IDs do not automatically imply disjoint scoring regions. We audit every same-episode, same-category, different-ID target pair in the scene-disjoint evaluation split. A pair is flagged as a *potential overlap* when the minimum finite Habitat geodesic between its two scoring-anchor sets is at most $2 \times 1.0\,\mathrm{m}$. Euclidean distance is used only for sound non-overlap pruning when it already exceeds this threshold; if no finite geodesic exists, the pair remains unresolved rather than falling back to Euclidean distance.

Table 11: Grounded-instance separability audit. "Same-category-distinct-ID (SCDI) visits" request an instance whose category includes another requested ID in the episode; "overlap visits" counts those involving at least one flagged pair. Pair percentages use all audited pairs; visit percentages use SCDI visits as the denominator.

Domain	Pairs	Overlap pairs	Rate	SCDI visits	Overlap visits	Unresolved
HSSD	1,137	345	30.34%	1,173	447 (38.11%)	0
HM3D	856	444	51.87%	1,032	605 (58.62%)	1

Table 11 flags 30.34% of HSSD pairs and 51.87% of HM3D pairs as potential overlaps; the corresponding visit rates are 38.11% and 58.62%. A flagged pair identifies possible shared acceptance locations, rather than an observed submission in the intersection; one HM3D pair with no finite path remains unresolved. This characterization connects physical instance identity to the actual scoring geometry. Requests specify stable IDs and descriptions, while the distance criterion can accept overlapping regions for nearby instances. Accordingly, the same-category analyses measure performance under distinct-instance requests, with the geometric overlap rate available for interpreting those results.

C PEARS Execution Details

PEARS uses shared subtask context to guide evidence acquisition and queries persistent stores to reuse earlier observations. Typed candidates distinguish evidence supporting the active target from locations that guide further search. The following interface, state lifecycle, and submission rules expand the three mechanisms in the main paper.

C.1 Public Inputs and Execution Cycle

PEARS follows the benchmark's prescribed instance order. Shared service context conditions evidence acquisition and verification for the currently prescribed visit. Table 12 distinguishes the available fields from their roles in retrieval and verification. Related-object cues are extracted from the current subtask's other instance categories, excluding categories equivalent to the active target and retaining at most 12 deduplicated terms. These cues guide contextual search for the prescribed target through the public interface in Appendix A. Each requested visit has a distinct identity for target-local bookkeeping, so a later request for the same physical instance starts a new attempt.

Figure 6 shows how observation updates, arbitration, and submission feedback connect. Each new observation updates scene evidence and target hypotheses. Navigation and scanning acquire further

Table 12: Public inputs and their roles in PEARS. Different retrieval and verification stages use different subsets of these fields.

Input	Use in the agent
RGB-D and pose	Update geometric and visual evidence; localize observations and execute navigation.
Active object ID	Identify the requested physical instance; separate visit keys track attempts. IDs supply no coordinates or visually observable labels.
Category and instance description	Specify the active target, construct target queries, and support instance discrimination.
Service goal, subtask title, step intent, navigation hint	Construct contextual queries and full-goal verification text.
Ordered subtask targets	Supply related-object categories for contextual search; the benchmark determines visit order.

evidence; rejected STOP proposals revise evidence and trigger recovery. Table 13 gives the execution sequence. Target-local state is initialized at visit transitions, while pose and reusable evidence carry forward within and across subtasks.

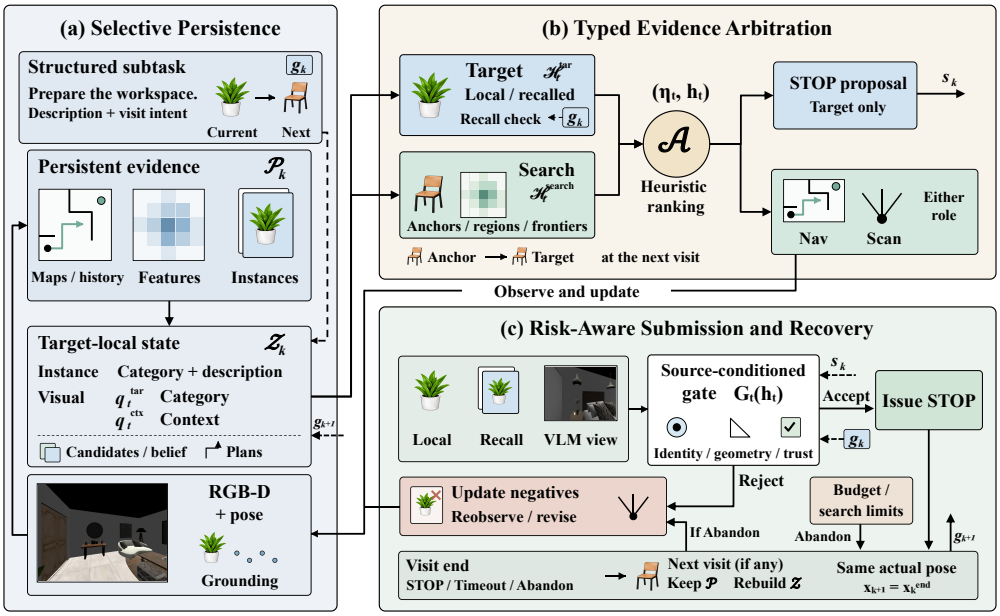


Figure 6: Detailed PEARS execution. Public target and subtask information condition distinct retrieval and verification paths. Typed arbitration $\mathcal{A}$ selects (η_t, h_t), and the source-conditioned gate $G_t(h_t)$ authorizes STOP (Eq. 3). Feedback connects submission to recovery and visit transitions; Table 13 gives their execution order.

C.2 Selective Persistence and Context-Guided Retrieval

State lifecycle. Episode initialization clears all agent state. At visit boundaries, reusable scene evidence and execution history form persistent state $\mathcal{P}_k$, while target-conditioned queries and decisions form local state $\mathcal{Z}_k$, as specified in Table 14. Let k index the flattened visit sequence and g_k denote its active public goal payload, including the instance specification and subtask context. Let x_k be the starting pose of visit k. For a visit with a successor, after trace τ_k and agent-side terminal record $\hat{y}_k$, the transition is

$$\mathcal{P}_{k+1} = \mathrm{Update}(\mathcal{P}_k, \tau_k, \hat{y}_k), \quad \mathcal{Z}_{k+1} = \mathrm{Init}(\mathcal{P}_{k+1}, g_{k+1}), \quad x_{k+1} = x_k^{\mathrm{end}}. \tag{4}$$

Here, $\hat{y}_k$ records observable goal-switch events, the agent's own terminal action, and internal diagnosis under the interface in Appendix A. The pose equality implements continuous execution. At boundaries within and across subtasks, retained scene evidence is queried under the new request, while local candidates and plans are rebuilt around its instance specification.

Table 13: PEARS execution cycle under the information interface in Appendix A.

1	Initialize persistent evidence $\mathcal{P}$ once at episode start; initialize local state $\mathcal{Z}$ for the first prescribed visit.
2	Observe RGB-D and pose; update scene evidence and target-local hypotheses. Query visual and instance stores using their respective target/context inputs.
3	Following the applicable controller priorities, select navigation, scanning, a target STOP proposal, or ABANDON under the branch-specific budget or search-exhaustion conditions in Appendix C.4.
4	Execute navigation/scanning or authorized ABANDON directly. For a STOP proposal, apply its source-conditioned gate: issue STOP if authorized, or update negative evidence and invoke branch-specific recovery.
5	After a nonterminal action, return to step 2. After issued STOP, timeout, or qualified abandonment, record the agent-side trace and transition event.
6	If a prescribed visit remains, retain $\mathcal{P}$ and the realized pose, initialize its $\mathcal{Z}$ and queries, and return to step 2. Otherwise finish the episode.

Table 14: State lifecycle at both visit and subtask boundaries.

Treatment	State
Retain/update	Obstacle/exploration maps; visual, object, and instance evidence; trajectory, terminal-action and reliability histories; cross-target negative evidence.
Rebuild	Active-goal candidates and belief nodes; probes, local plans, target-conditioned frontier/value state; target/context queries and their scores.
Continue	Simulator state and actual robot pose; prescribed next visit, within the same subtask or the next one.

Visual memory and its two queries. The visual memory projects normalized vision tokens from the SigLIP2 vision encoder through RGB-D into a sparse episodic top-down grid. Each cell r retains at most four prototypes $(p_{r,j}, n_{r,j}, \tau_{r,j})$: a normalized feature, support count, and last-seen step. If a new feature f has cosine similarity at least 0.90 with the closest prototype, the update is

$$p_{r,j} \leftarrow \mathrm{norm}\left(\frac{n_{r,j}p_{r,j} + f}{n_{r,j} + 1}\right).$$

Otherwise a prototype is added, or the least-supported, oldest entry in a full cell is replaced. Stored features are target-agnostic, and their cosine similarities with text features provide heuristic retrieval scores.

For long-term navigation, the visual target query q_t^{tar} uses the active category. The context query q_t^{ctx} combines the instance description, step intent, navigation hint, service goal, and subtask title. With both query features available, the reference configuration uses

$$S_t^{\mathrm{vis}}(r) = 0.8 \max_j \langle \phi(q_t^{\mathrm{tar}}), p_{r,j} \rangle + 0.2 \max_j \langle \phi(q_t^{\mathrm{ctx}}), p_{r,j} \rangle.$$

Here, t indexes the observation step and ϕ denotes the normalized text encoder. If only one query feature is available, its weight is renormalized to one; if neither is available, the query scores are cleared. Blacklisted cells and overly close proposals are suppressed before a cell or frontier is selected as a search waypoint. A new visit changes the queries and query-dependent scores while retaining the visual features, allowing prior observations to be searched under the new target and service context.

Instance retrieval and full-context verification. Grounded instance memory stores promoted centers, category identities and aliases, representative views, support/rejection histories, per-goal attempts, reliability, and final-STOP trust. The target retrieval query uses category and instance description. A separate full-goal text combines the description, step intent, navigation hint, service goal, and subtask title, including the target category when needed. This full-goal text supports verification of retrieved views and the contextual retrieval branch, complementing the category-and-description query used for instance retrieval.

Retrieval expands category aliases, excludes records already attempted for the same goal, and ranks candidates by image-text similarity with recency, support, and reliability adjustments. The policy first considers target-category records and checks selected stored views against the full-goal text.

