



GG-Nav: Goal-oriented Grounding Chain-of-Thought Elicits Reasoning in Navigation Foundation Models

Shaohao Zhu^{1,2}, Wensi Huang^{2,3}, Siqi Zhang^{2,4}, Yuqiang Yang², Meng Wei^{2,5}, Chenyang Wan^{1,2}, Tianle Liu¹, Tai Wang^{2,*}, Jinming Xu^{1,*}

¹Zhejiang University, ²Shanghai AI Laboratory, ³University of Science and Technology of China,

⁴Tongji University, ⁵The University of Hong Kong

*Corresponding author, †Project lead

Emerging strong Vision-Language Models (VLMs) have advanced navigation foundation models. However, these models still have unsatisfactory performance on goal-oriented navigation settings. In this paper, we first conduct a motivating study revealing that the key issue lies in the insufficient utilization of streaming observations and imprecise target localization. To tackle these challenges, we propose **GG-Nav**, a simple yet effective navigation framework that learns a structured chain-of-thought (CoT) mechanism via supervised fine-tuning to explicitly localize goal-relevant instances, accumulate grounding evidence, and support more reliable decision-making. Specifically, the structured CoT is designed with two key stages: (i) Observation-to-Evidence, which reasons over all intermediate frames between adjacent decision steps to retain useful goal-related evidence; and (ii) Evidence-to-Decision, which integrates the current observation with accumulated evidence to ground the target and support the generation of navigation actions or clarification questions. Extensive experiments show that GG-Nav outperforms the state-of-the-art methods in terms of success rates on the HM3D ObjectNav, VL-LN, and CoIN benchmarks by 3.1%, 16.6%, and 5.3%, respectively. Ablation studies further show that accumulating evidence from intermediate observations reduces evidence-missing failures by 11.6%, while explicit grounding reduces stopping errors by 49.4%. We also deploy GG-Nav on a Unitree Go2 robot, demonstrating its zero-shot generalization capability in unseen real-world environments.

Project page: <https://duangzhu.github.io/GG-Nav-Project-Page/>

Code: https://github.com/DuangZhu/GG_Nav

1 Introduction

Goal-oriented navigation is a fundamental capability for embodied agents operating in human environments [10, 33]. Given a simple natural-language instruction, such as *search for (the) chair*, a robot must explore an unseen area, interpret continuous egocentric observations, and make sequential decisions to reach the intended target. This problem includes object-goal navigation (ObjectNav) [2, 6, 27, 38], where reaching any instance of the target category is sufficient, and interactive instance-goal navigation (IIGN) [9, 24], where the agent must identify the intended target instance from an underspecified instruction by exploring the environment and interacting with humans. Compared with ObjectNav, IIGN is more practical but imposes stronger instance-level grounding requirements [9]: the agent must not only detect plausible candidate objects, but also identify the intended instance and maintain its identity consistently throughout navigation.

Recent advances in vision-language models (VLMs) have improved embodied navigation [27, 40, 41]. Despite this progress, our detailed motivating study on goal-oriented navigation in Sec. 3 reveals that current VLM-based methods still suffer from two major bottlenecks: they often miss target-related evidence in streaming observations and lack explicit target localization for precise stopping. To reduce decision latency, existing VLM-based methods typically select only sparse key observations for decision-making, rather than processing the full stream of visual frames [28, 29, 35, 40]. However, this sparse sampling strategy may overlook transient but crucial evidence of the target instance or distractors. Meanwhile, most VLM-based navigation methods directly map observations and language instructions to actions [31, 40, 41], leaving target grounding and

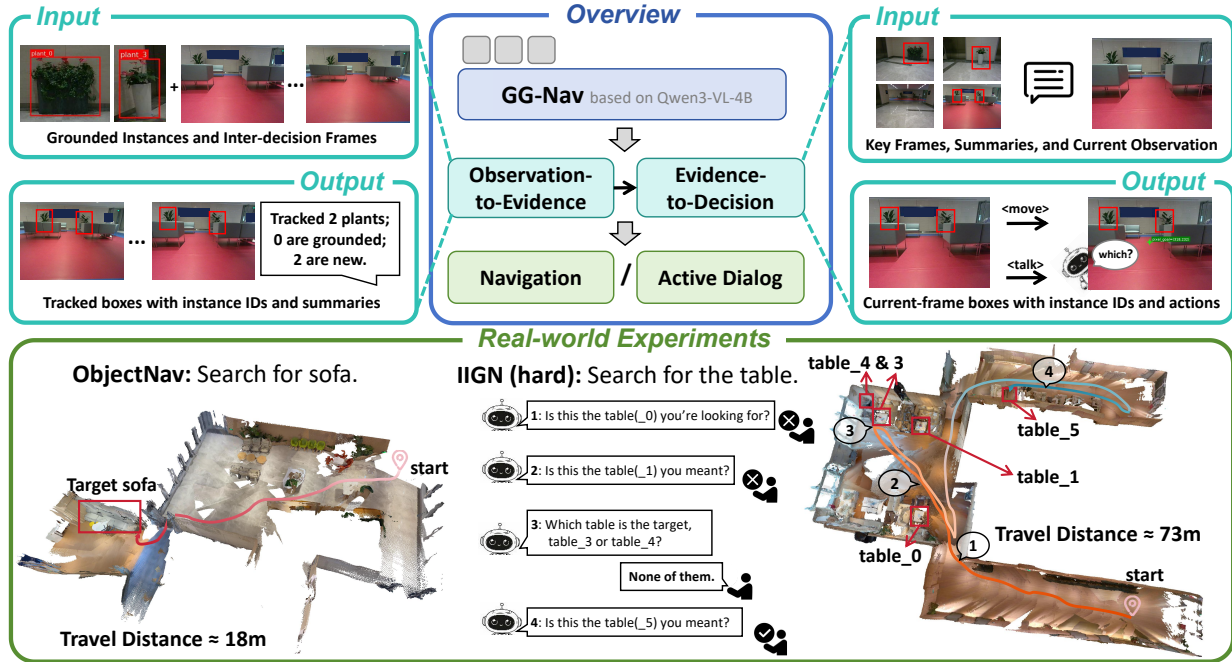


Figure 1 Overview of GG-Nav. Observation-to-Evidence converts intermediate frames between adjacent decision steps into accumulated grounding evidence. Evidence-to-Decision integrates the current observation with this evidence to ground goal-relevant instances and generate navigation actions or clarification questions. GG-Nav supports both ObjectNav and IIGN and supports real-world robot deployment.

localization implicit. As a result, the agent lacks an explicit estimate of where the target is, which makes it difficult to decide when to stop near the correct instance [2, 27, 44]. Although textual chain-of-thought reasoning has been explored [13, 27], free-form rationales are often weakly grounded in visual instances, limiting their ability to preserve instance-level evidence and support fine-grained supervision [13, 23].

To address these issues, we propose **GG-Nav**, a simple yet effective navigation framework that learns a **Goal-oriented Grounding CoT** mechanism via supervised fine-tuning to explicitly localize goal-relevant instances, accumulate grounding evidence, and support more reliable decision-making. As shown in Fig. 1, GG-Nav structures this process into two stages: Observation-to-Evidence and Evidence-to-Decision.

