PREEMPTIVE INSTRUCTION FOLLOWING WITH RESUMPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon embodied navigation may be interrupted by a higher-priority instruction before the current task sequence is complete. The agent must finish its current atomic task, execute the inserted task, and resume the remaining sequence from the updated environment state, where its viewpoint and available navigation candidates may have changed. To evaluate this capability, we introduce **PIFR** benchmark, a novel task for runtime task preemption and resumption in embodied navigation. PIFR includes single, repeated, and nested interruption settings, with stage-specific metrics for inserted-task completion, resumption of the original task, and completion of the full sequence. Evaluations show that recovery performance varies across interruption conditions, while aggregate success can obscure differences in interruption handling and resumption. We further propose a lightweight candidate-ranking baseline to improve target re-grounding after interruption. Together, PIFR and the baseline provide a systematic framework for studying interruption handling and task recovery in long-horizon embodied navigation.

1 INTRODUCTION

Embodied navigation research has advanced the ability of agents to understand their surroundings, follow instructions, and complete navigation tasks through work on vision-language navigation (Anderson et al., 2018), embodied simulation (Savva et al., 2019), interactive navigation (Thomason et al., 2020), and spatial memory in partially observed environments (Chaplot et al., 2020a; Wani et al., 2020). Subsequent studies have expanded the range of tasks and goal specifications. For example, TEACh (Padmakumar et al., 2022) investigates multi-step household tasks guided by human instructions; NaviLLM (Zheng et al., 2024) models diverse navigation tasks through a unified language interface; and GOAT-Bench (Khanna et al., 2024) evaluates long-horizon navigation with sequential category, language, and image goals. Other work studies context-dependent sequential grounding (Zhang et al., 2024b), combines visual grounding with active exploration (Zhu et al., 2025), maintains spatial representations over extended execution (Yang et al., 2025; Fan et al., 2025), or investigates asynchronous interaction, task prioritization, and runtime preemption (Wang et al., 2025; Zhou et al., 2025; Chen et al., 2026). These studies advance different capabilities relevant to long-horizon navigation, interaction, and task scheduling. However, to the best of our knowledge, existing embodied navigation benchmarks do not systematically evaluate the complete process in which an agent receives a higher-priority instruction while executing a base task sequence, completes its current atomic task before switching to the inserted task, and resumes the base sequence from the updated environmental state after completing the inserted task. Aggregate navigation success alone also cannot distinguish performance in handling the inserted task, resuming the original task, and completing the full sequence. A dedicated benchmark is therefore needed to evaluate runtime task preemption and resumption in a controlled, stage-specific manner.

We study *runtime task preemption and resumption*. Let the base task sequence be $A = (E_{A_1}, E_{A_2}, \ldots, E_{A_n})$. If the agent receives a higher-priority instruction while executing an atomic task, it first completes that task, then suspends the remaining tasks in A and executes the inserted task. Once the inserted task is complete, the agent resumes A at the next unexecuted atomic task. Here, **resumption** refers to restoring the logical progress of the base sequence and re-grounding its next objective from the current environmental state; it does not require returning to the pre-interruption position or reusing the original route. Figure 1 illustrates this process. The agent is ini-

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.

tially assigned the base sequence $A = (E_{A_1}, E_{A_2}, E_{A_3})$: retrieve food from a refrigerator, retrieve a book, and complete a subsequent task. While executing E_{A_1}, the agent receives a higher-priority inserted task E_B, which requires organizing the cabinets. To ensure that the current atomic task is completed, the agent finishes E_{A_1} before suspending the unexecuted tasks E_{A_2} and E_{A_3} and switching to E_B. After completing E_B, the agent resumes A at E_{A_2} and then proceeds to E_{A_3}. Because the inserted task may change the agent's location, viewpoint, and available navigation candidates, resuming E_{A_2} requires the agent to re-ground its target and plan from the current state, rather than return to its pre-interruption position or follow the original route.

To address this gap, we introduce **PIFR**, a benchmark for evaluating runtime task preemption and resumption in embodied navigation. PIFR examines whether an agent can handle an inserted task and subsequently resume its suspended task sequence under diverse goal specifications and interruption structures, including *single*, *repeated*, and *nested* interruptions. This design enables evaluation beyond conventional task completion by distinguishing performance during task interruption and subsequent resumption.

Evaluations on PIFR show that recovery performance varies with interruption conditions and goal specifications. They also reveal that aggregate navigation success can mask differences in an agent's ability to handle inserted tasks and resume suspended ones. To study this challenge, we further propose **PIFR-Rank**, a lightweight candidate-ranking baseline aimed at improving target re-grounding after an interruption. Together, PIFR and PIFR-Rank provide an evaluation benchmark and a reference method for investigating task recovery in long-horizon embodied navigation.



Figure 1: **Interruption-and-resumption navigation in PIFR.** During long-horizon navigation, an agent may need to suspend its current task, handle an inserted objective, and then return to the unfinished task. Blue, red, and green trajectories indicate base, interrupt, and resume execution, respectively.