Rejected or stale records are marked; uncertain target evidence can produce a re-observation probe. When no usable target probe is selected, context-category records are queried and checked as search anchors. Context categories come from the description, service fields, and related subtask objects. In either branch, a recalled record supplies a navigation probe at its observed location. Submission uses the source-conditioned checks in Appendix C.4.

C.3 Typed Evidence Arbitration

Online perceptual and geometric evidence. YOLO26x detects COCO-mappable target and context categories at confidence 0.25; GroundingDINO Swin-T supplies open-vocabulary and weak-YOLO evidence. Detections overlapping at IoU ≥ 0.5 are merged with YOLO preferred, and MobileSAM masks the retained boxes. Valid depths are projected and associated across views into object point clouds; detections without usable in-range depth are not promoted. Promotion, current-goal support, rejection, and verification history determine whether mapped evidence is navigable or trusted for submission.

Candidate roles and action authority. The unified belief contains typed nodes for promoted target instances, local detections, memory probes, visual probes, context anchors, frontiers, and recovery probes. At decision step t, target hypotheses $\mathcal{H}_t^{\mathrm{tar}}$ concern the active designated instance; search hypotheses $\mathcal{H}_t^{\mathrm{search}}$ identify useful places to gather evidence. A recalled object is assigned according to its role: an active-target probe and a contextual anchor do not have the same STOP authority. Visual regions and other exploratory proposals require grounded target support before they can contribute to a STOP proposal. Equation 3 summarizes the ordinary navigation cycle: $\eta_t \in \{\mathrm{Nav}, \mathrm{Scan}, \mathrm{StopProp}\}$, and $\eta_t = \mathrm{StopProp}$ requires $h_t \in \mathcal{H}_t^{\mathrm{tar}}$. Executing STOP additionally requires the submission gate defined below. The two hypothesis groups are roles within one typed belief representation. Qualified abandonment follows the branch-specific conditions below.

Each node records target relevance, identity support, predicted visibility, STOP support, information gain, path cost, false-STOP risk, loop risk, source, lifecycle, and verification state. These are deterministic heuristic scores, not calibrated posterior probabilities. For node i, the implementation computes

$$\begin{aligned} U_i^{\mathrm{nav}} &= 2.0p_i^{\mathrm{tar}} + 1.3p_i^{\mathrm{id}} + 0.8p_i^{\mathrm{vis}} + 0.7I_i - 0.5C_i - 1.2R_i^{\mathrm{stop}} - 0.7R_i^{\mathrm{loop}}, \\ U_i^{\mathrm{stop}} &= 4.0p_i^{\mathrm{stop}} - 5.0R_i^{\mathrm{stop}} - 0.6R_i^{\mathrm{loop}}. \end{aligned} \tag{5}$$

Here, $p_i^{\mathrm{tar}}, p_i^{\mathrm{id}}, p_i^{\mathrm{vis}}, p_i^{\mathrm{stop}}$ denote target, identity, visibility, and STOP support; $I_i, C_i, R_i^{\mathrm{stop}}, R_i^{\mathrm{loop}}$ denote information gain, path cost, STOP risk, and loop risk. The reference configuration keeps at most 96 nodes per goal, uses a navigation-candidate utility threshold of 0.25, and applies a region-navigation threshold of 0.20. The controller uses a STOP-support threshold of 0.70 for proposals; the belief precheck at submission uses $p_i^{\mathrm{stop}} \geq 0.72$ and a 0.20 margin between STOP and continuation utilities.

A nearby target-like node with sufficient STOP evidence proposes STOP; a nearby but uncertain node proposes scanning after 20 visit actions. If neither branch selects an action, the controller evaluates coverage-budget conditions before prioritizing target navigation, region navigation, and exploration (Appendix C.4). Scanning uses directed turns and re-observation; selected navigation goals are passed to the controller below.

Low-level navigation. The controller plans on the occupancy grid built from the agent's observations. If the start or goal cell is blocked, it selects the nearest free cell in a square search window with a $1.5\,\mathrm{m}$ half-width along each grid axis. It then uses eight-neighbor A* with Euclidean distance as the heuristic and disallows diagonal moves through blocked corners. Planning first searches a bounded region around the endpoints and expands to the full observed grid if needed. A waypoint about $1.0\,\mathrm{m}$ ahead along the path guides the next action. The controller turns toward it when heading error exceeds 25° and checks map and depth clearance before moving forward. Blocked motion and turn oscillation invoke recovery. Forward and turn increments follow Appendix A; target submission follows Appendix C.4.

Heuristic score construction. All support, gain, cost, and risk entries are clipped to $[0, 1]$ when a candidate is inserted or updated. For the instance-hypothesis branch, let b be the hypothesis belief,

v_n, v_s its navigation/STOP verification flags, n its current-goal sightings, s its support count, r its rejection count, and n_- its explicit current-view negatives. With $[z]_0^1 = \min(1, \max(0, z))$ and $[z]_+ = \max(0, z)$, the instance-hypothesis update computes

$$\begin{aligned} p^{\text{tar}} &= \sigma(1.2(b - 0.25)), \\ p^{\text{id}} &= [0.25 + 0.25v_n + 0.30v_s + 0.10\min(3, n) + 0.04\min(4, s)]_0^1, \\ R^{\text{stop}} &= [0.10 + 0.22r + 0.40n_-]_0^1, \\ p^{\text{stop}} &= [0.35p^{\text{tar}} + 0.45p^{\text{id}} + 0.20v_s - R^{\text{stop}}]_0^1. \end{aligned}$$

Here σ is the logistic function. The same branch sets $p^{\text{vis}} = [0.35 + 0.20v_n + 0.15\mathbf{1}[n > 0]]_0^1$, $I = [0.10 + 0.05\max(0, 3 - s)]_0^1$, $C = [\|x_i - x_t\|_2/8]_0^1$, and $R^{\text{loop}} = [0.10r]_0^1$. The distance is a policy-side planar cost proxy, distinct from the evaluator's geodesic reference distance.

For frontier candidates, with base frontier value v and best positive anchor bonus $a = \max_j U_j^{\text{nav}}[1 - d_{ij}/4]_+$ over up to 16 eligible anchors within 4 m, their fields are $p^{\text{tar}} = [0.20 + 0.20a]_0^1$, $p^{\text{id}} = 0.05$, $p^{\text{vis}} = [0.35 + 0.25a]_0^1$, $I = [v + 0.20a]_0^1$, and $C = [\|x_i - x_t\|_2/10]_0^1$; STOP support and both risk fields are zero. Frontier ranking uses the larger of its navigation utility and $v + 0.35a - 0.30C$. Probe candidates use the same representation; their authority to submit STOP depends on the source-specific evidence checks in Table 15.

C.4 Risk-Aware Submission and Recovery

For a STOP proposal at decision step t, let i_t index the selected hypothesis h_t. The main-text gate is $G_t(h_t) = G_{i_t}^{\text{stop}}$; its source-conditioned checks and rejection branch are

$$\begin{aligned} G_i^{\text{stop}} &= \mathbf{1}[\text{active}(i) \wedge \text{object}(i)] V_i^{\text{src}} V_i^{\text{sem}} V_i^{\text{geo}} \prod_{b \in \mathcal{B}_i} (1 - b), \\ \text{IssueStop}_t &\iff G_{i_t}^{\text{stop}} = 1, \\ G_{i_t}^{\text{stop}} = 0 &\implies a_t \in \mathcal{A}_t^{\text{rec}}. \end{aligned} \tag{6}$$

Here, $V_i^{\text{src}}, V_i^{\text{sem}}, V_i^{\text{geo}}$ summarize applicable source trust, semantic evidence, and geometric support; $\mathcal{B}_i$ groups lifecycle, stale, early-STOP, same-category, previous-rejection, and previous-STOP vetoes. The V terms and veto indicators b are binary. Active-goal and navigation-mode consistency are checked by the final source-specific arbiter. The equation summarizes the source-conditioned checks along each proposal's submission branch.

Semantic support combines the evidence available to the active branch, including Qwen decisions, current-view observations, and rule-based instance-belief overrides. A local candidate requires current evidence and sufficient promotion support. A memory candidate requires a trusted record and its applicable stored/full-goal evidence. A VLM-visible proposal requires current-view and geometric support. Soft aliases and provisional records support navigation, with submission governed by the checks in Table 15.

Table 15: Source-conditioned STOP support. All branches additionally enforce the active target, mode, lifecycle, and applicable veto checks. Search-only anchors support evidence gathering; only active-target hypotheses can propose STOP.

Proposal source	Evidence checked along its submission branch
Local target	Current target evidence, promotion/support and local geometry; domain-specific current-verification and early-stop rules.
Recalled instance	Record trust and applicable stored/full-goal evidence, followed by source-specific submission checks at the current pose.
VLM-visible target	Current-view semantic evidence and geometric support; the visible-probe override follows the domain configuration.
Context/visual region	Navigation or observation authority; requires grounded evidence for the active instance before proposing STOP.

Online verification prompts. The full-goal text combines the instance description with the active visit's intent, navigation hint, shared service goal, and subtask title, including the target category

when needed. The following condensed templates describe the two image-based checks and their common response format.

Current-view verification. Supply the current RGB scene with the candidate outlined in red, a candidate crop when available, the full-goal text, and the candidate category. Ask whether the highlighted object is the requested instance. Check its appearance, room, spatial relations, and service context using the scene and crop together. Choose `uncertain` when required details are not visible or several same-category objects remain plausible.

Stored-view verification. Supply a montage of one candidate's stored views, each pairing an outlined scene with a crop, together with the full-goal text and candidate category. In target mode, ask whether the combined views identify the requested instance; required details may be supported across different views. In context mode, ask whether the object or its surroundings provide a strong location cue for finding the target. Choose `uncertain` when the combined views leave the requested identity or context unresolved.

Response. Return a JSON object with `decision` in {`yes`, `no`, `uncertain`}, a `confidence` value in $[0, 1]$, and lists named `matched_evidence`, `contradicting_evidence`, and `missing_evidence`. Each list describes visible support, contradictions, or required details still unverified.

These responses update candidate verification and inform the applicable submission branch. Missing or invalid decisions are treated as uncertain. Stored-view context verification guides search anchors; active-target proposals proceed through the trust, current-pose, and geometric checks above.