In each reasoning process, the Observation-to-Evidence stage first requires the model to accumulate key observations, retain useful key frames, ground goal-related evidence, and generate a summary over all intermediate frames collected between adjacent decision steps. Subsequently, the Evidence-to-Decision stage integrates the current observation with the accumulated evidence to explicitly ground goal-relevant instances and generate either a navigation action or a clarification question. To learn this structured reasoning mechanism, we construct a fine-tuning dataset with 30K grounding-annotated videos for evidence extraction and instance tracking, together with 60K navigation trajectories for evidence-conditioned decision-making to fine-tune the foundation VLM. In this way, GG-Nav is equipped with the capability of producing instance bounding boxes and consistent instance IDs, making its intermediate reasoning procedure spatially grounded and visually verifiable.

We evaluate GG-Nav on HM3D [18] for ObjectNav and on VL-LN [9] and CoIN [24] for IIGN. GG-Nav achieves the state-of-the-art performance across all three benchmarks, outperforming the strongest competing methods by 3.1%, 16.6%, and 5.3%, respectively. Progressive ablations show that incorporating intermediate observations into decision-making improves model performance by reducing failures caused by missing target-related evidence by 11.6%. In addition, explicitly generating bounding-box grounding results improves target localization and enables the agent to stop at more precise positions, reducing stopping errors by 49.4%. The structured grounding outputs further support stage-wise failure analysis, revealing that exploration and instance grounding remain the main bottlenecks. Finally, as shown in Fig. 1, we deploy GG-Nav on a Unitree

Go2 robot and demonstrate its zero-shot capability in unseen real-world environments under both ObjectNav and IIGN settings.

2 Related Work

2.1 Goal-oriented Navigation Benchmarks

Goal-oriented navigation requires an embodied agent to reach a user-specified target, serving as a foundational skill for real-world tasks such as retrieval, delivery, and inspection. This paper focuses on text-instructed goal-oriented navigation, where the target is specified in natural language. Early benchmarks mainly focused on object-goal navigation (ObjectNav), where the agent is instructed to find any instance of a target category, such as “find a chair” [2]. While this setting provides a useful benchmark for goal-oriented navigation, it does not fully cover everyday robot-assistance scenarios, where users often refer to a specific or personally relevant instance. More recent work further considers instance-goal navigation, where the target is a specific object instance rather than any category member [11]. However, these settings often assume that users can provide detailed and unambiguous target descriptions, which may not hold in daily human-robot interaction. To better match realistic human-robot interaction, interactive instance-goal navigation (IIGN) was introduced, where users may provide underspecified instructions (Search for the chair) and the agent must resolve the target ambiguity through interaction with a human during exploration [9, 24]. From this perspective, ObjectNav and IIGN represent two practical forms of goal-oriented navigation for everyday robot assistance. Despite their different goal specifications, both require the agent to accumulate goal-relevant evidence from continuous observations and ground it to candidate visual targets for reliable decision-making. This motivates our unified framework, which supports both settings through explicit evidence accumulation and grounding.

2.2 Goal-oriented Navigation Methods

Classical methods. Before the emergence of large foundation models, goal-oriented navigation methods mainly followed two paradigms: end-to-end and modular approaches. End-to-end methods directly map instructions and observations to navigation actions through reinforcement learning, imitation learning, or hybrid training strategies [16, 20, 21, 32]. Progress in ObjectNav was driven by auxiliary objectives [32], large-scale human demonstrations [20], data augmentation [16], and BC-to-RL training [21]. In parallel, modular methods decomposed navigation into semantic mapping, high-level goal selection, and local planning, enabling agents to navigate via structured semantic maps [7, 19].

Foundation-model-based methods. With the rise of VLMs and LLMs, recent work has incorporated foundation models into goal-oriented navigation along three main lines. The first line follows a modular paradigm, using foundation models as components for commonsense reasoning [12, 14, 34], open-vocabulary perception [36, 42, 43], scene or memory summarization [4, 17, 30, 33], and high-level waypoint selection [3, 33, 37, 39]. Although effective, these methods typically use foundation models in a zero-shot manner, resulting in weak alignment with embodied navigation, and their modular pipelines are prone to error accumulation in complex scenes [17, 36, 41, 42].

The second line directly adapts foundation models to goal-oriented navigation via supervised fine-tuning on embodied trajectories, learning end-to-end policies from observations and instructions to actions or exploration decisions. Representative methods include Uni-NaVid [40], Hydra-Nav [27], NavFoM [41], and Nav- R^2 [31]. These methods better leverage semantic priors and reduce error accumulation from modular pipelines, but placing large VLMs in the control loop tightly couples high-level reasoning with low-level control, limiting real-time and smooth long-horizon navigation [28]. To mitigate this issue, a third line adopts dual-system designs, where VLMs perform sparse high-level reasoning while lightweight controllers handle fast local execution, as in Nav-R1 [13], FiLM-Nav [35], and InternVLA-N1 [28].

Despite these advances, existing dual-system methods still face two key challenges. First, sparse VLM reasoning improves efficiency but leaves many intermediate observations unused, even though they may contain transient evidence of the target or distractors. Second, target localization and spatial reasoning often remain implicit in the decision-making process. In this paper, we address these limitations by learning a structured goal-oriented CoT mechanism.

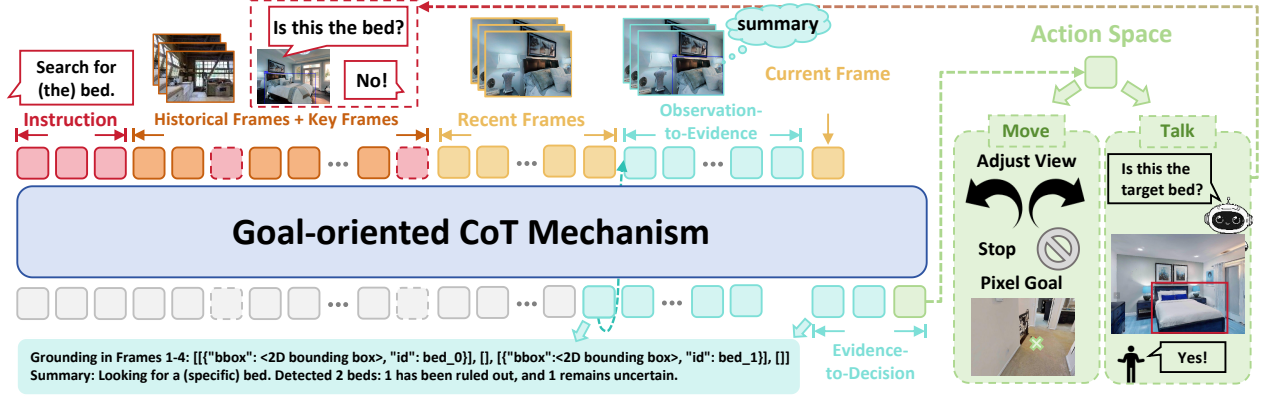


Figure 2 Inference process of GG-Nav. Given the instruction, compact visual history, dialogue-associated evidence, and intermediate frames since the previous decision step, GG-Nav first converts the intermediate frames into accumulated grounding evidence through Observation-to-Evidence, and then uses Evidence-to-Decision to integrate this evidence with the current observation and generate either a navigation action or a clarification question.

3 Motivating Study

In this section, we aim to conduct a quantitative analysis on top of a basic adapted navigation foundation model to investigate the key challenges in goal-oriented navigation.

Existing methods typically construct visual history by uniformly sampling a small number of past frames and adding dialogue-related key frames. For example, InternVLA-N1 [25] uses eight uniformly sampled history frames together with interaction frames as visual context. We fine-tune the System 2 component of InternVLA-N1 on a targeted goal-oriented navigation dataset and evaluate it on 500 episodes from VL-LN Bench [9]. This baseline achieves a success rate of 25.4%.

Table 1 Failure analysis of adapted InternVLA-N1 on VL-LN Bench.