Our contributions are summarized as follows:

- We introduce **PIFR**, a benchmark for runtime task preemption and resumption in embodied navigation. The curated datasets cover diverse goal specifications and single, repeated, and nested interruption structures.
- We develop a continuous-execution protocol and stage-specific evaluation measures to assess inserted-task completion, resumption of the base task, and completion of the full task sequence, enabling systematic analysis of navigation performance under different interruption conditions.
- We propose **PIFR-Rank**, a lightweight candidate-ranking baseline that improves target re-grounding after interruption through candidate-level supervision while preserving the original navigation pipeline.

2 Related Work

2.1 Embodied Navigation and Long-Horizon Task Execution

Early embodied navigation research develops methods and benchmarks for distinct task formulations, including vision-and-language navigation (Anderson et al., 2018), ObjectNav (Chaplot et al., 2020b), and ImageNav (Zhu et al., 2017). These studies establish foundations for embodied perception and navigation while introducing different goal types and task protocols. Subsequent work explores whether a unified agent can handle a broader range of navigation tasks. Towards Versatile Embodied Navigation (Wang et al., 2022) learns a unified policy across multiple tasks, while

NaviLLM (Zheng et al., 2024) models diverse embodied tasks through a shared language interface. MTU3D (Zhu et al., 2025) combines visual grounding, active exploration, and spatial memory to support category-, language-, and image-conditioned navigation. OctoNav (Gao et al., 2026) further extends generalist navigation to free-form instructions and diverse goal modalities. These studies primarily broaden the types of tasks and goals that an agent can handle.

A complementary line of work investigates multi-goal navigation and long-horizon task execution. MultiON (Wani et al., 2020) studies whether an agent can visit multiple targets sequentially within a single episode. GOAT-Bench (Khanna et al., 2024) evaluates long-term navigation with goals specified as categories, language, or images. Long-Horizon Vision-Language Navigation (Song et al., 2025) considers multi-stage instructions and consecutive subtask execution, while Lifelong Embodied Navigation Learning (Wang et al., 2026) studies the acquisition of new navigation capabilities from a continual-learning perspective. Overall, these works extend the scale, duration, or goal diversity of navigation tasks, but primarily evaluate the execution of predefined tasks or task sequences. They do not systematically examine a different execution scenario in which a new task is inserted before the original sequence is complete and the agent must resume that sequence from the environment state reached after executing the inserted task.

2.2 Interruptions in Embodied Task Execution

Embodied agents often receive new instructions during ongoing interactions. HANNA (Nguyen & Daumé III, 2019) enables navigation agents to request multimodal assistance from a simulated human, while Vision-and-Dialog Navigation (Thomason et al., 2020) and TEACh (Padmakumar et al., 2022) study instruction-driven interaction and multi-step household task execution. These works examine how agents use additional instructions or dialogue to guide ongoing behavior, but do not evaluate the recovery of an unfinished navigation task after the agent executes an intervening task. Recent proposed InterruptBench (Zou et al., 2026) explicitly studies user interruptions during web-agent execution, including changes to ongoing requests. Its setting concerns modifying the active user request. In this work, however, PIFR instead considers embodied navigation, where an inserted task is executed while the original task sequence remains unfinished and must be resumed from the resulting environmental state. This setting requires evaluating both the execution of the inserted task and the recovery of the suspended navigation task. To the best of our knowledge, existing embodied navigation benchmarks do not provide a systematic evaluation of this process.

3 PIFR Benchmark

We introduce **PIFR**, a novel task for evaluating interruption and resumption in persistent embodied navigation. Unlike conventional sequential navigation, in which an agent proceeds through a predefined sequence of objectives, PIFR allows an unfinished task sequence to be suspended by an inserted task and resumed after that task is completed. We first formally define the PIFR task (sec. 3.1) and present three representative instantiations (sec. 3.2). We then describe how the datasets are constructed under this formulation and provide the corresponding implementation details (sec. 3.3).

3.1 Task Definition

Let $\mathcal{A}$ denote the set of atomic navigation tasks. Each task $x \in \mathcal{A}$ specifies a navigation goal and its corresponding instruction. A base task sequence is a finite ordered sequence

$$X = (x_1, x_2, \ldots, x_n), \qquad x_i \in \mathcal{A}.$$

Let $\mathcal{T}_0 = \mathcal{A}^+$ denote the set of all nonempty finite sequences of atomic tasks. We use $X[i{:}j]$ to denote the contiguous subsequence of X from task i to task j.

We recursively define composite task sequences to represent interruptions. Let $\mathcal{K}(X) \subseteq \{1, \ldots, n-1\}$ denote the set of valid interruption boundaries in X. An interruption at boundary k occurs after x_k is completed and before x_{k+1} begins. Given two task sequences or composite task sequences X and Y, embedding Y into X at boundary $k \in \mathcal{K}(X)$ is denoted by $X \oplus_k Y$ and defined as

$$X \oplus_k Y := X[1{:}k] \to Y \to X[k+1{:}n],$$

where $\to$ denotes execution order. The embedded sequence Y may itself be a composite sequence, allowing interruptions to be recursively nested.