If a proposal is rejected, $\mathcal{A}_t^{\mathrm{rec}}$ contains the responses authorized by the current recovery branch: scanning, re-observation, rescue navigation, candidate switching, a safe turn, or qualified ABANDON. Rejection updates negative evidence and cooldowns; repeated hard rejection can suppress a candidate. Observation and candidate revision continue the active visit, while watchdog and repeated-rejection branches can end it through ABANDON when their conditions are met. An issued STOP is separately scored by the evaluator and ends the visit whether valid or invalid.

Abandonment follows branch-specific budget and search-exhaustion rules. In the reference configuration, the belief controller enters coverage mode while its agent-side accepted-STOP counter is below four; this counter tracks authorized STOP records rather than evaluator successes. Nearby target candidates are first considered for STOP or scanning. If neither branch returns an action, coverage-mode budget checks can authorize ABANDON: the hard budget is 220 visit actions, while the soft budget of 140 actions additionally requires the controller's no-progress counter to reach 60 and either target navigation utility below 0.25 or no region candidate with utility at least 0.20. Both branches follow a 40-action minimum-effort check. The no-progress counter takes the maximum of navigation/search-stall counters and 60 times the region-loop strike count; a no-frontier state raises it to at least 60. The hard-budget branch terminates on elapsed visit actions independently of this counter and candidate utility.

The separate target-exhaustion path checks minimum effort, sustained non-progress, confirmed-target and memory-probe availability, and the belief state. Its near-evidence protection can invoke rescue before abandonment, and loop-recovery branches likewise apply their configured rescue checks. An issued STOP, timeout, or authorized abandonment follows the same visit transition in Equation 4, retaining the realized pose and reusable evidence for the next prescribed target. The agent-side transition record follows the definition of $\hat{y}_k$ in Appendix C.2.

C.5 DOMAIN-SPECIFIC CONFIGURATION

The two policies share the evidence and decision representation. HSSD adds aliases for its 28 target categories and uses the domain-specific submission settings in Table 16. Appendix G reports the runtime environment, models, and execution settings.

The HSSD configuration also disables the soft-target STOP override. Cross-domain results therefore characterize each scene-family/configuration pair; the within-domain comparisons and ablations provide the method evidence.

Table 16: Selected reference configuration values. The common policy-side requirements apply to their named branches; neither column changes the evaluator's 1-m success radius.

Setting	HM3D	HSSD
Simulator agent radius	0.17 m	0.18 m
VLM-visible probe STOP override enabled	True	False
Local unverified branch: minimum visit actions	8	12
Local unverified branch: maximum object distance	0.90 m	0.75 m
Local unverified branch: require promotion	True	True
Local unverified branch: require current verification	True	True
Near-promoted local STOP override enabled	True	False
Full-goal STOP: require current verification	True	True
Full-goal VLM marks final STOP trusted	False	False

D Comparative Evidence and Completion Challenges

The supplementary experiments develop the three findings in Section 6: complete-request performance exposes progress left outside fulfilled requests, continuous execution permits later requests to complete after earlier failures, and the PEARS configurations connect evidence reuse and candidate control to completion and terminal behavior. The analyses use frozen evaluation records; narrower diagnostics appear in Appendix E. These comparisons evaluate the implemented navigation systems and do not separately identify the effect of service-context language.

D.1 Navigation Mechanisms and Long-Term Baseline Adapters

UniGoal (Yin et al., 2025) guides navigation by matching goal and scene graphs. Context-Nav (Jang & Kim, 2026) combines context-guided exploration with viewpoint-aware 3D verification. MemoryExplorer (Wang et al., 2026a) couples active memory retrieval with proactive exploration. GOAT SenseAct-NN (Khanna et al., 2024b) uses a recurrent navigation policy with implicit scene memory. PEARS combines selective persistence, typed evidence arbitration, and risk-aware submission and recovery; Appendix C details these mechanisms.

All adapters execute the prescribed target stream through the same visit-transition interface and evaluator. Simulator state and pose continue across visits, with method-specific internal state retained. Table 17 distinguishes the information available at the interface from the fields actually consumed by each adapter.

Table 17: Benchmark adapter behavior. Every method advances on a terminal visit outcome; the retained and rebuilt state is method-specific.

Adapter	Consumed goal input	Visit-boundary state and control
UniGoal-LT (Yin et al., 2025)	Category, description, intent, navigation hint, service goal construct the goal graph via local Qwen.	Retain map and scene graph; rebuild goal graph, selected node, navigation goal and counters. Five standard actions.
Context-Nav-LT (Jang & Kim, 2026)	Category and instance description form intrinsic target text; intent, hint and service goal form context text.	Retain scene geometry and mapped instances; refresh target/value state, exploration mask and local controller. Five actions.
MemoryExplorer-LT (Wang et al., 2026a)	Category and public descriptive/service fields form the object-search instruction for the MemoryExplorer agent.	Retain episode map, view history and model session; clear selected snapshot, local navigation target and visit counters. Five actions.
GOAT SenseAct-NN-LT (Khanna et al., 2024b)	Encode the instance description, falling back to category. Shared service fields are not part of this text embedding.	Retain recurrent hidden state; update goal text/embedding. Seven actions, including the policy's 30° LookUp/LookDown.

UniGoal-LT and Context-Nav-LT use the local Qwen3.5-9B backend. MemoryExplorer-LT combines MemoryExplorer, CLIP, YOLO-World and SAM through an observation-only sequential-ObjectNav adapter. GOAT uses the pretrained monolithic SenseAct-NN policy with OpenAI CLIP

RN50 goal embeddings and stochastic action selection. Its retained camera-pitch actions are marked † in comparisons. All task metrics are computed using SSLongNav's evaluator and hierarchy. The resulting comparison concerns these benchmark adaptations, including their input processing, retained state, and action interfaces; it does not establish a ranking on the methods' original tasks.

D.2 Paired Comparisons and Instance-Level Performance

Figure 7 complements the main comparison with SP–SSC profiles for all methods. The separation shows how much within-request progress remains outside fully completed subtasks, with both metrics averaged over the same subtasks.

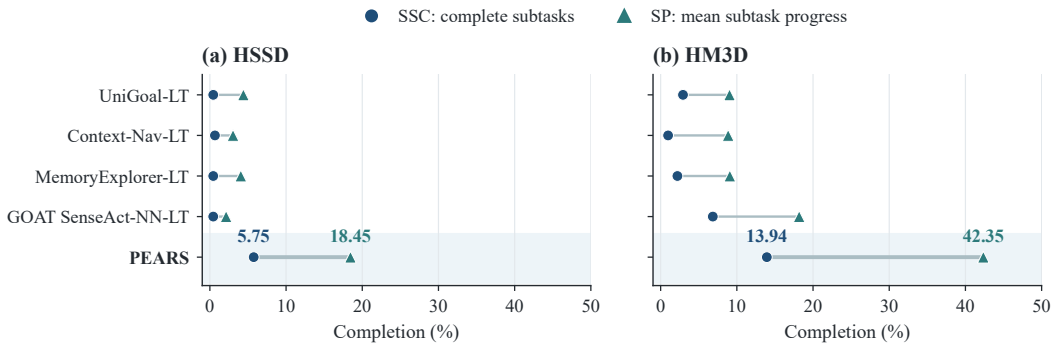


Figure 7: Partial progress and strict completion across methods. Each line connects SSC (circles) to SP (triangles), with length equal to SP minus SSC. Both metrics weight subtasks equally. PEARS is highlighted.

Paired comparisons support the strict-completion advantage reported in the main paper. The aligned-record analysis includes Context-Nav-LT, GOAT SenseAct-NN-LT and PEARS on identical scene/episode keys; the main comparison additionally reports UniGoal-LT and MemoryExplorer-LT. Table 18 gives the differences for the aligned methods.

Statistical conventions. Before analysis, frozen records are checked against their episode, scene, subtask and visit counts and their formal metrics. Paired intervals use 10,000 scene-cluster bootstrap draws with seed 20260827, retaining every episode from a resampled scene. The length/position and calibration analyses use the same cluster unit with their stated seeds. These intervals quantify scene-sampling variation of the frozen runs. In a stratum with no observed successes, empirical bootstrap completion intervals degenerate to $[0, 0]$; this reflects the observed support rather than establishing a zero population success probability. Random-partition ranges in Appendix D.5 instead describe variation over legal boundary definitions. SSC/SP/SS-SPL weight subtasks; IVC and terminal-outcome rates weight visits.

Table 18: Paired PEARS minus adapted-agent differences on identical frozen records, in points. Each cell is point difference [scene-cluster 95% interval]. SSC, SP, and IVC are percentages; SS-SPL is dimensionless and shown as $\times 100$.

Domain	Baseline	ΔSSC	ΔSS-SPL	ΔSP	ΔIVC
HSSD	Context-Nav-LT	5.09 [2.72, 7.56]	3.39 [1.99, 4.88]	15.41 [12.16, 18.98]	16.74 [13.61, 19.93]
	GOAT SenseAct-NN-LT†	5.31 [3.10, 7.69]	3.19 [1.69, 4.77]	16.32 [13.38, 19.50]	17.45 [14.79, 20.26]
HM3D	Context-Nav-LT	12.96 [9.14, 17.00]	7.11 [5.25, 9.08]	33.49 [29.48, 37.87]	35.35 [31.68, 39.32]
	GOAT SenseAct-NN-LT†	7.09 [2.41, 11.86]	1.84 [-1.16, 4.88]	24.17 [20.16, 28.26]	24.38 [19.95, 28.57]

Against Context-Nav-LT, PEARS gains 5.09 and 12.96 SSC percentage points on HSSD and HM3D; against GOAT, the gains are 5.31 and 7.09. All four SSC intervals lie above zero. Thus, the paired differences extend to complete subtasks. The HM3D GOAT SS-SPL difference is positive, but its interval includes zero; the reported interval does not establish an efficiency-weighted advantage for that comparison.

Distinct-instance requests within a category. A visit belongs to the same-category-distinct-ID (SCDI) stratum when at least two requested IDs in its episode share its normalized category; repeated visits to one ID alone do not qualify. Normalization lowercases labels, replaces underscores and hyphens with spaces, and collapses whitespace, without merging synonyms. Other visits form the comparison stratum. These descriptive strata also differ in category, layout and language composition. Invalid STOP identifies an unsuccessful submission; scoring geometry is characterized in Appendix B.8.