Target visibility	Error type	# Failures	Ratio
Not observed 119 (32%)	Exploration error	119	32%
Target appears in observations 254 (68%)	Evidence miss	153	41%
	Reaching error	26	7%
	Stopping error	75	20%
Total failures		373	100%

As summarized in Tab. 1, we further analyze the 373 failure cases and find that the target object actually appears in the agent’s RGB observations in 68% (254) of failed episodes. Among all failures, 41% (153) occur because the target appears only briefly in intermediate frames that are not selected by the sparse history sampling strategy, causing the model to miss critical evidence. Another 20% (75) correspond to cases where the model detects and even confirms the target through interaction but still fails to stop accurately near the target. This indicates that implicit action prediction is not sufficient to translate detected evidence into precise target localization and reliable navigation. These observations suggest that applying navigation foundation models to goal-oriented navigation still faces two key challenges: discovering goal-relevant evidence from streaming observations, and grounding and reusing accumulated evidence to localize the target and support reliable decision-making. These two challenges motivate the method presented next.

4 The Proposed Framework

4.1 Task Formulation

We consider two text-instructed goal-oriented navigation tasks: ObjectNav [2] and IIGN [9]. Given an instruction I_c specifying a target category c , let $\mathcal{G}(I_c)$ denote the set of valid goal instances. In ObjectNav, any instance of category c is a valid goal, whereas IIGN requires the agent to identify a unique target instance v^* among multiple same-category candidates. The agent must therefore accumulate visual evidence during exploration and may ask a human oracle questions to disambiguate the target instance.

As shown in Fig. 2, the action space consists of navigation actions $\mathcal{A}^{\text{move}}$, including view adjustment, pixel-goal navigation, and STOP. In IIGN, the agent may additionally choose a dialogue action from $\mathcal{A}^{\text{talk}}$, represented as a natural-language question grounded in one or more observed candidate instances. The oracle’s answer is then appended to the dialogue history for subsequent decision-making. An episode is considered successful if the agent issues STOP within the allowed step budget and terminates sufficiently close to a valid goal instance in $\mathcal{G}(I_c)$. As a result, ObjectNav regards any instance of category c as a valid goal, whereas IIGN accepts only the intended target instance v^* .

4.2 Goal-oriented Grounding CoT Mechanism

Goal-oriented CoT Formulation. Building on the above analysis, GG-Nav aims to equip a VLM with a structured reasoning mechanism that discovers goal-relevant evidence from streaming observations and grounds accumulated evidence for reliable decision-making. For implementation, we build GG-Nav upon Qwen3-VL-4B [1] and formulate each decision step i as a structured reasoning trace $\mathcal{C}_i^{\text{goal}} = (R_i^{\text{OE}}, R_i^{\text{ED}}, a_i)$, which decomposes the reasoning process into Observation-to-Evidence (OE) and Evidence-to-Decision (ED). Specifically, R_i^{OE} revisits intermediate observations to recover goal-relevant evidence through frame-wise grounding records and evidence summaries; R_i^{ED} grounds the current observation and associates visible candidates with accumulated evidence; and a_i denotes the resulting action.

4.2.1 Observation-to-Evidence

Observation-to-Evidence is triggered when the agent reaches a new decision step and invokes the VLM. Its inputs include the instruction, a compact set of historical frames, key frames with grounded instances, and intermediate frames collected since the previous decision step. The compact set of historical frames consists of frames uniformly sampled between the beginning of the task and the previous decision step. Each key frame is paired with its grounding result and, if available, its question-answering record, informing the model of previously examined candidates and providing visual evidence for instance tracking. This context enables the model to associate newly observed candidates with recorded instances.

Conditioned on this context, the model reasons over every intermediate frame and generates an Observation-to-Evidence CoT (R_i^{OE}). For each frame, it produces a structured JSON record containing the 2D bounding boxes and temporally consistent IDs of all instances belonging to the instructed category. It then summarizes which instances have been previously encountered and which are newly observed. These frame-wise grounding records and summaries transform transient inter-decision observations into explicit evidence, allowing the model to better exploit the streaming observation.

4.2.2 Evidence-to-Decision

This stage integrates the accumulated grounding evidence with the current observation to support action generation. The model first produces a current-frame grounding result, explicitly predicting the bounding boxes and consistent IDs of goal-relevant instances and associating them with the accumulated evidence R_i^{ED} . It then conditions on all the information to generate the next decision (a_i). In ObjectNav, the output is restricted to $a_i \in \mathcal{A}^{\text{move}}$; in IIGN, $a_i \in \mathcal{A}^{\text{move}} \cup \mathcal{A}^{\text{talk}}$. For dialogue actions, the model asks a question grounded in selected intermediate or current frames and their instance IDs. After receiving a reply from the human or oracle, the question-answer pair is stored with the grounded frame as evidence for accumulation and subsequent decision making. This stage provides a spatially verifiable basis for target localization and decision-making.

4.3 Data Collection

We fine-tune GG-Nav with two sources of task-specific supervision, namely *videos with grounding annotations* and *navigation trajectory data*, to better follow the proposed goal-oriented CoT mechanism. The former provides dense supervision for Observation-to-Evidence by supervising instance grounding and tracking without requiring action annotations, while the latter supervises the full task execution process.

Videos with grounding annotations. To enhance temporal grounding and evidence accumulation, we construct training samples from video streams with object-level grounding annotations. We use the Aria Synthetic Environments Dataset, which provides continuous egocentric videos with object segmentation masks. For each video, we first sample an object category c that appears in the scene. We then sample K context frames containing at least one instance of category c and extract a contiguous clip of L frames from the same video stream. Each context frame is paired with the grounding annotations of the selected category, including 2D bounding boxes and instance IDs. An Observation-to-Evidence sample can be represented as

$$s^{\text{oe}} := \left\{ \left\{ I_c, \{(\bar{o}_j, \bar{g}_j)\}_{j=1}^K, \{o_t\}_{t=1}^L \right\} \rightarrow \left\{ \{g_t\}_{t=1}^L, r \right\} \right\},$$

where the left side denotes the input, with I_c representing the category-level instruction, $(\bar{o}_j, \bar{g}_j)$ the j -th context frame and its grounding annotation, and o_t the t -th intermediate frame, while the right side denotes the output, with g_t representing the corresponding structured grounding record and r the accumulated-evidence summary. This construction follows the format of Observation-to-Evidence and enables the use of video data without action labels.

Navigation trajectory data. To jointly supervise all stages in GG-Nav, we further collect ObjectNav and IIGN trajectories in Habitat [22]. Following prior work [9], we use annotated scenes from the training splits of HM3D [18] and MP3D [5]. For each episode, we use a frontier-based exploration algorithm to iteratively select frontier waypoints as navigation targets. The selected waypoints are connected by an A* planner [8] to obtain collision-free paths, which are then converted into executable low-level actions.

During data collection, we record RGB observations, agent poses, action labels, and the 2D bounding boxes and instance IDs of all visible instances belonging to the target category. For IIGN episodes, we construct interaction supervision by generating question-answer pairs with a rule-based procedure. The question is triggered only when the agent observes a previously unseen candidate instance from the target category. The answer is then produced by an automated oracle according to the ground-truth target identity. The resulting question-answer pair is added to the data as dialogue history.