The set of composite tasks is defined recursively as

$$\mathcal{T}_{d+1} = \mathcal{T}_d \cup \{X \oplus_k Y \mid X, Y \in \mathcal{T}_d,\ k \in \mathcal{K}(X)\}, \qquad d \geq 0,$$

where $\mathcal{T}_d$ contains task sequences with nesting depth at most d. The set of all finite composite task sequences is

$$\mathcal{T}_{\text{comp}} = \bigcup_{d=0}^{\infty} \mathcal{T}_d.$$

Each PIFR event is an instance in $\mathcal{T}_{\text{comp}}$, subject to the interruption and nesting constraints specified by the benchmark.

During a composite event, the simulator state is preserved across interruptions and resumptions. Once an inserted sequence is completed, the suspended sequence resumes at its next unfinished task from the resulting environment state. Thus, resumption preserves task progress but does not restore the agent's pre-interruption position or trajectory. The instruction for the next unfinished task is re-issued upon resumption, requiring the agent to ground its goal in the updated physical and perceptual context.

3.2 Canonical Instantiations

Let $A = (a_1, \ldots, a_m)$ be a base sequence, and let B and C be inserted sequences. The following three PIFR settings are instances of the recursive embedding definition.

❶ **Single interruption.** A single inserted sequence B is embedded at boundary k_A of A:

$$A \oplus_{k_A} B = A[1{:}k_A] \to B \to A[k_A + 1{:}m].$$

The agent executes the prefix of A, then B, and finally resumes A at its next unfinished task.

❷ **Repeated interruption.** Two inserted sequences, B and C, are embedded successively at distinct boundaries $k_1 < k_2$ of the base sequence A:

$$\big(A \oplus_{k_1} B\big) \oplus_{k_2} C = A[1{:}k_1] \to B \to A[k_1 + 1{:}k_2] \to C \to A[k_2 + 1{:}m].$$

Here, k_1 and k_2 are indexed with respect to the original sequence A. After completing B, the agent resumes A and continues to the second interruption boundary. It then executes C before resuming the remaining suffix of A.

❸ **Nested interruption.** Sequence B is embedded in A at boundary k_A, and sequence C is embedded in B at boundary k_B:

$$A \oplus_{k_A} \big(B \oplus_{k_B} C\big) = A[1{:}k_A] \to B[1{:}k_B] \to C \to B[k_B + 1{:}n] \to A[k_A + 1{:}m],$$

where $B = (b_1, \ldots, b_n)$. The agent first executes the prefix of A and then the prefix of B. Sequence B is subsequently interrupted by C. After completing C, the agent resumes and completes B, and then resumes A. This defines the last-in, first-out resumption order induced by recursive embedding.

As shown in Figure 2, The three settings differ in their embedding structure: **Single** embeds one episode at one boundary; **Repeated** embeds episodes at multiple boundaries of the same base episode; and **Nested** recursively embeds an interruption within an inserted episode. Any limits on the number of interruptions or nesting depth are specified by the benchmark construction protocol.

3.3 Dataset Curation

We construct three PIFR dataset variants from SG3D (Zhang et al., 2024b), GOAT-Bench (Khanna et al., 2024), and HM3D-OVON (Yokoyama et al., 2024): **PIFR-SG3D**, **PIFR-GOAT**, and **PIFR-OVON**, respectively. Each variant inherits the goal specifications of its source benchmark. Specifically, PIFR-SG3D supports description-based instance grounding, PIFR-GOAT includes category, language, and image goals, and PIFR-OVON supports open-vocabulary object-category navigation. We preserve these goal specifications and compose the source navigation tasks into the composite task episodes defined above, enabling the evaluation of task resumption across diverse goal settings.



Figure 2: **The overview of the composition of PIFR and the Single, Repeated and Nested interruption.** Among the three different types of interrupts—Single, Repeated, and Nested—the orange arrow represents an interrupt, and the blue arrow represents a resume.

Data Splits and Statistics. We retain the original data splits of each source benchmark. PIFR-SG3D uses the `train` and `val` splits, whereas PIFR-GOAT and PIFR-OVON use the `train`, `val_seen`, `val_seen_synonyms`, and `val_unseen` splits. As reported in Table 1, PIFR contains 80,417 composite episodes, comprising 74,223 training episodes and 6,194 validation episodes. Specifically, PIFR-SG3D contains 16,019 training and 466 validation episodes; PIFR-GOAT contains 28,148 training and 731 validation episodes; and PIFR-OVON contains 30,056 training and 4,997 validation episodes.

The training set contains 59,101 Single episodes (79.63%), 7,558 Repeated episodes (10.18%), and 7,564 Nested episodes (10.19%), yielding an approximate distribution of 80%/10%/10%. Each composite episodes comprises multiple atomic navigation tasks. Across the full benchmark, an episodes contains 9.78 atomic navigation segments on average, with a median of 8. The mean episodes lengths are 9.23, 11.62, and 12.30 segments for the Single, Repeated, and Nested settings, respectively.