Table 19: Grounded-instance outcome strata for the aligned-record methods. All outcome columns are visit-level percentages; n_c and n_v count represented scenes and visits. SCDI denotes a requested category with at least two distinct object IDs in the episode. Scenes may contribute to both strata, so scene counts are not additive.

Domain	Method	Stratum	n_c	n_v	IVC	Inv. STOP	Timeout	Abandon
HSSD	Context	SCDI	43	1173	3.32	23.19	0.00	73.49
		Other	41	225	3.11	15.11	0.00	81.78
	GOAT†	SCDI	43	1173	2.73	77.24	20.03	0.00
		Other	41	225	1.78	66.67	31.56	0.00
	PEARS	SCDI	43	1173	18.84	52.26	9.89	19.01
		Other	41	225	26.22	40.44	8.89	24.44
HM3D	Context	SCDI	35	1032	9.01	26.07	0.00	64.92
		Other	35	773	9.83	16.82	0.00	73.35
	GOAT†	SCDI	35	1032	19.77	73.06	7.17	0.00
		Other	35	773	21.09	68.31	10.61	0.00
	PEARS	SCDI	35	1032	41.57	32.66	6.40	19.38
		Other	35	773	48.90	22.51	5.56	23.03

Table 20: SCDI-minus-Other outcome contrasts corresponding to Table 19, in percentage points. Each cell is point difference [scene-cluster 95% interval].

Domain	Method	ΔIVC	ΔInv. STOP	ΔTimeout	ΔAbandon
HSSD	Context-Nav-LT	0.21 [-3.04, 2.79]	8.08 [2.52, 13.33]	0.00 [0.00, 0.00]	-8.29 [-13.28, -2.81]
	GOAT SenseAct-NN-LT†	0.95 [-2.71, 3.64]	10.57 [1.58, 19.65]	-11.52 [-19.58, -3.79]	0.00 [0.00, 0.00]
	PEARS	-7.38 [-17.71, 1.06]	11.81 [5.75, 20.24]	1.00 [-2.45, 4.68]	-5.43 [-11.74, -0.32]
HM3D	Context-Nav-LT	-0.82 [-5.84, 4.03]	9.25 [5.16, 13.11]	0.00 [0.00, 0.00]	-8.43 [-13.41, -3.36]
	GOAT SenseAct-NN-LT†	-1.32 [-7.16, 4.83]	4.76 [-1.45, 10.66]	-3.44 [-6.15, -0.92]	0.00 [0.00, 0.00]
	PEARS	-7.33 [-12.89, -1.84]	10.15 [5.80, 14.52]	0.83 [-1.20, 2.79]	-3.65 [-7.36, 0.28]

For PEARS, SCDI IVC is 18.84% versus 26.22% for Other visits on HSSD, and 41.57% versus 48.90% on HM3D. Invalid STOP increases by 11.81 and 10.15 percentage points. Both invalid-STOP contrasts and the HM3D IVC contrast have intervals excluding zero; the HSSD IVC contrast spans zero. These descriptive associations motivate examining instance grounding and submission together. They do not isolate the effect of same-category distractors: the strata also differ in targets, language, and layout, and distinct IDs need not have disjoint scoring regions.

D.3 Subtask Length and Episode Position

Length stratification shows how successful visits combine within groups, while position stratification shows how terminal behavior evolves along the ongoing request stream. Both use the full PEARS records: 45 HSSD episodes, 452 subtasks and 1,398 visits; 70 HM3D episodes, 409 subtasks and 1,805 visits. Tables 21 and 22 expand the length analysis in the main paper. Figure 8 adds the complete outcome distribution across episode position.

For length strata, SSC/SP are averaged over subtasks of the stated length, and terminal outcomes over their visits. For position, visit $t \in \{0, \ldots, T-1\}$ has progress $p = t/\max(T-1, 1)$ and one of five equal-width bins, including $p = 1$ in the fifth. Intervals use 10,000 scene-cluster draws with seed 20260728. S, I, T and A denote success, invalid STOP, timeout and abandonment.

For multi-visit subtasks, partial progress substantially exceeds strict completion. HSSD's five-visit groups attain 18.67% SP but 0.00% SSC. HM3D's seven-visit groups attain 51.54% SP and 3.92% SSC, compared with 5.66% SSC for eight-visit groups. SSC is non-monotonic across lengths. Differences in target and context composition prevent interpreting this stratification as a causal effect of adding visits.

Table 21: HSSD challenge diagnostics by subtask length. SSC and SP are point [scene-clustered descriptive 95% interval]; terminal columns are visit-level point percentages. n_s and n_v count subtasks and visits.

L	n_s	n_v	SSC [95% CI]	SP [95% CI]	S	I	T	A
1	33	33	6.06 [0.00, 17.50]	6.06 [0.00, 17.50]	6.06	60.61	9.09	24.24
2	171	342	9.36 [5.07, 13.69]	17.25 [11.43, 23.51]	17.25	56.73	9.36	16.67
3	100	300	3.00 [0.00, 6.52]	18.00 [14.06, 22.76]	18.00	55.00	10.00	17.00
4	60	240	6.67 [1.64, 12.24]	25.00 [19.32, 30.86]	25.00	46.67	8.33	20.00
5	45	225	0.00 [0.00, 0.00]	18.67 [13.85, 23.18]	18.67	44.89	11.56	24.89
6	43	258	2.33 [0.00, 7.32]	24.42 [19.38, 29.73]	24.42	43.41	9.69	22.48

Table 22: HM3D challenge diagnostics by subtask length, following the notation and interval construction of Table 21.

L	n_s	n_v	SSC [95% CI]	SP [95% CI]	S	I	T	A
1	19	19	21.05 [0.00, 40.00]	21.05 [0.00, 40.00]	21.05	42.11	21.05	15.79
2	97	194	22.68 [14.89, 30.43]	38.66 [31.48, 45.76]	38.66	34.54	6.70	20.10
3	59	177	15.25 [3.57, 29.09]	38.98 [30.43, 48.15]	38.98	36.16	3.39	21.47
4	51	204	19.61 [5.26, 31.25]	45.59 [32.14, 57.44]	45.59	25.00	8.33	21.08
5	44	220	13.64 [5.00, 21.74]	49.09 [41.21, 58.28]	49.09	22.73	6.36	21.82
6	35	210	2.86 [0.00, 10.71]	38.10 [27.78, 46.67]	38.10	25.71	8.10	28.10
7	51	357	3.92 [0.00, 9.30]	51.54 [44.67, 58.81]	51.54	26.89	3.36	18.21
8	53	424	5.66 [0.00, 12.20]	45.75 [40.10, 51.86]	45.75	28.54	6.13	19.58

For single-visit subtasks, SP and SSC coincide. Singletons use deterministic text templates, so the $L = 1$ stratum differs from multi-visit groups in both size and language source.

Figure 8 shows that later visits continue to produce both success and several distinct terminal outcomes. From the first to final position bin, HSSD IVC changes from 16.67% to 20.07%, with invalid STOP changing from 49.66% to 54.42%. HM3D IVC changes from 50.54% to 46.44%. Intermediate bins are non-monotonic. Position also changes the target, spatial and episode composition, so these profiles characterize sustained-navigation behavior rather than a causal effect of elapsed execution.

Table 23: HSSD terminal composition by fixed-width episode-progress bin. Every percentage is reported as point [scene-clustered descriptive 95% interval]. Bins 1–5 correspond to $[0, .2), [.2, .4), [.4, .6), [.6, .8), [.8, 1]$.

Bin	n_v	IVC [95% CI]	Invalid STOP [95% CI]	Timeout [95% CI]	Abandon [95% CI]
1	294	16.67 [11.56, 22.30]	49.66 [43.58, 56.06]	11.56 [7.82, 15.51]	22.11 [17.26, 27.15]
2	269	19.33 [14.29, 24.45]	48.70 [42.59, 55.04]	8.92 [5.34, 12.90]	23.05 [18.25, 27.90]
3	272	22.79 [17.67, 27.94]	47.06 [40.14, 54.17]	8.09 [4.91, 11.40]	22.06 [17.34, 27.07]
4	269	21.56 [16.42, 26.89]	51.67 [46.45, 56.93]	10.41 [7.58, 13.48]	16.36 [12.73, 20.07]
5	294	20.07 [14.53, 26.06]	54.42 [48.37, 60.64]	9.52 [6.03, 12.99]	15.99 [12.07, 20.00]

Table 24: HM3D terminal composition by fixed-width episode-progress bin, following Table 23.

Bin	n_v	IVC [95% CI]	Invalid STOP [95% CI]	Timeout [95% CI]	Abandon [95% CI]
1	370	50.54 [44.22, 57.10]	28.38 [22.99, 33.71]	5.68 [3.67, 7.93]	15.41 [11.29, 20.16]
2	353	41.08 [35.43, 46.89]	28.33 [24.53, 32.34]	6.80 [4.32, 9.50]	23.80 [18.60, 29.36]
3	350	39.14 [33.14, 45.43]	29.43 [24.43, 34.27]	7.43 [4.50, 10.70]	24.00 [20.00, 28.33]
4	353	45.89 [40.29, 51.55]	26.35 [21.57, 31.06]	5.38 [3.53, 7.24]	22.38 [18.63, 26.29]
5	379	46.44 [40.06, 52.84]	29.02 [24.43, 33.89]	5.01 [2.88, 7.37]	19.53 [15.32, 23.77]