After generating trajectories, we split each trajectory into decision-step segments and use each segment as a training sample. Formally, the i -th training segment is represented as

$$s_i^{\text{nav}} := \left\{ \left\{ I_c, H_i, D_i, O_i^{\text{rec}}, o_i^{\text{cur}} \right\} \rightarrow \left\{ R_i^{\text{OE}}, R_i^{\text{ED}}, a_i \right\} \right\},$$

where H_i denotes the compact historical visual memory, D_i denotes the accumulated dialogue history, O_i^{rec} is the sequence of intermediate frames retained since the previous decision, and o_i^{cur} is the current observation. R_i^{OE} denotes the Observation-to-Evidence output, R_i^{ED} denotes the current grounding component of Evidence-to-Decision, and a_i is the supervised navigation or interaction action.

In particular, in this work, we generate approximately 30K videos with grounding annotations and 60K navigation trajectories, comprising 20K ObjectNav trajectories and 40K IIGN trajectories, to fine-tune the model.

5 Experiments

Our experiments are designed to answer four questions: 1) whether GG-Nav improves goal-oriented navigation; 2) whether intermediate observations reduce failures caused by missed evidence; 3) whether explicit grounding improves target localization and stopping; and 4) whether the learned mechanism transfers to unseen real-world environments.

Evaluation metrics. We evaluate both ObjectNav and IIGN using standard navigation metrics. **SR** measures whether the agent successfully stops within 0.25 m of a target object. **SPL** additionally accounts for path

efficiency by weighting successful episodes according to the ratio of the shortest path length to the executed path length. For IIGN, we additionally report **NQ**, the average number of questions asked per episode, to quantify the interaction cost of instance disambiguation.

Implementation details. We initialize GG-Nav from Qwen3-VL-4B-Instruct and fully fine-tune the vision encoder, multimodal projector, and language-model backbone for 2.5 epochs. We use AdamW [15] with a learning rate of 2×10^{-5} for the language-model backbone and multimodal projector, a smaller learning rate of 5×10^{-6} for the vision encoder, and a global batch size of 128. Additional training details are provided in Appendix A.

Table 2 Baseline comparison on HM3D ObjectNav and the VL-LN and CoIN IIGN benchmarks.

Method	Interaction	HM3Dv2		VL-LN			CoIN (Seen)			CoIN (Seen Synonyms)			CoIN (Unseen)		
		SR $\uparrow$	SPL $\uparrow$	SR $\uparrow$	SPL $\uparrow$	NQ $\downarrow$	SR $\uparrow$	SPL $\uparrow$	NQ $\downarrow$	SR $\uparrow$	SPL $\uparrow$	NQ $\downarrow$	SR $\uparrow$	SPL $\uparrow$	NQ $\downarrow$
FBE	$\times$	31.5	13.69	9.2	5.09	-	4.69	2.75	-	2.35	1.55	-	3.92	2.05	-
VLFM [36]	$\times$	63.6	32.5	8.8	5.63	-	0.48	0.29	-	0.28	0.02	-	0.22	0.10	-
MTU3D [44]	$\times$	59.3	19.87	12.8	5.44	-	27.79	11.23	-	26.54	12.95	-	8.71	3.34	-
InternVLA-N1 [25]	$\checkmark$	69.2	27.61	25.4	16.18	1.98	17.57	9.51	2.79	20.94	11.13	2.54	17.21	8.20	2.61
AIUTA [24]	$\checkmark$	-	-	4.0	2.16	5.42	7.42	2.92	1.67	14.38	7.99	1.36	6.67	2.30	1.13
GG-Nav	$\checkmark$	72.3	31.12	42.0	23.22	3.01	29.96	13.54	4.66	31.86	16.62	4.92	19.61	8.82	1.79

5.1 Baseline Comparison

We evaluate GG-Nav on three benchmarks: HM3D for ObjectNav, and VL-LN and CoIN for IIGN. We compare it with representative non-interactive ObjectNav baselines, including FBE, VLFM [36], and MTU3D [44], as well as interactive IIGN baselines, including InternVLA-N1 [25] and AIUTA [24]. Non-interactive methods only execute navigation actions, while interactive methods can additionally ask clarification questions.

As shown in Tab. 2, GG-Nav achieves the best overall performance under both ObjectNav and IIGN settings. Specifically, on the HM3D ObjectNav benchmark, GG-Nav obtains 72.3% SR and 31.12% SPL, improving over the strongest baseline, InternVLA-N1, by 3.1% in SR and 3.51% in SPL. On the VL-LN benchmark, GG-Nav achieves 42.0% SR, outperforming the second-best method by 16.6%. Moreover, on all three evaluation splits of CoIN, GG-Nav consistently achieves the best SR and SPL. These consistent gains demonstrate that GG-Nav provides a unified and effective framework for both ObjectNav and IIGN.

Modular methods remain limited in solving the IIGN problem. Specifically, even with careful parameter tuning, AIUTA achieves only 4%, 7.42%, and 6.67% on the VL-LN benchmark and the seen/unseen splits of the CoIN benchmark, respectively. This limitation is mainly because existing modular methods construct semantic maps based on image-level scores, which can indicate whether a view contains the target category but cannot explicitly localize or bind a specific instance. As a result, when multiple candidates appear in an image, these methods may fail to distinguish the correct target, leading to incorrect navigation decisions. In addition, introducing more specialized modules for questioning and object tracking can make the modular pipeline more fragile, because errors may accumulate across modules and eventually result in incorrect navigation decisions.

5.2 Ablation Study I: Fundamental Designs

We conduct a progressive ablation study to verify the contribution of each design component. The action-only baseline takes the instruction, historical key frames, and dialogue-associated frames as context and directly predicts the next action. We then add intermediate frames as additional observations, while still requiring the model to output only the final action. The third variant further introduces Observation-to-Evidence, which converts intermediate frames into structured grounding evidence, while the full GG-Nav model additionally performs Evidence-to-Decision to integrate this evidence for action generation.

As shown in Tab. 3, our full model achieves the highest success rates across all three benchmarks, indicating the effectiveness of our overall design. Compared with the action-only baseline, the full model improves SR from 62.5% to 72.3% on HM3D, from 20.8% to 42.0% on VL-LN, and from 18.0% to 31.9% on CoIN. The gains in SPL are also substantial on HM3D and VL-LN, increasing from 21.56% to 31.12% and from 10.56%

Table 3 Progressive ablation study.

Variant	Setting			HM3D		VL-LN			CoIN (Seen Synonyms)		
	Inter. Frames	O-to-E	E-to-D	SR $\uparrow$	SPL $\uparrow$	SR $\uparrow$	SPL $\uparrow$	NQ $\downarrow$	SR $\uparrow$	SPL $\uparrow$	NQ $\downarrow$
Action-only	$\times$	$\times$	$\times$	62.5	21.56	20.8	10.56	2.44	18.0	11.99	2.14
+ Intermediate frames	$\checkmark$	$\times$	$\times$	66.7	25.78	36.8	20.39	2.79	28.3	18.13	3.82
+ Observation-to-Evidence	$\checkmark$	$\checkmark$	$\times$	68.9	26.94	36.6	20.08	2.80	27.7	14.68	3.66
+ Evidence-to-Decision	$\checkmark$	$\checkmark$	$\checkmark$	72.3	31.12	42.0	23.22	3.01	31.9	16.62	4.92

Note. ‘Inter. Frames’ denotes using intermediate frames; O-to-E and E-to-D denote Observation-to-Evidence and Evidence-to-Decision.

to 23.22%, respectively. These results show that intermediate observations provide useful evidence beyond key frames, while Observation-to-Evidence and explicit current-frame grounding within Evidence-to-Decision further improve decision making. We also observe that the full model asks more questions, suggesting a trade-off between navigation success and interaction efficiency in ambiguous scenes.