Episode Construction and Validation. For PIFR-SG3D and PIFR-GOAT, we select base episodes from source episodes containing at least four atomic navigation tasks, thereby providing multiple valid embedding boundaries. Since HM3D-OVON consists of single-goal episodes, we deterministically combine four episodes from the same scene, each with a different target category, to form a multi-task base episode. Inserted episodes are sampled from compatible episodes in the same scene, and their first goals are required to differ from those in the parent episode.

We construct composite events by embedding inserted episodes at valid atomic-task boundaries according to the previously defined $\oplus_k$ operation. The Single setting samples one embedding boundary; the Repeated setting samples two distinct boundaries in the base episode; and the Nested setting additionally requires a valid boundary within an inserted episode, allowing another episode to be embedded recursively. Because individually feasible source episodes may become infeasible when combined, we verify geometric navigability along the complete execution chain. We also use deterministic sampling and balance events across interruption structures and scenes to support reproducibility and dataset diversity.

Table 1: PIFR dataset. **Seen** means categories and instances are in the training data. **Syn.** denotes unseen descriptions use synonyms of goals observed during training. **Unseen** indicates categories are not in the training data.

Variant	Train	Test			Total
		Seen	Syn.	Unseen	
PIFR-SG3D	16,019	466	–	–	16,485
PIFR-GOAT	28,148	239	248	244	28,879
PIFR-OVON	30,056	1,631	1,623	1,743	35,053
Total	**74,223**		**6,194**		**80,417**

Table 2: Comparison with representative embodied navigation benchmarks. **Pers.**, **Int.**, and **Res.** denote persistent execution, interruption, and resumption.

Benchmark	Description	Pers.	Int.	Res.
HM3D-OVON	A	×	×	×
MultiON	$A \to B \to C$	✓	×	×
GOAT-Bench	$A \to B \to C$	✓	×	×
LHPR-VLN	$A \to B \to C$	✓	×	×
PIFR	**Single / Repeated / Nested**	✓	✓	✓

Comparison with Existing Benchmarks. Table 2 compares PIFR with representative embodied navigation benchmarks in terms of temporal task organization. Existing benchmarks cover single-goal navigation, ordered multi-goal tasks, and long-horizon execution, but generally require agents

to progress through objectives in a predefined order. PIFR additionally supports interruption and resumption of task sequences through recursive embedding. Its Nested setting further evaluates recovery when an inserted sequence is itself interrupted.

4 PIFR-RANK

Overview. When an inserted task sequence is completed, the agent resumes the most recently suspended sequence from the resulting environment state and receives its next unfinished atomic-task instruction. Since the intervening execution may change the agent's location, observations, and spatial context, the goal must be re-grounded among the currently available candidates. To improve this process, we propose **PIFR-Rank**, which trains the grounding scorer with candidate-level supervision and a ranking loss, while assigning greater weight to decisions made upon resumption. Figure 3 illustrates the candidate-generation pipeline and the proposed ranking objective.

4.1 CANDIDATE CONSTRUCTION AND TRAIN

At grounding decision i, the system constructs a candidate set $\mathcal{C}_i = \{c_{ij}\}_{j=1}^{N_i}$ from the current observation and navigation context. It includes object candidates from observed regions and frontier candidates associated with unexplored regions. Given the active instruction q_i, the grounding scorer assigns each candidate a score

$$s_{ij} = f_\theta(c_{ij}, q_i),$$

where f_θ denotes the scorer optimized during training.

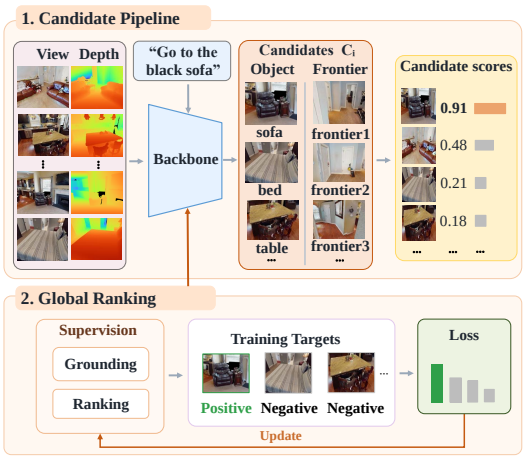

Figure 3: Overview of PIFR-Rank. The upper shows object and frontier candidate generation from visual observations and the active instruction. The lower illustrates the ranking objective, which encourages target-consistent candidates to score above distractors.

Each candidate receives a binary label $y_{ij} \in \{0, 1\}$ indicating whether it is valid for the active goal. When the target has been observed, matching object candidates are labeled positive; if multiple candidates match, all receive positive labels. When the target has not been observed, training-only target and scene information is used to label a suitable frontier candidate from the existing candidate set. These labels do not add candidates or affect candidate generation or selection at inference time.

Let $m_{ij} \in \{0, 1\}$ indicate whether an entry is a valid candidate ($m_{ij} = 1$) or padding ($m_{ij} = 0$). For each decision with at least one valid candidate and one positive candidate, the padding-aware binary cross-entropy loss is

$$\ell_i^{\text{BCE}} = \frac{\sum_j m_{ij}\ \text{BCEWithLogits}(s_{ij}, y_{ij})}{\sum_j m_{ij}}. \tag{1}$$

This loss supports multiple positive candidates and excludes padding entries.