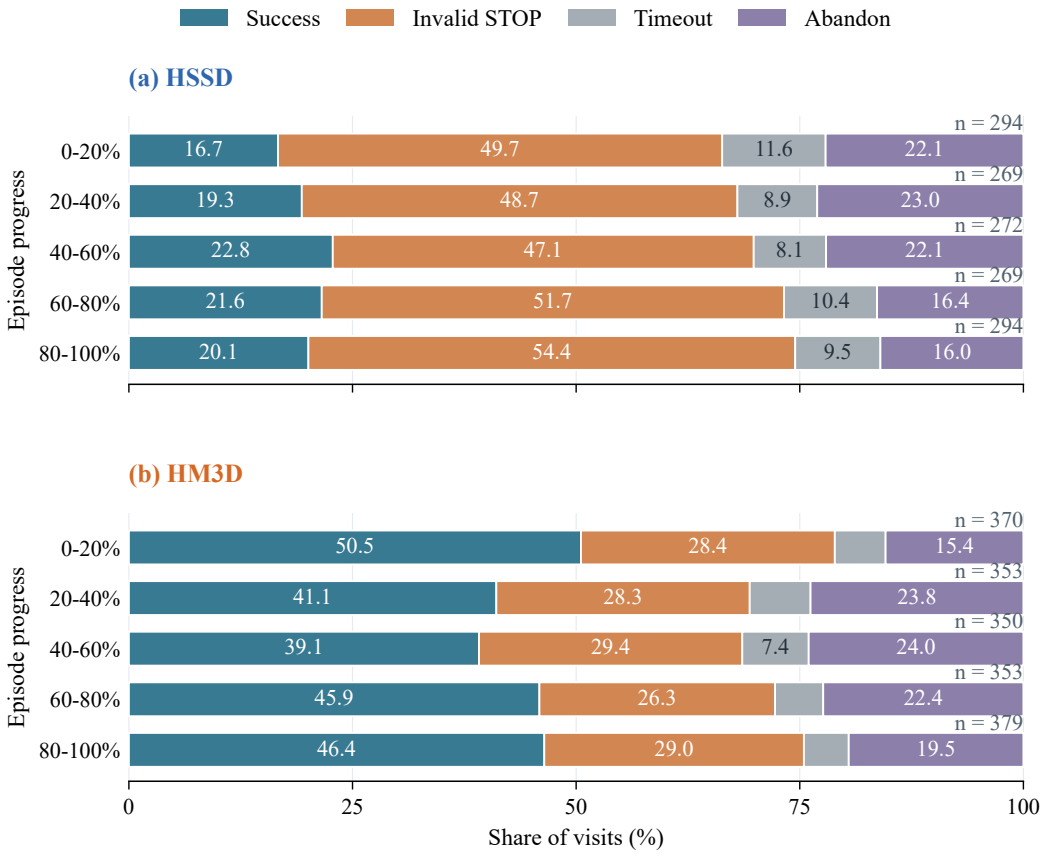


Figure 8: PEARS terminal outcomes across five equal-width episode-progress bins. Each horizontal bar partitions visits in one bin into success, invalid STOP, timeout and abandonment; n is the number of visits. Both domains use the same percentage scale. Tables 23 and 24 give exact percentages and scene-cluster intervals.

D.4 Success Composition and Continued Execution

For PEARS, we examine how visit successes form complete requests, how often they persist over consecutive windows, and what remains achievable after failure. Table 25 summarizes strict completion, its episode-conditioned exchangeability reference, and rolling-window completion (RWC_k).

Table 25: Complementary PEARS reliability summaries of visit success, strict completion, exchangeability null, and boundary-free consecutive completion (%).

Domain	IVC	SSC	Null SSC	RWC_2	RWC_3	RWC_4	RWC_6
HSSD	20.03	5.75	3.08	7.32	2.68	0.71	0.09
HM3D	44.71	13.94	9.93	28.70	18.14	11.40	4.52

Success composition within requests. For episode e, let N_e be its number of visits, M_e its number of successful visits, and k_u the benchmark length of subtask u. Under exchangeability of the fixed success multiset within an episode, the exact probability that all visits in u are successful is

$$p_u^{\mathrm{null}} = \frac{\binom{M_e}{k_u}}{\binom{N_e}{k_u}}, \qquad \mathrm{SSC}_{\mathrm{null}} = \frac{1}{|\mathbb{U}_{\mathcal{E}}|} \sum_{u \in \mathbb{U}_{\mathcal{E}}} p_u^{\mathrm{null}},$$

with $p_u^{\mathrm{null}} = 0$ when $M_e < k_u$. This null conditions on episode-level competence and benchmark group lengths while randomizing only where successes fall. The 95% intervals below use 10,000 scene-cluster bootstrap draws with seed 20260824 and retain all records from each resampled scene.

Table 26: Episode-conditioned calibration from unrounded estimates. SSC (%), differences (percentage points), and observed/null ratios are shown with scene-cluster bootstrap 95% intervals.

Domain	Observed SSC	Null SSC	Observed–null	Ratio
HSSD	5.75 [3.67, 7.99]	3.08 [2.17, 4.05]	2.68 [0.82, 4.60]	1.87 [1.26, 2.63]
HM3D	13.94 [10.33, 17.77]	9.93 [7.78, 12.22]	4.01 [1.46, 6.57]	1.40 [1.15, 1.67]

Observed SSC exceeds the episode-conditioned null in both domains: 5.75% versus 3.08% on HSSD, and 13.94% versus 9.93% on HM3D (1.87× and 1.40×). Table 26 reports observed-minus-null differences of 2.68 and 4.01 percentage points, with both intervals above zero. Even after conditioning on each episode's total successes and request lengths, their arrangement yields more complete requests than the exchangeability reference predicts. This motivates reporting request completion alongside visit totals. Spatial locality, target difficulty, and policy state can contribute to that arrangement; the comparison does not isolate service language or a policy component.

Sustained success across consecutive visits. Let $S_{e,t} \in \{0, 1\}$ be the success of visit t in episode e, and let T_e be its number of visits. Define the eligible-window count $W_{e,k} = \max(T_e - k + 1, 0)$. We compute

$$\mathrm{RWC}_k = \frac{\sum_e \sum_{t=1}^{W_{e,k}} \prod_{j=0}^{k-1} S_{e,t+j}}{\sum_e W_{e,k}}, \qquad k \in \{2, 3, 4, 6\}.$$

Episodes shorter than k contribute no windows; an empty sum is zero, and RWC is defined when the total eligible-window count is positive. Unlike SSC, this measure never consults a benchmark subtask ID or cut position.

For $k = (2, 3, 4, 6)$, HSSD has $(1{,}353, 1{,}308, 1{,}263, 1{,}173)$ eligible windows, of which $(99, 35, 9, 1)$ are fully successful. HM3D has $(1{,}735, 1{,}665, 1{,}596, 1{,}460)$ eligible windows, of which $(498, 302, 182, 66)$ are fully successful. These counts reproduce the RWC percentages in Table 25.

Only 0.09% of HSSD six-visit windows and 4.52% of HM3D six-visit windows are entirely successful. RWC characterizes sustained success across the continuous stream, including windows that cross prescribed subtask boundaries. Overlapping windows share visits and are not independent completion trials. Figure 5(c) displays the supplied lower and upper bounds alongside these point estimates; no confidence level is assigned to those bounds here.

Navigation after unsuccessful visits. For continuation, the last visit of each episode is excluded because it has no successor. Table 27 reports next-visit success conditioned on success, invalid STOP, timeout or abandonment of the preceding visit.

Table 27: Failure-conditioned continuation profile. Values are next-visit success % [scene-cluster 95% interval]; n counts eligible transitions.

Domain	Previous outcome	n	Next-visit success
HSSD	Success	275	36.00 [29.52, 41.99]
	Invalid STOP	674	14.99 [12.45, 17.72]
	Timeout	132	15.91 [9.76, 22.05]
	Abandonment	272	19.12 [14.93, 23.51]
HM3D	Success	788	63.20 [58.86, 67.24]
	Invalid STOP	488	32.79 [29.30, 36.28]
	Timeout	105	34.29 [24.07, 45.00]
	Abandonment	354	24.58 [19.83, 29.65]

Next-visit success after success is 36.00% on HSSD and 63.20% on HM3D; after invalid STOP it is 14.99% and 32.79%. Successful visits also occur after timeout and abandonment. Continuing the episode therefore exposes useful navigation progress that stopping at the first failure would omit, while preserving the current subtask's strict failure outcome.

A separate within-episode diagnostic averages success-after-success minus success-after-failure across the 45 HSSD and 69 HM3D episodes containing both transition types. The differences are 13.85 [7.61, 20.02] and 26.30 [21.05, 31.46] percentage points. This estimand averages episode-level contrasts and pools the unsuccessful predecessor types; it is not obtained by subtracting the pooled rates in Table 27. Conditioning on episode reduces between-episode variation, while target difficulty, spatial locality, and policy state remain associated with the observed transitions.

Completing subsequent requests. We count a completed subtask as occurring after failure when at least one earlier subtask in the same episode is incomplete. Each subtask is counted once in the original order; the earlier failure need not be immediately preceding. Of the 26 completed HSSD subtasks, 21 meet this condition, including 19 multi-visit subtasks. On HM3D, 41 of 57 completed subtasks meet the condition, including 37 multi-visit subtasks. These are whole requests completed within episodes that are already unsuccessful under an all-requests criterion. Separate subtask outcomes retain this distinction throughout the run. The counts describe where completed requests occur; they are not conditional success rates over all requests attempted after failure.

D.5 Subtask-Boundary Sensitivity Audit

This audit changes only the evaluator partition over an already executed visit stream. For every episode, the ordered success vector, terminal outcomes, trajectory, and language observed during execution remain frozen. G0 is the benchmark spatial grouping. G1 samples uniformly from the unique permutations of the episode's benchmark subtask-length multiset, so repeated lengths are not over-weighted. G2 uses dynamic-programming completion counts to sample uniformly from all positive integer compositions with the benchmark number of visits and subtasks, subject to maximum length 6 on HSSD and 8 on HM3D. G3 forms fixed contiguous chunks of length 2, 3, or 4 and retains a final remainder; because this changes the number of groups, it is reported only as a stress test. G4 samples uniformly from the unique legal partitions obtained by moving one original cut one visit left or right. G1, G2, and G4 each use 10,000 full-split draws with seed 20260811. Figure 9 summarizes the resulting alternative-boundary SSC; Tables 28 and 29 provide the corresponding values.

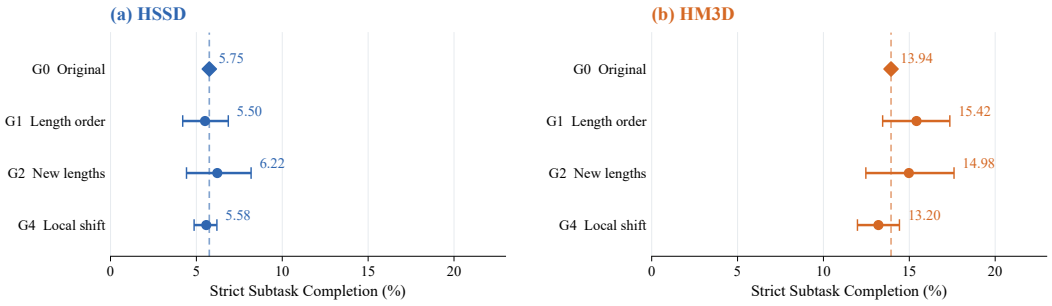


Figure 9: Strict completion under alternative partitions of the frozen PEARS visit outcomes. Dashed lines mark the original grouping G0. Points show mean SSC across 10,000 legal partitions for G1/G2/G4; bars show 2.5th–97.5th percentiles over partitions, not confidence intervals. Both panels share an SSC scale. Grouping rules appear in Appendix D.5.