To better understand failure modes, we conduct a fully automated stage-wise error analysis using simulator instance-segmentation observations, agent grounding/tracking logs, and final poses. Each failed episode is attributed to one pipeline stage: *exploration*, *grounding*, *tracking*, *reaching*, or *stopping*. Exploration errors occur when the target instance never appears in the simulator’s observations; grounding errors when it appears but is never associated with the target instance ID in the model’s grounding output; tracking errors when a previously grounded target is later lost or switched to another instance; reaching errors when the correct target association is maintained but the final position is at least 1 m from the target; and stopping errors when the final position is within 1 m but more than 0.25 m from the target.

Table 4 Stage-wise error analysis across benchmarks.

Benchmark	Variant	E	G	T	R	S
HM3D	Action-only	111	162	-	-	102
	+ Intermediate frames	107	141	-	-	85
	+ Observation-to-Evidence	93	154	6	20	38
	+ Evidence-to-Decision	92	122	4	27	32
VL-LN	Action-only	163	164	-	54	15
	+ Intermediate frames	123	143	-	23	27
	+ Observation-to-Evidence	126	147	22	9	13
	+ Evidence-to-Decision	137	103	15	17	18
CoIN	Action-only	32	155	-	51	40
	+ Intermediate frames	26	141	-	17	59
	+ Observation-to-Evidence	21	153	39	13	19
	+ Evidence-to-Decision	29	126	23	25	28

Note. Each failed episode is attributed to the stage in the pipeline: Exploration (E), Grounding (G), Tracking (T), Reaching (R), and Stopping (S).

Introducing intermediate observations consistently reduces grounding errors by enabling the agent to exploit recently observed images for decision making. Specifically, compared with the ‘Action-only’ variant, the ‘+ Intermediate frames’ variant decreases grounding errors from 162 to 141 on HM3D, from 164 to 143 on VL-LN, and from 155 to 141 on CoIN, corresponding to an average reduction of 11.6%. These reductions suggest that a non-negligible portion of grounding failures stems from the limited use of observations under sparse decision making: the target instance may appear in frames between two decision steps, but this evidence is unavailable to the action-only baseline when predicting the next action. By incorporating intermediate observations, the agent can carefully examine such skipped frames and better ground potential targets.

Explicit grounding also improves target localization and enables the agent to stop at more precise positions. This effect is reflected in the reduction of stopping errors when moving from the ‘+ Intermediate frames’ variant to the full model: the number of stopping errors decreases from 85 to 32 on HM3D, from 27 to 18 on VL-LN, and from 59 to 28 on CoIN, corresponding to an average reduction of 49.4%. These results

Table 5 Ablation on dialog-frame retention strategies.

Strategy	Setting	SR↑	SPL↑	NQ↓	Tokens ↓
Decision keyframe	50	43.4	24.36	3.32	3204
	100	41.2	22.62	3.08	3270
	200	42.0	23.22	3.01	3381
	400	41.0	22.62	2.74	3443
Spatially local keyframe	3	41.6	24.02	3.13	3191
	6	40.2	22.83	2.92	3324
	12	39.6	22.05	2.79	3447
	24	40.4	23.03	2.74	3470
Dialog keyframe	2	41.2	23.88	3.17	3279
	4	41.0	22.66	2.71	3370
	8	39.8	22.81	2.69	3444
	16	38.4	21.90	2.63	3452

Note. Tokens denotes the average number of input tokens.

suggest that explicit target grounding provides a concrete visual basis for both predicting **STOP** and selecting navigation waypoints, leading the agent to stop more accurately.

Despite these improvements, the remaining failures are still primarily concentrated in exploration and grounding errors. In the full model, these two categories account for 77.3% of all failures on HM3D, 82.8% on VL-LN, and 67.1% on CoIN. This distribution indicates that the major bottlenecks lie in long-term memory and instance-level grounding. Further progress therefore likely requires stronger memory mechanisms that help the agent identify under-explored regions more effectively, as well as more robust grounding capability that can distinguish and maintain target instances throughout navigation.

5.3 Ablation Study II: Dialog-frame Retention

To support long-horizon instance grounding, GG-Nav accumulates grounding evidence over time and reuses it as memory for later instance tracking and decision making. However, continuously retaining all historical evidence leads to an ever-growing memory, which increases inference cost and may introduce noise into decision-making. Therefore, we investigate how selective dialog-frame retention affects long-horizon IIGN performance. We consider three memory variants: Decision keyframe, Spatially local keyframe, and Dialog keyframe, which compress the full dialog history into a small set of selected frames. Specifically, the Decision keyframe strategy retains dialog frames generated within the most recent K_t decision steps, the Spatially local keyframe strategy preserves dialog frames whose recorded positions fall within a radius r (in meters) of the current decision location, and the Dialog keyframe strategy keeps the most recently generated K_f dialog frames. We conduct this experiment on VL-LN Bench, and the results are shown in Tab. 5.

The results reveal a consistent trade-off across the three dialog-frame retention strategies: retaining fewer dialog frames generally improves navigation performance but increases the number of clarification questions. For Decision keyframe, the shortest window of 50 decision steps achieves the best SR and SPL, reaching 43.4% and 24.36%, but increases NQ to 3.32. In contrast, expanding the window to 400 reduces NQ to 2.74, while lowering SR and SPL to 41.0% and 22.62%. Similar trends appear for Spatially local keyframe and Dialog keyframe: more compact dialog-frame memory reduces noisy grounding evidence and benefits decision-making, but may discard previously disambiguated instances, causing the agent to ask repeated clarification questions.

Among the three dialog-frame retention strategies, Decision keyframe achieves the best performance. Its 50-step setting obtains the best SR and SPL among all variants, outperforming the strongest settings of Spatially local keyframe and Dialog keyframe while using a comparable number of input tokens. This suggests that Decision keyframe better aligns with the temporal nature of interactive instance grounding: it preserves sufficient visual continuity for target tracking, whereas Spatially local keyframe may discard visible but distant instance evidence, and Dialog keyframe may suffer from unstable tracking in instance-dense scenes.