To encourage positive candidates to outrank distractors, we define

$$s_i^+ = \max_{j:\, m_{ij}=1,\, y_{ij}=1} s_{ij}, \qquad s_i^- = \max_{j:\, m_{ij}=1,\, y_{ij}=0} s_{ij}. \tag{2}$$

When both positive and negative candidates are present, the ranking loss is

$$\ell_i^{\text{rank}} = \max\left(0,\ \delta - s_i^+ + s_i^-\right), \tag{3}$$

where $\delta > 0$ is the margin. It requires the highest-scoring positive to exceed the highest-scoring negative by at least δ; it does not constrain every positive-negative pair. The ranking loss is zero when no valid negative candidate is available. Decisions without a valid positive candidate are excluded from both candidate-level losses.

A *resumption decision* is the grounding decision for the next unfinished atomic task when a suspended sequence resumes after an inserted sequence completes. This definition applies to each

resumption in Repeated and Nested events, including the resumption of an inner sequence and of the outer base sequence. Let r_i denote the execution role of decision i, and let $w(r_i)$ assign greater weight to resumption decisions. For the set $\mathcal{I}$ of decisions with at least one valid candidate and one positive candidate, we optimize

$$\mathcal{L}_{\mathrm{GR}} = \frac{\sum_{i\in\mathcal{I}} w(r_i)\left(\ell_i^{\mathrm{BCE}} + \ell_i^{\mathrm{rank}}\right)}{\sum_{i\in\mathcal{I}} w(r_i)}. \tag{4}$$

Thus, decisions with positive candidates but no negative candidates still contribute their BCE loss. Execution-role labels affect only training weights and are not inputs to the online policy.

At inference time, the agent uses the active instruction, visual observations, navigation context, and candidates generated by the navigation system. Candidate labels, execution-role annotations, future tasks, and training-only reference information are unavailable. PIFR-Rank changes the training objective while retaining the original candidate-generation and online navigation procedures.

5 EXPERIMENTS

Implementation Details. PIFR-Rank is fine-tuned independently on PIFR-SG3D, PIFR-GOAT, and PIFR-OVON. We optimize the model for 5 epochs using AdamW with a learning rate of 1×10^{-4}, $\beta = (0.9, 0.98)$, and weight decay of 0.01. The GlobalRank objective uses a ranking margin $\delta = 0.5$, and a Resume sample weight of 3. All models are trained on NVIDIA RTX 4090 GPUs with a batch size of 8.

Metrics. Following standard embodied navigation evaluation (Khanna et al., 2024), we report **Success Rate (SR)** and **Success weighted by Path Length (SPL)** to measure navigation success and path efficiency. An atomic navigation segment is considered successful if the agent terminates within 0.25m geodesic distance of a valid target viewpoint. SR and SPL are computed over completed atomic navigation segments. To evaluate interruption-and-resumption behavior, we additionally report **Interrupt SR** and **Resume SR**. For each segment q, let d_q and d_q^{end} denote the shortest path length and the geodesic distances to the active target at its end, respectively, and let ℓ_q be the agent path length, $\mathcal{Q}$ denote the set of valid atomic navigation segments. A segment is successful if $S_q = [d_q^{\mathrm{end}} \le \tau]$, where $\tau = 0.25\,\mathrm{m}$. The four metrics are shown below:

- **Overall navigation metrics.** Success Rate (**SR**) measures the proportion of successful atomic navigation segments, while Success weighted by Path Length (**SPL**) accounts for path efficiency:

$$\textbf{SR} := \frac{100}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} S_q, \qquad \textbf{SPL} := \frac{100}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} S_q \frac{d_q}{\max(d_q, \ell_q, \epsilon)}. \tag{5}$$

- **Stage-specific metrics.** Interruption SR (**Int.**) and Resumption SR (**Res.**) measure the success rates of interruption-entry and resumption-entry segments, respectively. Let $\mathcal{T}_{\mathrm{int}}$ and $\mathcal{T}_{\mathrm{res}}$ denote the corresponding sets of valid segments:

$$\textbf{Int.} := \frac{100}{|\mathcal{T}_{\mathrm{int}}|}\sum_{q\in\mathcal{T}_{\mathrm{int}}} S_q, \qquad \textbf{Res.} := \frac{100}{|\mathcal{T}_{\mathrm{res}}|}\sum_{q\in\mathcal{T}_{\mathrm{res}}} S_q. \tag{6}$$

5.1 BENCHMARKING RESULTS WITH NAVIGATION BASELINES

We evaluate R2F (Argenziano et al., 2026), E-VA (Fan et al., 2025), Uni-NaVid (Zhang et al., 2024a), MTU3D (Zhu et al., 2025) , and PIFR-Rank on PIFR-SG3D, PIFR-GOAT, and PIFR-OVON under the Single, Repeated, and Nested interruption settings.