Table 28: Boundary sensitivity of SSC (%) across methods on HSSD and HM3D. Each random-scheme entry is mean [2.5th, 97.5th percentile] over legal partition draws. These ranges quantify variation across boundary definitions and are not confidence intervals.

Method	G0	G1	G2	G4
HSSD				
UniGoal-LT	0.44	0.81 [0.22, 1.55]	1.42 [0.44, 2.43]	0.58 [0.22, 1.11]
Context-Nav-LT	0.66	0.76 [0.44, 1.33]	1.10 [0.44, 1.99]	0.59 [0.22, 0.66]
MemoryExplorer-LT	0.44	0.61 [0.00, 1.11]	1.01 [0.22, 1.99]	0.51 [0.22, 0.88]
GOAT SenseAct-NN-LT†	0.44	0.18 [0.00, 0.66]	0.47 [0.00, 1.11]	0.45 [0.22, 0.66]
PEARS	5.75	5.50 [4.20, 6.86]	6.22 [4.42, 8.19]	5.58 [4.87, 6.19]
HM3D				
UniGoal-LT	2.93	1.16 [0.49, 1.96]	1.20 [0.24, 2.20]	1.96 [1.47, 2.44]
Context-Nav-LT	0.98	1.46 [0.73, 2.44]	1.91 [0.73, 3.18]	0.82 [0.49, 1.22]
MemoryExplorer-LT	2.20	0.93 [0.24, 1.71]	1.64 [0.49, 2.93]	1.90 [1.47, 2.20]
GOAT SenseAct-NN-LT†	6.85	5.36 [3.91, 6.85]	5.82 [4.16, 7.58]	6.06 [5.38, 6.60]
PEARS	13.94	15.42 [13.45, 17.36]	14.98 [12.47, 17.60]	13.20 [11.98, 14.43]

For PEARS, the original G0 SSC lies within the G1, G2, and G4 percentile ranges in both domains (Table 28; Figure 9). Alternative partitions change mean SSC, but none of these three ranges ex-

cludes the original PEARS score. PEARS has the highest G0 SSC and the highest mean SSC under each random scheme in both domains; this comparison of means does not assert a ranking in every draw. The ranges describe variation over evaluator partitions, not sampling confidence intervals. Because the policy's executed stream and observed language are fixed, this audit does not test how the policy would behave under newly supplied subtask contexts. RWC in Appendix D.4 measures consecutive success without using these cuts.

Table 29: Fixed-chunk SSC stress tests for PEARS (%). G3 partitions each episode into contiguous chunks of length L and retains a final remainder, changing the number of scored groups.

Domain	$L = 2$	$L = 3$	$L = 4$
HSSD	8.13	2.74	1.10
HM3D	28.40	18.68	11.53

Table 29 reports fixed-chunk completion for three group lengths. Together with the random partitions, these results quantify the dependence of absolute SSC on the evaluator's completion units while holding the executed visit stream fixed.

Before sampling, the audit reproduces G0 counts and metrics and verifies that subtask lengths match contiguous visit runs. Each regrouping preserves visit outcomes and IVC.

D.6 System Ablations: Completion and Terminal Behavior

The full configuration achieves the highest SSC, SP, and IVC among these HM3D variants (Table 31). No instance memory has slightly higher SS-SPL than the full configuration (7.18 versus 7.16 on the displayed $\times 100$ scale). Table 30 specifies the switches; their downstream candidate, recovery, and submission paths can change together. "Combined" denotes the same no-belief plus relaxed-verification configuration as the final row of Table 3.

Table 30: Scope of the system-ablation switches.

Variant	Disabled or relaxed components
No instance memory	Dedicated grounded-instance storage and retrieval
No visual memory	Visual store, similarity retrieval and its frontier utility
No structured belief	Belief controller, direct gates, hypotheses, frontier utility, structured target memory
Relaxed STOP verification	Source/current checks, geometry, record trust and associated STOP guards
Combined	Both no-belief and relaxed-verification settings

Table 31: PEARS system ablations on the 70-episode HM3D evaluation split (409 subtasks, 1,805 visits). SSC, SP, IVC, and terminal-outcome rates are percentages; SS-SPL is shown as $\times 100$. Bold marks the highest task metric.

Variant	SSC	SS-SPL	SP	IVC	Inv. STOP	Timeout	Abandon
Full configuration	**13.94**	7.16	**42.35**	**44.71**	28.31	6.04	20.94
No instance memory	10.76	**7.18**	30.86	31.25	26.76	16.73	25.26
No visual memory	11.25	6.09	35.44	37.78	27.59	6.87	27.76
No structured belief	7.09	3.56	31.64	33.91	21.99	2.66	41.44
Relaxed STOP verification	2.69	2.69	5.90	4.71	9.36	23.43	62.49
Combined	7.58	6.16	22.75	23.93	73.30	0.78	1.99

Removing instance memory lowers SSC by 3.18 and IVC by 13.46 percentage points, with more visits timing out. Removing visual memory lowers SSC by 2.69 and IVC by 6.93 percentage points, with more abandonment. These terminal shifts are consistent with the roles of instance retrieval in providing concrete target cues and visual memory in guiding where to search. They do not separately estimate the effect of retaining service-context inputs.

Removing structured belief lowers SSC by 6.85 percentage points and raises abandonment by 20.50 points, alongside fewer invalid STOPs. This switch also removes direct gates, hypotheses, frontier utility, and structured target memory, so it does not isolate the typed arbiter. With structured belief retained, relaxed verification produces the largest IVC drop (40.00 points), with more timeouts and

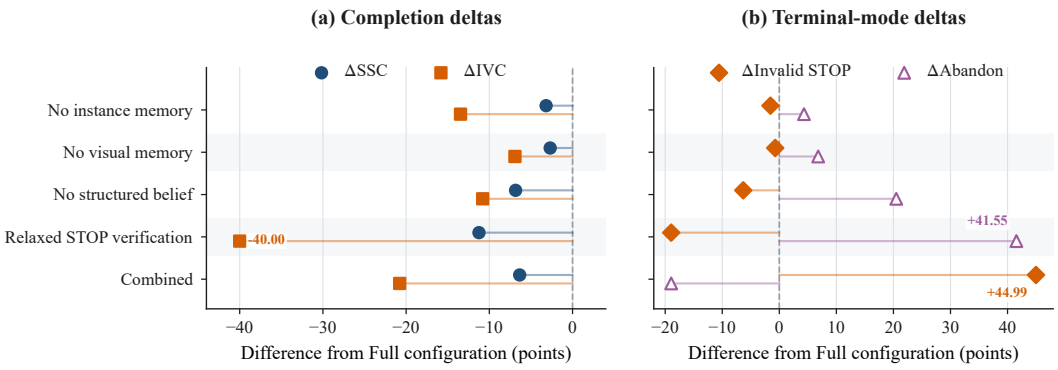


Figure 10: Changes from the full PEARS configuration in completion and terminal outcomes, in percentage points. Table 30 specifies each intervention; Table 31 reports all absolute metrics. SSC uses subtasks as its unit; the other displayed rates use visits.

abandonment. Since terminal rates use all visits, fewer invalid STOPs can accompany reduced submission and poorer completion. Combining both interventions instead raises invalid STOP by 44.99 points. These contrasting terminal profiles (Figure 10) support interpreting candidate control and submission verification jointly.

E Targeted Mechanism Diagnostics

These diagnostics complement the system ablations with interventions on persistent state and candidate roles, connecting the selective-persistence and typed-arbitration mechanisms in Section 5 to complete-request outcomes. The fixed set contains 10 HM3D episodes from 10 distinct scenes, with 64 subtasks and 286 visits per replica, using seed 100 and the first 10 episodes in the fixed evaluation ordering. The five configurations share episode keys, target order, initial poses and five paired replica IDs. For each configuration, pooling the five replicas gives 50 episode-runs, 320 subtask evaluations and 1,430 visit evaluations; the number of independent scenes remains 10. Tables 33–35 report task and terminal results under the execution-condition assumptions stated below.

E.1 Interventions and Paired Analysis

Table 32: Targeted intervention scope and assumed shared execution conditions. All configurations preserve simulator pose.

Setting	Intervention	Held fixed / scope
Reference	No diagnostic intervention	Same assumed build, checkpoint, GPU allocation and logging across all settings.
Broad state reset	Clear broad stores/maps/history; retain legacy category hypotheses.	Same 10 episodes, target order, initial poses and five paired replica IDs.
No history reranking	Remove only history/reliability reranking.	Memory capacity, top-k, thresholds and current-view verification retained.
Cross-visit instance reset	Clear registered instance stores and related history at each visit boundary.	276 boundaries per replica; 1,380 pooled; retain geometry and target-agnostic visual memory.
Type-agnostic arbitration	One scalar candidate utility; risk terms retained.	Same runtime and logging as reference; retained risk terms documented.

Differences use unrounded aggregates and 10,000 paired scene-cluster bootstrap draws (seed 20260728), under the assumed shared execution conditions in Table 32. The five replicas are repeated evaluations of the same ten scenes, so the scene-level sample size is ten. Four intervals in Table 34 lie entirely below zero: SS-SPL for broad state reset, SSC and SS-SPL for cross-visit instance-state reset, and SSC for type-agnostic arbitration. The remaining intervals include zero. We

interpret these paired contrasts and terminal behavior on this diagnostic set alongside the full-split ablations.

Across paired replicas H-1–H-5, each containing 64 subtasks, completed-subtask counts are 10–12 for the reference, 6–8 for broad state reset, 8–10 without history-aware reranking, 4–6 for cross-visit instance-state reset, and 5–7 for type-agnostic arbitration. Each intervention completes fewer subtasks than the reference in every paired replica.