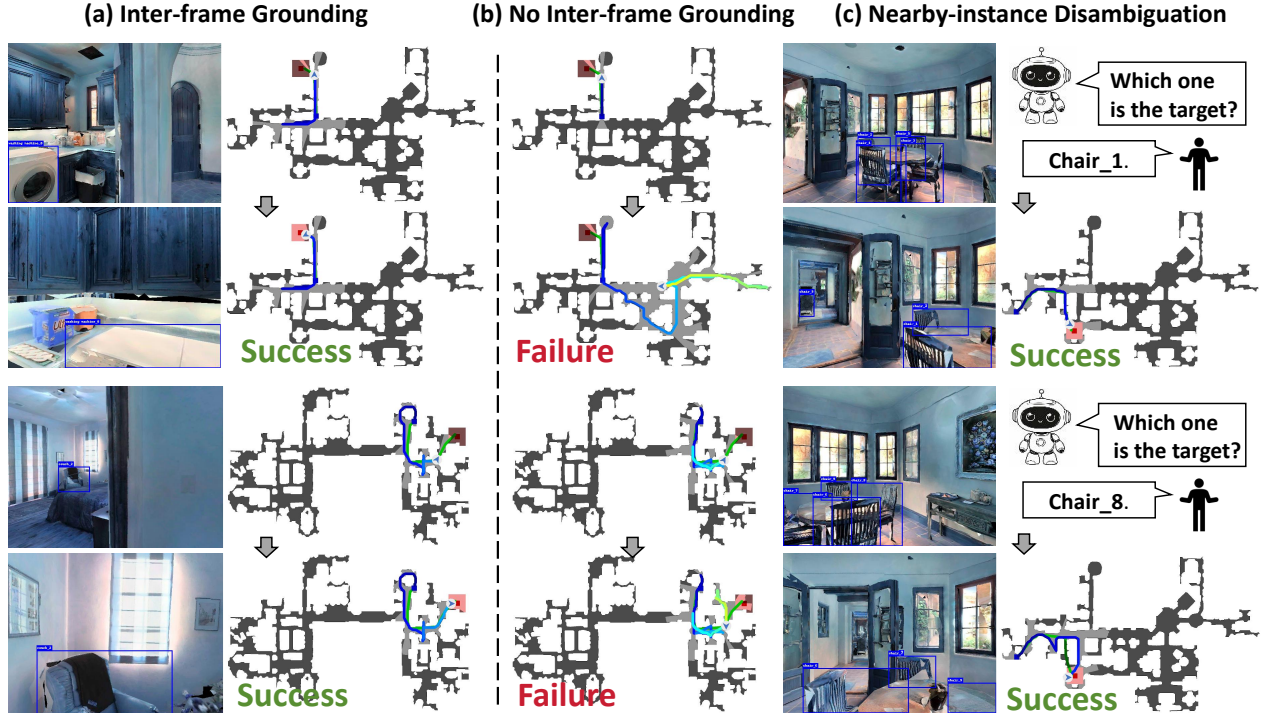


Figure 3 Case studies on using intermediate observations and disambiguating similar targets a in distractor-rich scene.

5.4 Case Study

To better understand where GG-Nav provides the most substantial benefits, we present qualitative case studies in two representative scenarios: intermediate-frame grounding (Fig. 3(a)-(b)) and similar-instance disambiguation (Fig. 3(c)). We first examine episodes in which the target instance appears only in intermediate frames between two decision steps, highlighting the importance of incorporating intermediate observations into the decision-making process. We then analyze target-dense scenes that contain multiple nearby and visually similar instances from the same category, demonstrating that GG-Nav can robustly disambiguate visually similar instances and maintain the correct referent over time. These two scenarios qualitatively demonstrate that GG-Nav addresses the two issues identified above by fully leveraging intermediate-frame evidence and accurately grounding the target instance despite distractors.

Intermediate observations are critical for decision making because the target may appear only briefly between adjacent decision steps. The paired examples in Fig. 3(a)-(b) compare episodes with and without intermediate observations. In Fig. 3(a), the first column shows the observations and grounding results when the target first appears and at the final step, while the second column and Fig. 3(b) show the corresponding exploration trajectories. Without intermediate-frame observations, briefly visible targets may be missed, causing the agent to treat the region as target-free and move away from the goal.

GG-Nav can also follow human clarification and stop at the correct target in the presence of visually similar nearby distractors. As shown in Fig. 3(c), multiple chairs appear around the same table, making it challenging to identify and track the intended target during navigation. After the agent asks for clarification, the human specifies the target instance, such as *Chair_1* or *Chair_8*. In both cases, the agent successfully preserves the clarified referent, navigates to the selected chair, and stops near the correct target while facing it. This demonstrates that the proposed grounding mechanism can maintain instance-level consistency even in target-dense scenes with strong visual ambiguity.

5.5 Real-world Experiment

Experiment Setup. We conduct real-world experiments on a Unitree Go2 quadruped robot. The robot is equipped with an Intel RealSense D455 RGB-D camera mounted at a height of 0.7 m and tilted downward

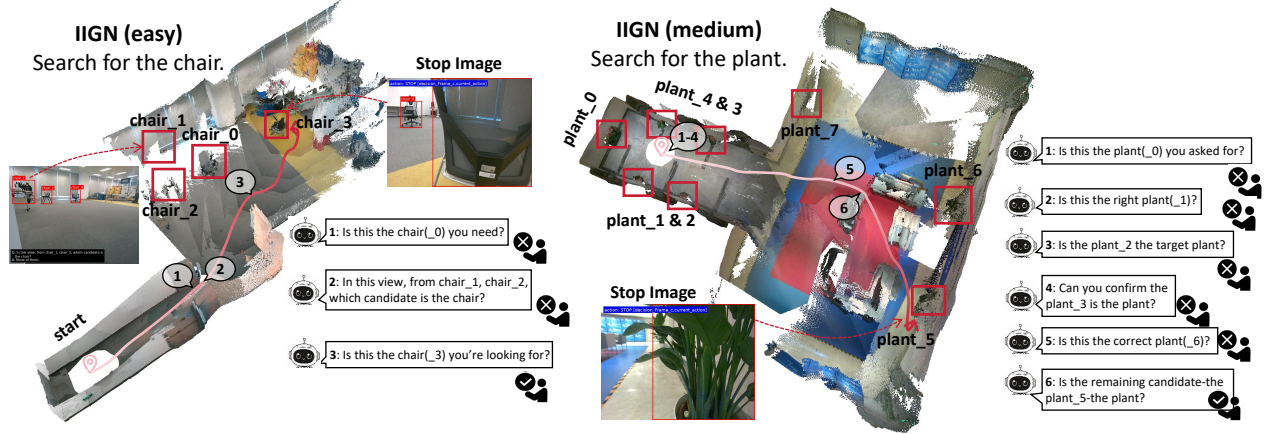


Figure 4 Real-world navigation visualization. The left side shows the IIGN (easy) scenario, where the agent is required to search for ‘chair_3’ among three distractors. The right side shows the IIGN (medium) scenario, where the robot needs to reach ‘plant_5’ among seven distractors. The robot-human dialogues are shown on the right, while selected observation frames and the final stopping frames are displayed next to the reconstructed scenes.

by 15°. GG-Nav runs on a remote server with an RTX 4090 GPU. Given a goal-oriented instruction, the robot streams RGB-D observations to the server for inference. The model predicts either a navigation action or a clarification question. A navigation action is a discrete view adjustment or a pixel goal. Pixel goals are back-projected with RGB-D observations, transformed into world-frame waypoints using odometry, and tracked by an onboard MPC controller; discrete actions are executed by a PID-mode pose controller. When the model asks a clarification question, a human provides the answer through the server interface.

Quantitative Analysis. To quantitatively evaluate the robustness and generalization of GG-Nav in real-world settings, we compare it with InternVLA-N1 [25] on ObjectNav and three IIGN settings with increasing ambiguity and navigation difficulty. Performance is measured by Success Rate (SR); an episode is successful only if the robot stops within 0.25 m of, and faces, the intended target instance. To better visualize the long-horizon interactive exploration process, we use VGGT- Ω [26] to reconstruct the scene. This reconstruction is used only for visualization; GG-Nav itself does not require map construction during navigation. The ObjectNav and IIGN (hard) settings are shown in Fig. 1, while the IIGN (easy) and IIGN (medium) settings are shown in Fig. 4. Fig. 5 presents the quantitative comparison between GG-Nav and InternVLA-N1, showing that GG-Nav consistently outperforms InternVLA-N1 across ObjectNav and all IIGN settings.

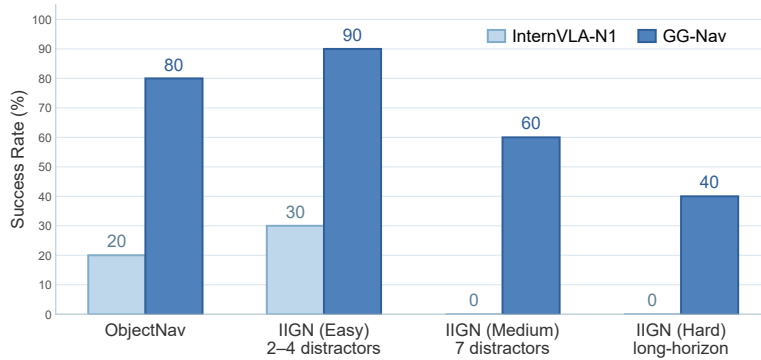


Figure 5 Real-world navigation success rates.