Main Results on Seen Goals Table 3 compares representative navigation methods on the Seen splits of PIFR-SG3D, PIFR-OVON, and PIFR-GOAT under single, repeated, and nested interruption. The results show that method performance depends on both the source benchmark and the interruption structure. PIFR-Rank achieves higher SR than MTU3D across all three structures on PIFR-SG3D and PIFR-OVON. On PIFR-GOAT, MTU3D performs better under single interruption, whereas PIFR-Rank achieves higher SR under repeated and nested interruption. In particular, under nested interruption, PIFR-Rank improves SR from 44.3% to 49.6%.

Table 3: Results on PIFR-SG3D / PIFR-OVON (Seen) / PIFR-GOAT (Seen) under different interruption settings (**❶ Single Interruption, ❷ Repeated Interruption, ❸ Nested Interruption**).

Method	❶ Single Interruption				❷ Repeated Interruption				❸ Nested Interruption			
	SR↑	SPL↑	Int.↑	Res.↑	SR↑	SPL↑	Int.↑	Res.↑	SR↑	SPL↑	Int.↑	Res.↑
PIFR-SG3D												
R2F	4.8	2.8	3.2	3.0	3.7	1.9	3.5	5.0	3.3	2.3	3.1	2.8
E-VA	4.7	3.5	3.2	6.3	4.9	3.4	2.5	10.0	6.5	4.7	5.7	10.4
Uni-NaVid	10.7	7.1	9.0	9.0	7.0	4.6	7.4	7.5	6.0	3.6	4.6	3.8
MTU3D	22.6	12.8	22.0	23.0	15.8	**11.4**	11.3	20.0	15.6	9.2	**11.3**	**15.1**
Ours	**26.7**	**15.5**	**22.5**	**31.4**	**17.4**	11.1	**12.5**	**25.0**	**18.9**	**11.3**	**11.3**	13.2
PIFR-OVON												
R2F	24.3	10.4	24.4	25.5	22.4	10.2	23.5	19.1	23.2	10.2	19.3	22.3
E-VA	20.4	10.2	22.6	13.7	19.2	10.3	20.5	17.6	21.6	11.4	22.0	21.5
Uni-NaVid	34.7	13.0	37.8	31.6	34.6	12.9	29.9	32.4	33.2	12.1	31.7	32.2
MTU3D	49.6	23.7	44.9	**56.1**	45.9	22.0	40.7	45.6	46.0	23.2	41.1	45.1
Ours	**51.7**	**26.5**	**49.0**	49.0	**47.5**	**23.9**	**47.2**	**46.2**	**47.3**	**24.9**	**43.8**	**50.0**
PIFR-GOAT												
R2F	17.5	6.6	15.6	15.8	15.4	7.5	13.5	14.0	18.8	8.6	15.8	24.0
E-VA	15.4	11.3	15.5	17.5	16.9	11.1	18.0	20.0	16.7	11.6	8.0	24.0
Uni-NaVid	23.8	12.2	19.5	24.2	19.6	10.3	16.5	19.0	22.6	11.1	15.4	26.0
MTU3D	**52.8**	**32.6**	**47.4**	**61.1**	42.9	28.1	36.0	54.0	44.3	29.9	34.0	48.0
Ours	50.0	31.7	44.9	58.2	**43.8**	**29.7**	**38.0**	**55.0**	**49.6**	**34.8**	**48.0**	**60.0**

The stage-specific metrics provide additional information beyond overall navigation success. For example, on PIFR-OVON under single interruption, PIFR-Rank obtains a higher SR than MTU3D (51.7% vs. 49.6%), but a lower Resume SR (49.0% vs. 56.1%). This difference shows that overall task success and recovery performance do not always change together, motivating the separate analysis of goal-split generalization and multi-round recovery below.

Results on Synonym and Unseen Goals Figure 4 presents SR, SPL, Interrupt SR, and Resume SR on the Synonym and Unseen splits of PIFR-OVON and PIFR-GOAT. The results show that performance changes are not uniform across datasets or interruption structures. On PIFR-OVON, PIFR-Rank achieves higher SR than MTU3D under all three interruption structures on the Unseen split, whereas its advantage on the Synonym split is small and varies by structure. On PIFR-GOAT, PIFR-Rank outperforms MTU3D on the Synonym split and under single and repeated interruption on the Unseen split, but not under nested interruption. These results indicate that robustness to synonym variation and generalization to unseen goals present different challenges, and that their effects depend on the temporal structure of the episode. The four metrics also reveal that changes in overall success do not necessarily imply corresponding changes in interruption handling or resumption. Since the source benchmarks define their Synonym and Unseen splits independently, comparisons should be made within each benchmark and split protocol.