Table 33: Absolute targeted-diagnostic results pooled across five replicas, with 320 subtask evaluations and 1,430 visit evaluations per setting. SSC, SP, and IVC are percentages (↑); SS-SPL is dimensionless, shown as $\times 100$ (↑).

Setting	SSC	SS-SPL	SP	IVC
Reference configuration	**17.19**	**6.07**	**38.25**	44.06
Broad state reset	10.94	2.15	36.65	**44.41**
No history-aware reranking	14.06	5.52	35.45	41.26
Cross-visit instance-state reset	7.81	3.21	33.63	40.56
Type-agnostic arbitration	9.38	4.12	34.61	38.46

Table 34: Paired differences for the targeted interventions relative to the reference configuration, pooled across five replicas and reported as point difference [scene-clustered descriptive 95% bootstrap interval]. SSC, SP, and IVC differences are percentage points; SS-SPL differences use the displayed $\times 100$ scale. Differences are computed from unrounded aggregates using 10,000 paired draws with seed 20260728.

Variant	ΔSSC [95% CI]	ΔSS-SPL [95% CI]	ΔSP [95% CI]	ΔIVC [95% CI]
Broad state reset	-6.25 [-12.92, 0.32]	-3.92 [-5.87, -1.97]	-1.60 [-8.04, 5.01]	+0.35 [-6.83, 7.73]
No history-aware reranking	-3.13 [-10.00, 4.06]	-0.55 [-3.24, 2.25]	-2.80 [-10.96, 5.09]	-2.80 [-12.37, 6.69]
Cross-visit instance-state reset	-9.38 [-15.87, -2.81]	-2.86 [-5.46, -0.23]	-4.62 [-11.11, 2.07]	-3.50 [-11.01, 4.59]
Type-agnostic arbitration	-7.81 [-14.84, -0.31]	-1.95 [-4.64, 0.93]	-3.64 [-11.14, 3.81]	-5.59 [-12.49, 1.53]

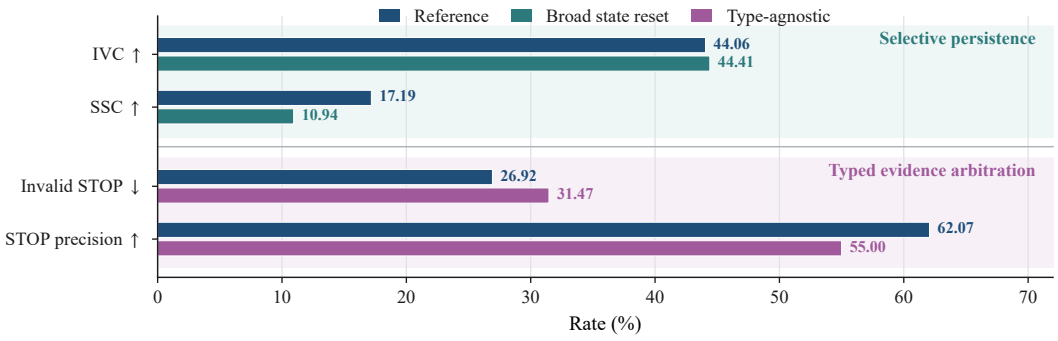


Figure 11: Targeted contrasts for persistence and candidate roles, pooled across five paired replicas. The first two rows compare reference and broad state reset; the last two compare reference and type-agnostic arbitration. Bars show point estimates. STOP precision is success among submitted STOPs; invalid STOP uses all visits. Tables 32 and 34 give intervention conditions and paired intervals.

Table 35: Terminal outcomes pooled across five replicas for the aligned targeted interventions, shown as count (percentage). Each row contains 1,430 visit evaluations.

Setting	Success	Invalid STOP	Timeout	Abandon
Reference configuration	630 (44.06)	385 (26.92)	80 (5.59)	335 (23.43)
Broad state reset	635 (44.41)	395 (27.62)	25 (1.75)	375 (26.22)
No history-aware reranking	590 (41.26)	395 (27.62)	75 (5.24)	370 (25.87)
Cross-visit instance-state reset	580 (40.56)	370 (25.87)	20 (1.40)	460 (32.17)
Type-agnostic arbitration	550 (38.46)	450 (31.47)	60 (4.20)	370 (25.87)

E.2 Persistence and the Composition of Successes

Pooled across five replicas, broad state reset yields 635 successful visits versus 630 for the reference out of 1,430 visit evaluations, but only 35 complete subtasks versus 55 out of 320 subtask evaluations. IVC is 44.41% versus 44.06%, whereas SSC falls by 6.25 percentage points. This descriptive contrast illustrates the complementarity of visit and request evaluation: similar visit totals can be distributed across different numbers of fully completed requests. The paired SSC interval spans zero (Table 34). Broad state reset clears the stores, maps, and history specified in Table 32, while retaining legacy category hypotheses; it is not a removal of every form of memory.

Resetting cross-visit instance state lowers SSC by 9.38 percentage points, with the paired interval below zero, and increases abandonment from 335 to 460 of the 1,430 pooled visit evaluations. Because this intervention retains scene geometry and target-agnostic visual memory, it specifically examines carrying registered instance state and related history into later visits. This is consistent with the role of grounded-instance retrieval in selective persistence. Removing history-aware reranking also lowers all four point estimates, but each paired interval includes zero. These findings concern the specified persistence configurations on ten scenes under the stated execution-condition assumptions.

E.3 Candidate Roles and Submission Behavior

Type-agnostic arbitration retains risk terms while changing candidate organization (Table 32). It has 450 invalid STOPs versus 385 for the reference out of 1,430 visit evaluations, and STOP precision of 55.00% (550/1,000 submissions) versus 62.07% (630/1,015). Its paired SSC difference is also below zero throughout the reported interval (Table 34). These outcomes are consistent with assigning different submission authority to target and search hypotheses, as in Section 5. STOP precision conditions on submissions, while invalid STOP uses all visits; Figure 11 displays both views. The diagnostic complements the coupled full-split ablation, with the ten-scene scope and execution-condition assumptions specified above.

F Qualitative Executions

These five qualitative cases from seed-100 PEARS reruns illustrate complete visitation, cross-subtask revisits, and progress after unsuccessful visits; they are separate from the frozen main comparison and ablation aggregates. Cases C and D come from the same episode. Table 36 summarizes their structures and outcomes. Cases A/B show selected target-evidence frames; C/D/E show terminal frames. Numbered labels identify requested targets in visit order, in-image boxes show online detections, and outer-frame colors encode evaluator outcomes. The cases explain execution and scoring, while preceding sections provide aggregate mechanism evidence.

Table 36: Qualitative cases and their evaluation roles. S/I/T denote visit success, invalid STOP, and timeout. Complete/incomplete refers to a whole subtask only when all its visits are shown.

Case	Domain	Displayed structure	Outcomes and evaluation role
A	HM3D	One eight-visit subtask	Eight S; all required visits succeed
B	HM3D	Two visits across a subtask boundary	S–S for the same instance; cross-subtask revisit
C	HSSD	One four-visit subtask	S–S–S–I; 75% progress, incomplete
D	HSSD	One six-visit subtask	I–I–S–S–S–T; 50% progress after continued execution, incomplete
E	HM3D	Two ordered, same-category instances	I–S for distinct picture IDs; instance-specific requests

F.1 Case A: Eight-Visit Strict Completion

The prescribed chain is pillow, couch, TV stand, TV, cabinet, window curtain, blanket and armchair. Figure 12 shows target evidence for every visit. All eight evaluator outcomes are successful, giving this subtask 100% progress and strict completion. The example demonstrates that complete execution of a long ordered instance sequence is achievable under the protocol.

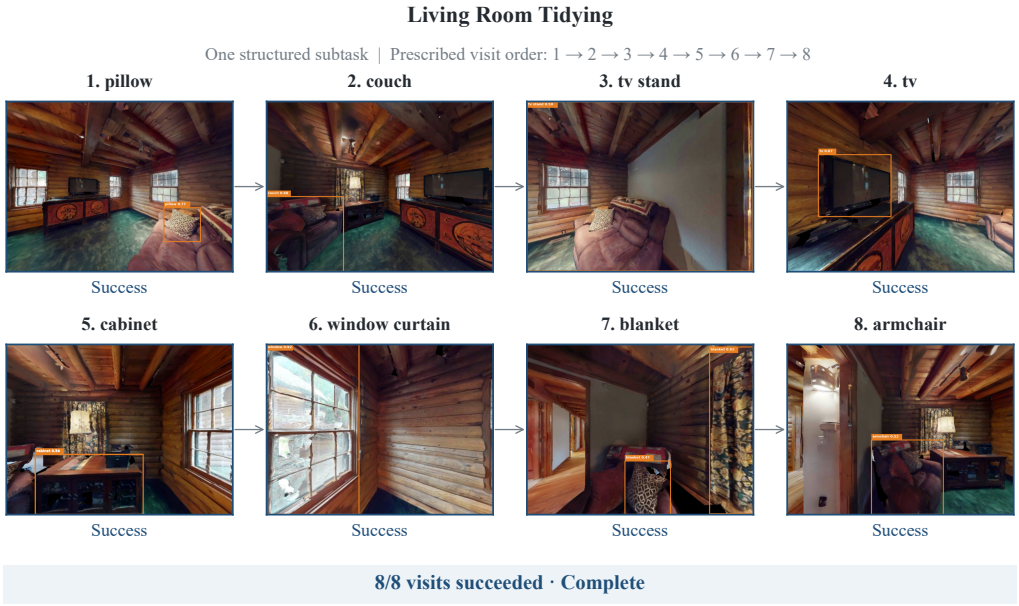


Figure 12: Eight-visit strict completion under the original service-context title. Numbered target-evidence frames follow the prescribed visit order; the outcomes are the evaluator results for those visits. All eight visits succeed.

F.2 Cases B and C: Revisit and Partial Completion

In Figure 13, `couch_132` is the final visit under "Tidy Living Room and Bedroom Area" and the first visit under "Clear Living Room Path." The requested visit and service context change while the physical instance remains the same. Both visits succeed without a pose reset.

The HSSD request in Figure 14 is "Please clear the table, move the stool and chair to the sofa, and place the cushion on the sofa." Navigation to `table_83`, `stool_4` and `chair_56` succeeds, followed by an invalid STOP for `cushion_51`. The evaluator therefore retains 75% progress and assigns zero strict completion.