Qualitative Analysis. Detailed navigation processes of GG-Nav and InternVLA-N1 are provided in the supplementary video. We observe that InternVLA-N1 often fails to effectively rule out each potential target candidate and to stop within the 0.25 m success threshold. By converting intermediate frames into accumulated grounding evidence through Observation-to-Evidence, GG-Nav retains transient observations of potential

target instances. As shown in Fig. 4, in the IIGN (easy) setting, the agent actively asks clarification questions about ‘chair_1’ and ‘chair_2’, even though each instance is visible in only a single frame during the entire process. In the IIGN (medium) setting, the agent starts in a dense cluster of visually similar plants, yet it can still accurately distinguish different instances and ask targeted clarification questions for disambiguation. Moreover, as shown in the stopping frames of both settings, GG-Nav can accurately stop near the target instances while facing them.

6 Conclusion

We have presented GG-Nav, a navigation framework that learns a structured CoT mechanism to address insufficient observation utilization and imprecise target localization in goal-oriented navigation. Our results highlight that reliable navigation depends not only on the generation capability of vision-language models, but also on preserving transient visual evidence and maintaining consistent target grounding throughout navigation. Evaluations across ObjectNav and IIGN benchmarks demonstrate that GG-Nav consistently improves navigation and stopping reliability, while its deployment on a Unitree Go2 robot further validates its applicability to unseen real-world environments. These findings show that explicitly extracting goal-relevant visual information as structured evidence and reasoning over this evidence can improve the reliability of navigation decisions. By supporting the reliable execution of individual goal-oriented navigation subtasks, our proposed framework further lays a foundation for long-horizon navigation, whose tasks can often be decomposed into a sequence of such subtasks.

References

- [1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.
- [2] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020.
- [3] Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5228–5234. IEEE, 2024.
- [4] Yihan Cao, Jiazhao Zhang, Zhinan Yu, Shuzhen Liu, Zheng Qin, Qin Zou, Bo Du, and Kai Xu. Cognav: Cognitive process modeling for object goal navigation with llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9550–9560, 2025.
- [5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *IEEE International Conference on 3D Vision (3DV)*, Oct. 2017.
- [6] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33: 4247–4258, 2020.
- [7] Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to map for active semantic goal navigation. *arXiv preprint arXiv:2106.15648*, 2021.
- [8] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [9] Wensi Huang, Shaohao Zhu, Meng Wei, Jinming Xu, Xihui Liu, Hanqing Wang, Tai Wang, Feng Zhao, and Jiangmiao Pang. Vl-ln bench: Towards long-horizon goal-oriented navigation with active dialogs. *arXiv preprint arXiv:2512.22342*, 2025.
- [10] I-Tak Ieong and Hao Tang. Multimodal perception for goal-oriented navigation: A survey. *arXiv preprint arXiv:2504.15643*, 2025.
- [11] Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zolt 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 (CVPR)*, 2024.
- [12] Yuxuan Kuang, Hai Lin, and Meng Jiang. Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 338–351, 2024.
 - [13] Qingxiang Liu, Ting Huang, Zeyu Zhang, and Hao Tang. Nav-r1: Reasoning and navigation in embodied scenes. *arXiv preprint arXiv:2509.10884*, 2025.
 - [14] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.
 - [15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
 - [16] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: Treasure hunt data augmentation for semantic navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15374–15383, 2021.
 - [17] Dujun Nie, Xianda Guo, Yiqun Duan, Ruijun Zhang, and Long Chen. Wmnav: Integrating vision-language models into world models for object goal navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2392–2399. IEEE, 2025.
 - [18] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL <https://openreview.net/forum?id=v4OuqNs5P>.
 - [19] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022.
 - [20] Ram Ramrakhyia, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
 - [21] Ram Ramrakhyia, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
 - [22] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.
 - [23] Qi Sun, Pengfei Hong, Tej Deep Pala, Vernon Toh, U-Xuan Tan, Deepanway Ghosal, and Soujanya Poria. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. *arXiv preprint arXiv:2412.11974*, 2024.
 - [24] 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 (ICCV)*, pages 18781–18792, October 2025.
 - [25] InternNav Team. InternVLA-N1: An open dual-system navigation foundation model with learned latent plans, 2025.
 - [26] Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-omega, 2026. URL <https://arxiv.org/abs/2605.15195>.
 - [27] Zixuan Wang, Huang Fang, Shaoan Wang, Yuanfei Luo, Heng Dong, Wei Li, and Yiming Gan. Hydra-nav: Object navigation via adaptive dual-process reasoning. *arXiv preprint arXiv:2602.09972*, 2026.

- [28] Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *arXiv preprint arXiv:2512.08186*, 2025.
- [29] Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.
- [30] Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. *arXiv preprint arXiv:2401.02695*, 2024.
- [31] Wentao Xiang, Haokang Zhang, Tianhang Yang, Zedong Chu, Ruihang Chu, Shichao Xie, Yujian Yuan, Jian Sun, Zhining Gu, Junjie Wang, et al. Nav- r^2 : Dual-relation reasoning for generalizable open-vocabulary object-goal navigation. *arXiv preprint arXiv:2512.02400*, 2025.
- [32] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectnav. *arXiv preprint arXiv:2104.04112*, 2021.
- [33] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *Advances in neural information processing systems*, 37:5285–5307, 2024.
- [34] 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*, pages 19057–19066, 2025.
- [35] Naoki Yokoyama and Sehoon Ha. Film-nav: Efficient and generalizable navigation via vlm fine-tuning. *arXiv preprint arXiv:2509.16445*, 2025.
- [36] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48. IEEE, 2024.
- [37] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. *arXiv preprint arXiv:2304.05501*, 2023.
- [38] Mingming Yu, Fei Zhu, Wenzhuo Liu, Yirong Yang, Qunbo Wang, Jing Liu, et al. C-nav: Towards self-evolving continual object navigation in open world. *Advances in Neural Information Processing Systems*, 38:126181–126207, 2026.
- [39] Shuaihang Yuan, Hao Huang, Yu Hao, Congcong Wen, Anthony Tzes, and Yi Fang. Gamap: Zero-shot object goal navigation with multi-scale geometric-affordance guidance. *Advances in Neural Information Processing Systems*, 37:39386–39408, 2024.
- [40] Jiazhaoh 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*, 2024.
- [41] Jiazhaoh Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025.
- [42] Lingfeng Zhang, Qiang Zhang, Hao Wang, Erjia Xiao, Zixuan Jiang, Honglei Chen, and Renjing Xu. Trihelper: Zero-shot object navigation with dynamic assistance. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10035–10042. IEEE, 2024.
- [43] Zibo Zhou, Yue Hu, Lingkai Zhang, Zonglin Li, and Siheng Chen. Beliefmapnav: 3d voxel-based belief map for zero-shot object navigation. *arXiv preprint arXiv:2506.06487*, 2025.
- [44] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8120–8132, 2025.

A Implementation Details

We provide the training setup, data mixture, and inference configuration needed to reproduce GG-Nav.