Stage-wise Recovery under Multiple Interruptions Figure 5 decomposes Resume SR by the order of recovery. For repeated interruption (ABACA), it reports the first and second resumptions

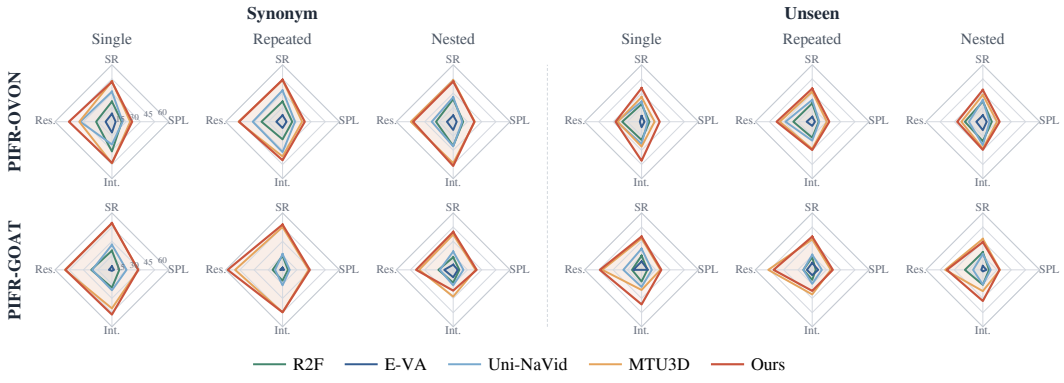

Figure 4: Radar plots of SR, SPL, Interrupt SR, and Resume SR on the Synonym and Unseen splits of PIFR-OVON and PIFR-GOAT under single, repeated, and nested interruption.

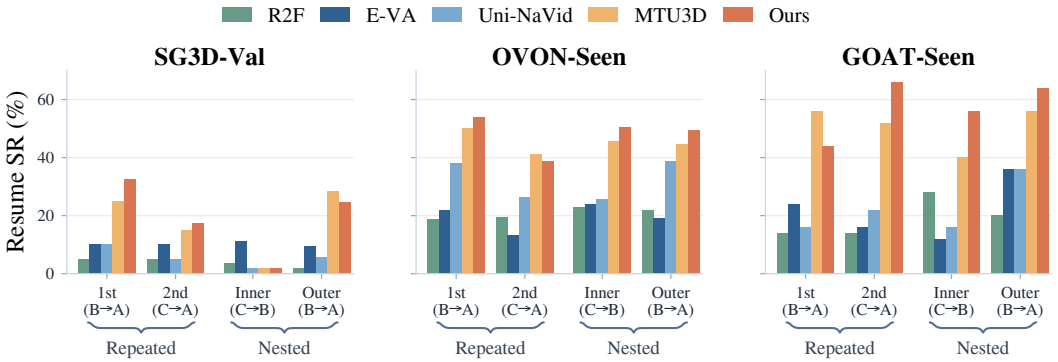

Figure 5: **Repeated and nested execution reveal distinct stage-wise recovery profiles.** Resume SR is decomposed into the first and second resumptions for ABACA ($B \to A$ and $C \to A$), and the inner and outer resumptions for ABCBA ($C \to B$ and $B \to A$). Recovery behavior differs substantially across navigation settings, showing that a single ABA recovery score cannot fully characterize repeated and nested task resumption.

($B \to A$ and $C \to A$). For nested interruption (ABCBA), it reports the inner and outer resumptions ($C \to B$ and $B \to A$). PIFR-Rank exhibits different stage-wise profiles across the three benchmarks. Under repeated interruption, its Resume SR decreases from 32.5% to 17.5% on PIFR-SG3D and from 53.8% to 38.7% on PIFR-OVON, but increases from 44.0% to 66.0% on PIFR-GOAT. Under nested interruption, its inner and outer Resume SRs are 1.9% and 24.5% on PIFR-SG3D, 50.5% and 49.5% on PIFR-OVON, and 56.0% and 64.0% on PIFR-GOAT. Thus, later or nested recovery does not produce a uniform monotonic decline. Instead, recovery performance varies with both the source benchmark and the position of the resumption within the interruption sequence.

5.2 Ablation Study of PIFR-Rank

We conduct ablations on the Seen validation splits of PIFR-SG3D, PIFR-OVON, and PIFR-GOAT. To give each dataset and interruption structure equal weight, we report an equal-cell macro average over the nine dataset-by-structure combinations. Within each cell, SR and SPL are averaged over atomic navigation segments, whereas Interrupt SR and Resume SR are computed over segments assigned to the corresponding execution stages.

Table 4: Ablation study of PIFR-Rank. **BCE.**, **CR.** and **RW.** denote BCE Loss, candidate ranking and resumption weighting, respectively. The best result for each metric is highlighted in bold.

Method	SR↑	SPL↑	Int. SR↑	Res. SR↑
Baseline	37.03	21.55	31.34	40.82
Baseline + **BCE.**	**39.82**	23.32	33.46	44.91
Baseline + **BCE.** + **CR.**	39.28	**23.54**	34.36	43.10
Baseline + **BCE.** + **CR.** + **RW.**	39.21	23.25	**35.25**	**45.04**

As shown in Table 4, adding candidate-level BCE supervision to the baseline improves all four metrics: SR increases from 37.03% to 39.82%, SPL from 21.55% to 23.32%, Interrupt SR from 31.34% to 33.46%, and Resume SR from 40.82% to 44.91%. Adding candidate ranking (CR) further improves SPL and Interrupt SR to 23.54% and 34.36%, respectively, while SR and Interrupt SR decrease slightly. Adding resumption weighting (RW) yields the highest Interrupt SR and Resume SR, at 35.25% and 45.04%, respectively; SR and SPL are 39.21% and 23.25%, below the BCE-only variant. Thus, BCE provides the broadest improvement over the baseline, while CR and RW shift the metric profile toward stronger stage-specific performance, especially during resumption.