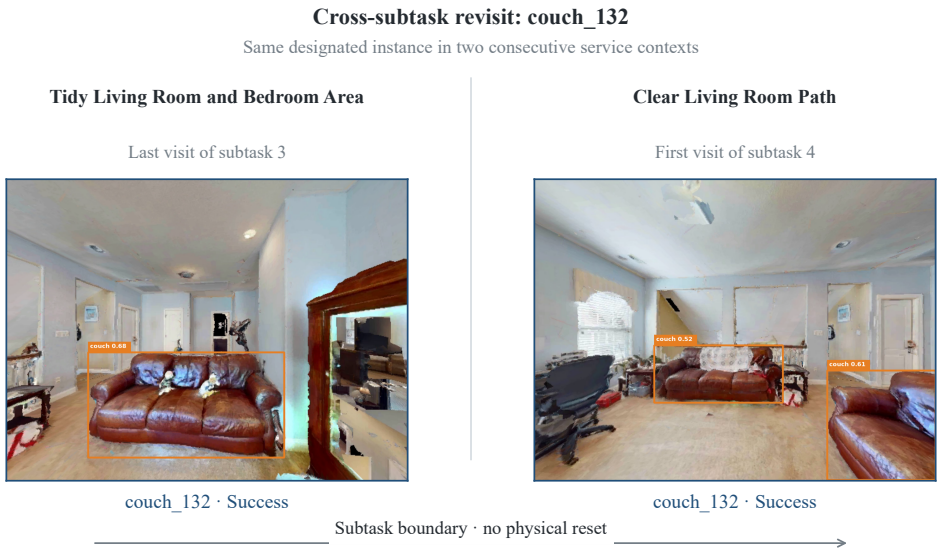


Figure 13: A cross-subtask revisit of the same designated instance. The two service contexts request `couch_132` consecutively across a subtask boundary, without a pose reset. The selected target-evidence frames and outcome labels describe these two visits.

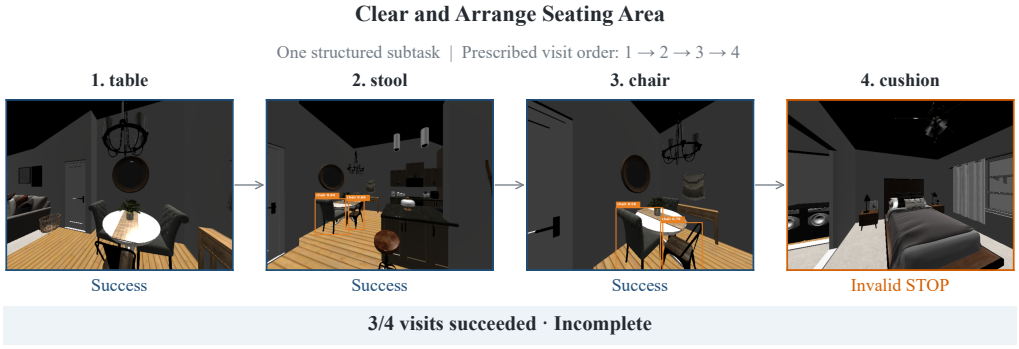


Figure 14: Partial completion of one four-visit subtask. Three successful visits are followed by an invalid STOP, giving 75% progress and an incomplete subtask. Each image is the recorded terminal frame of its visit.

F.3 Cases D and E: Continuation and Instance-Level Requests

Case D expands the six-visit timeline in Figure 5(d). Figure 15 shows three successful visits after the initial failures, while the strict score keeps the request incomplete. The terminal views follow visit order and illustrate continued progress within this request. Appendix D.4 complements this visit-level example with aggregate counts of later requests completed in full after an earlier subtask has failed.

Case E contrasts consecutive instance visits to `picture_571` and `picture_570`. Their public descriptions specify different collages: the former contains urban scenes, a map, a sculpture and architectural details; the latter contains a building facade, a map and a mechanical or sculptural close-up. The visits end in invalid STOP and success, respectively (Figure 16). This shows the descriptive distinctions present in instance-level requests. The pair is also flagged by the potential scoring-region overlap audit in Appendix B.8; the outcome labels do not identify which visual object caused a submission.

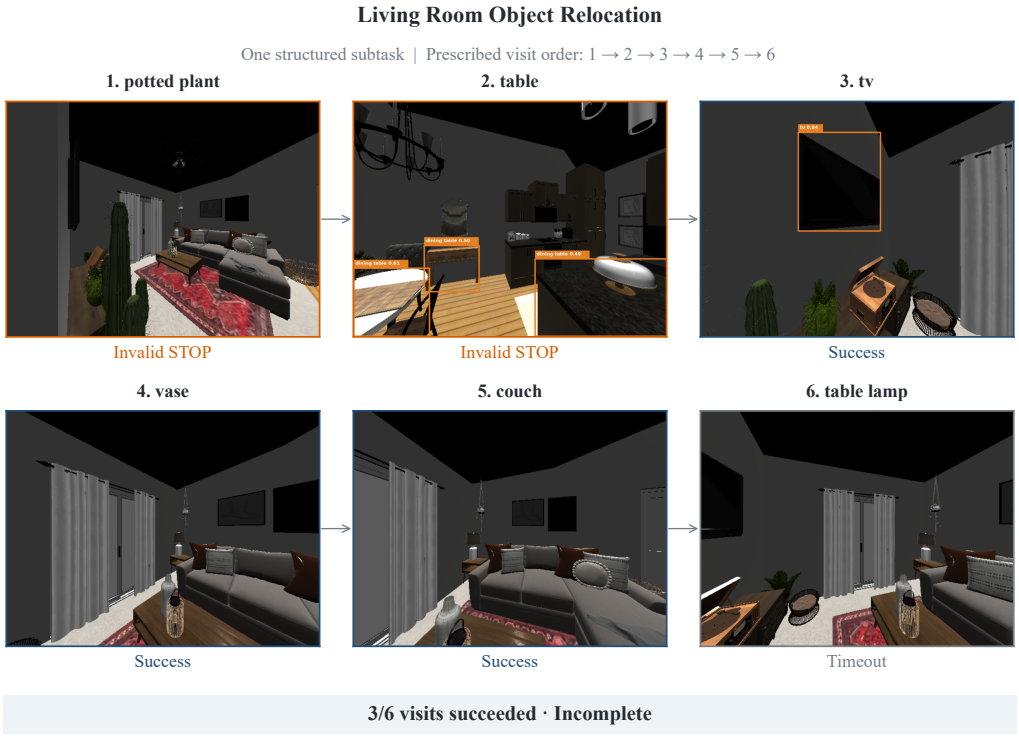


Figure 15: Continued execution within one six-visit subtask. Two invalid STOPs are followed by three successes and a timeout. The terminal frames show the prescribed visit order; three of six visits succeed, while the subtask remains incomplete.

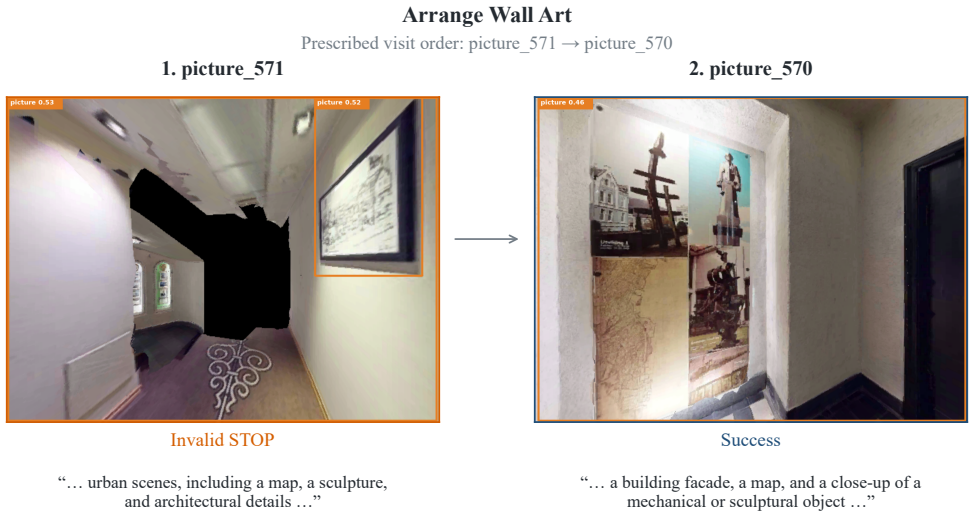


Figure 16: Ordered visits to two distinct picture instances under their original service context. Verbatim excerpts of the public descriptions distinguish the requests. Terminal frames and labels show an invalid STOP and a success, respectively.

G RUNTIME AND REPRODUCIBILITY

Environment and models. The runtime uses Python 3.10.14, Habitat-Sim/Habitat-Lab/Habitat-Baselines 0.2.5, PyTorch 2.5.1 with CUDA 12.1, Transformers 5.5.4, NumPy 1.23.5 and OpenCV-Python 4.6.0.66. PEARS uses Qwen3.5-9B, GroundingDINO Swin-T OGC, YOLO26x, MobileSAM ViT-T and SigLIP2 base patch16-384. The local Qwen service uses BF16, greedy decoding, and a default limit of 512 generated tokens. Table 16 gives the domain-specific settings.

Execution and evaluation. The reference setup uses one Qwen service and five episode workers per GPU across two GPUs. The targeted diagnostics use two NVIDIA A100-SXM4-80GB GPUs and seed 100, with intervention-specific conditions in Table 32. The separate qualitative reruns also use seed 100. Appendices A.3 and D.2 specify aggregation and statistical analysis.

Evaluation workflow. Initialize the simulator from each episode's scene and start pose, load its ordered subtasks, and initialize the agent's evidence stores. Expose the active instance specification and subtask context through the interface in Appendix A. At each step, update observations, retrieve evidence, arbitrate candidates, and execute the selected action using Appendix C. STOP, timeout, or abandonment advances to the next visit while retaining the realized pose and reusable evidence. The evaluator records visit outcomes, traveled distances, and visit-start reference distances, then aggregates complete subtasks, partial progress, and efficiency under Appendix A.3. Statistical comparisons use the scene-cluster procedures in Appendix D.2.

Availability. SSLongNav and the accompanying research materials will be made publicly available following formal publication.