6 Conclusion

We present PIFR, a benchmark for evaluating task-sequence interruption and resumption in embodied navigation. By introducing single, repeated, and nested interruptions, PIFR assesses both navigation success and recovery of unfinished tasks from the post-interruption state. Experiments show that aggregate success alone can obscure differences in interruption handling and resumption, whose performance varies across execution structures. We also introduce PIFR-Rank, a candidate-ranking baseline that improves resumption-related metrics, though gains vary across settings. PIFR provides a testbed for studying task recovery in long-horizon embodied navigation.

AI USE STATEMENT

In this work, the AI tools are employed solely to improve the overall presentation of this manuscript. They were not involved in defining the research objectives, developing the methodology, establishing technical formulations, designing experiments. All contributions and results are reviewed by the authors. We take full responsibility for the manuscript.

ETHICS STATEMENT

Our work introduces PIFR, a simulation-based benchmark for persistent interruption and resumption in embodied navigation. The benchmark construction and evaluation procedures are intended for research use, subject to the applicable licenses and usage conditions of the source datasets.

The benchmark may inherit biases in scene coverage, object categories, language annotations, and task distributions from its source datasets. Although we introduce multiple interruption structures and goal-space evaluation splits, these settings do not capture the full complexity of real-world environments. In particular, successful performance in simulation should not be interpreted as evidence of reliable deployment in physical robots. We encourage future work to examine broader environmental diversity, fairness, and safety before applying the proposed benchmark in real-world settings.

REPRODUCIBILITY STATEMENT

We provide details to support reproduction of the benchmark , methods, and experiments. Specifically, sec. 3 describes the sequence construction process, reachability validation, and interruption patterns. sec. 4 specifies the candidate labeling procedure and training objective of PIFR-Rank. sec. 5 reports the training settings, evaluation protocol, metrics, and results for the evaluated methods. And all methods are evaluated under the same PIFR protocol and metrics.

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 *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 3674–3683. IEEE, 2018.

Francesco Argenziano, John Mark Alexis Marcelo, Michele Brienza, Abdel Hakim Drid, Emanuele Musumeci, Daniele Nardi, Domenico D Bloisi, and Vincenzo Suriani. R2f: Repurposing ray frontiers for llm-free object navigation. *arXiv preprint arXiv:2603.08475*, 2026.

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

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

Guojun Chen, Alex Schott, and Lin Zhong. Typego: An os runtime for embodied agents. *arXiv preprint arXiv:2607.05482*, 2026.

Yue Fan, Xiaojian Ma, Rongpeng Su, Jun Guo, Rujie Wu, Xi Chen, and Qing Li. Embodied videoagent: Persistent memory from egocentric videos and embodied sensors enables dynamic scene understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6342–6352. IEEE, 2025.

Chen Gao, Liankai Jin, Xingyu Peng, Jiazhao Zhang, Yue Deng, Annan Li, He Wang, and Si Liu. Octonav: Towards generalist embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40074–40084, 2026.

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

Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 684–695, 2019.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2017–2025, 2022.

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

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

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pp. 394–406. PMLR, 2020.

Haisheng Wang, Dong Liu, and Weiming Zhi. Coneqsa: Concurrent and asynchronous embodied questions scheduling and answering. *arXiv preprint arXiv:2509.11663*, 2025.

Hanqing Wang, Wei Liang, Luc V Gool, and Wenguan Wang. Towards versatile embodied navigation. *Advances in neural information processing systems*, 35:36858–36874, 2022.

Xudong Wang, Jiahua Dong, Baichen Liu, Qi Lyu, Lianqing Liu, and Zhi Han. Lifelong embodied navigation learning. *arXiv preprint arXiv:2603.06073*, 2026.

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

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3d-mem: 3d scene memory for embodied exploration and reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17294–17303. IEEE, 2025.

Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5543–5550. IEEE, 2024.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024a.

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

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13624–13634. IEEE, 2024.

Shengli Zhou, Xiangchen Wang, Jinrui Zhang, Ruozai Tian, Rongtao Xu, and Feng Zheng. p3: Toward versatile embodied agents. *arXiv preprint arXiv:2508.07033*, 2025.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3357–3364. ieee, 2017.

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

Henry Peng Zou, Chunyu Miao, Wei-Chieh Huang, Yankai Chen, Yue Zhou, Hanrong Zhang, Yaozu Wu, Liancheng Fang, Zhengyao Gu, Zhen Zhang, et al. When users change their mind: Evaluating interruptible agents in long-horizon web navigation. *arXiv preprint arXiv:2604.00892*, 2026.