A.1 Training Setup

GG-Nav is initialized from Qwen3-VL-4B-Instruct. We optimize the vision encoder, multimodal projector, and language-model backbone for 2.5 epochs using AdamW. The learning rate is 2×10^{-5} for the language-model backbone and multimodal projector and 5×10^{-6} for the vision encoder. We use a global batch size of 128, zero weight decay, a warm-up ratio of 0.03, a cosine learning-rate schedule, and a maximum gradient norm of 1. Training uses bfloat16, gradient checkpointing, and DeepSpeed ZeRO-3, with a maximum sequence length of 8,192 tokens. Input RGB images are resized to 384×384 . Following the visual-token constraints of Qwen3-VL, the minimum and maximum input sizes are set to 3,136 and 313,600 pixels, respectively.

A.2 Training Data Mixture

The source corpus contains approximately 30K videos with grounding annotations and 60K navigation trajectories, comprising 20K ObjectNav trajectories and 40K IIGN trajectories. As described in Sec. 4.3, each source sequence yields multiple training instances. The resulting training set contains 1,000,000 temporal-grounding samples, 1,178,492 ObjectNav samples, and 3,543,549 IIGN samples. Thus, the 5,722,041-sample mixture comprises 17.5%, 20.6%, and 61.9% of these three supervision types, respectively.

A.3 Inference Configuration

Visual context. At each decision step, we uniformly sample eight historical frames from the start of the episode through the preceding decision step. We additionally retain key frames containing grounded target-category instances, together with the associated dialogue. All observations collected since the preceding decision are passed to Observation-to-Evidence. Unless stated otherwise, the benchmark experiments use the Decision-keyframe retention strategy with a window of 200 decision steps.

Action space. The navigation action space comprises view adjustment, pixel-goal navigation, and STOP. A view-adjustment action turns the agent left or right by an angle in $\{30^\circ, 60^\circ, 90^\circ, 120^\circ\}$. The model predicts a pixel goal as a normalized horizontal-vertical coordinate on a $[0, 1000]$ grid; this output is then converted to an image coordinate $[w, h]$, where $0 \leq w < W$ and $0 \leq h < H$. Depth and camera intrinsics are used to back-project this coordinate to a local waypoint. Habitat’s shortest-path follower executes the waypoint in simulation, whereas the real robot uses the MPC/PID control stack described in Appendix E. In IIGN, the model may instead produce a natural-language clarification question. Before predicting a pixel goal, it emits a downward-arrow token to request a downward-tilted observation; the corresponding simulation action changes camera elevation by 30° .

Decision timing. VLM inference is event-driven. After a pixel goal, the local controller executes a fixed number of low-level steps before the next VLM call. After a view adjustment, the requested rotation is completed before inference resumes. Following a clarification question, the model is invoked again after the answer has been appended to the dialogue history. Consequently, high-level decisions occur at action-dependent intervals rather than at a fixed frame rate.

Decoding and failure handling. For benchmark evaluation, we generate at most 2,048 new tokens using stochastic decoding with temperature 0.7 and nucleus-sampling threshold $p = 0.9$. We parse the generated output into grounding records and an action. If parsing fails or the action is invalid, the model is invoked again with the same decision context, for at most three attempts in total. An episode is marked unsuccessful if all three attempts fail.

B Structured Goal-Oriented Grounding CoT

B.1 Grounding Output Format

GG-Nav represents each frame-level grounding result as a JSON list. Every detected target-category instance has the form $\{ "bbox_2d": [x_{min}, y_{min}, x_{max}, y_{max}], "label": "c_k" \}$, where c is the instructed category and k is a model-assigned, temporally consistent track ID. Bounding-box coordinates are normalized to a $[0, 1000]$ scale and ordered as $[x_{min}, y_{min}, x_{max}, y_{max}]$. If no target-category instance is detected in a frame, the model outputs an empty list, $[]$.

B.2 Prompt and Output Examples

Observation-to-Evidence. The input consists of the compact visual history, the previous look-down frame, and the recent observations collected before the current decision. The model outputs one grounding list for each recent observation, in the same temporal order, followed by a summary of newly observed or still-uncertain instances. For example, its prompt is:

```
You are an autonomous navigation assistant. Your task is to search for the tv. These are your historical observations: <image>
x 8. Your previous look-down image is <image>. Now, first track object identities in the recent observations before the
current decision: <image> x 3. These tracking images are only for maintaining consistent identities, matching previously
seen instances, and summarizing newly discovered instances. Do not decide the final action yet.
```

For three recent observations of the same television, the corresponding output is:

```
<think>
""json
[{"bbox_2d": [814, 504, 1000, 652], "label": "tv_0"}]
""
""json
[{"bbox_2d": [471, 502, 614, 625], "label": "tv_0"}]
""
""json
[{"bbox_2d": [110, 502, 287, 643], "label": "tv_0"}]
""
I found 1 uncertain tv(s) in the recent tracking images.
Uncertain instances: tv_0.
</think>
```

Although the bounding box changes across viewpoints, the identifier `tv_0` remains consistent. This stage only accumulates grounding evidence and does not predict an action.

Evidence-to-Decision. The current observation is appended after the Observation-to-Evidence output. The model grounds the current frame and then generates a view-adjustment action, pixel goal, **STOP**, or clarification question. When asking about accumulated evidence, the prefix `Frame_k` binds a question to the k -th intermediate frame, whereas `Frame_c` denotes the current frame. An abbreviated prompt is:

```
Your current image is <image>. First ground the target-category instances in this image, and then decide the next action
from all available context. You may move or talk. Bind a question about an intermediate image to "Frame_k" and a
question about the current image to "Frame_c". To output a waypoint, first request a downward-tilted observation and
then predict its image coordinate. If the waypoint is outside the current view, use TURN LEFT or TURN RIGHT.
Output STOP only after the task has been completed.
```

Dialogue-action example. When the target is absent from the current frame but a candidate exists in the accumulated evidence, the model can ask a frame-grounded question:

```
<think>
I did not detect any tv in the current image.
</think>
<talk>Frame_1: Is this one the tv?</talk>
```

Here, the current-frame grounding is empty, but the previously accumulated `tv_0` evidence remains available. The model therefore asks about the candidate in `Frame_1` rather than discarding it or generating an ungrounded question.

Navigation-action example. When the target is grounded in the current observation, the model first requests an observation tilted downward by 30° . It then predicts a pixel goal on this look-down frame:

Assistant:

<think>Target tv found. Navigate to it.</think><move>↓

Input:

<image> (30-degree look-down frame)

Assistant:

<move>371 550

Here, 371 550 is the normalized horizontal-vertical coordinate on the $[0, 1000]$ grid. It is converted to the resolution of the look-down image and back-projected with depth and camera intrinsics to obtain a local waypoint for execution.

B.3 VLM-based Instance Association

The VLM predicts visible target-category instances and their temporally consistent track IDs from historical grounding records, retained key frames, and recent observations. It reuses an existing ID when the context supports an identity match and otherwise creates a new ID. We do not use an external object tracker, geometric matcher, or post-hoc identity-association module. Dialogue answers are stored with the corresponding grounded frame and track ID for subsequent decisions.

C Training Data Construction

C.1 Videos with Grounding Annotations

We construct Observation-to-Evidence training samples from continuous egocentric videos in the Aria Synthetic Environments Dataset. For each video, we sample an object category present in the scene, select K context frames containing at least one instance of that category, and extract a contiguous clip of L frames, where $K \in \{1, \dots, 8\}$ and $L \in \{1, 2, 3, 7\}$. We retain an instance only when its visible segmentation mask contains at least 200 pixels and convert the valid mask to its enclosing 2D bounding box. The context annotations condition the model, whereas the clip boxes, instance IDs, and evidence summary form the supervision. The resulting instances constitute the temporal-grounding portion of the training mixture summarized in Appendix A.

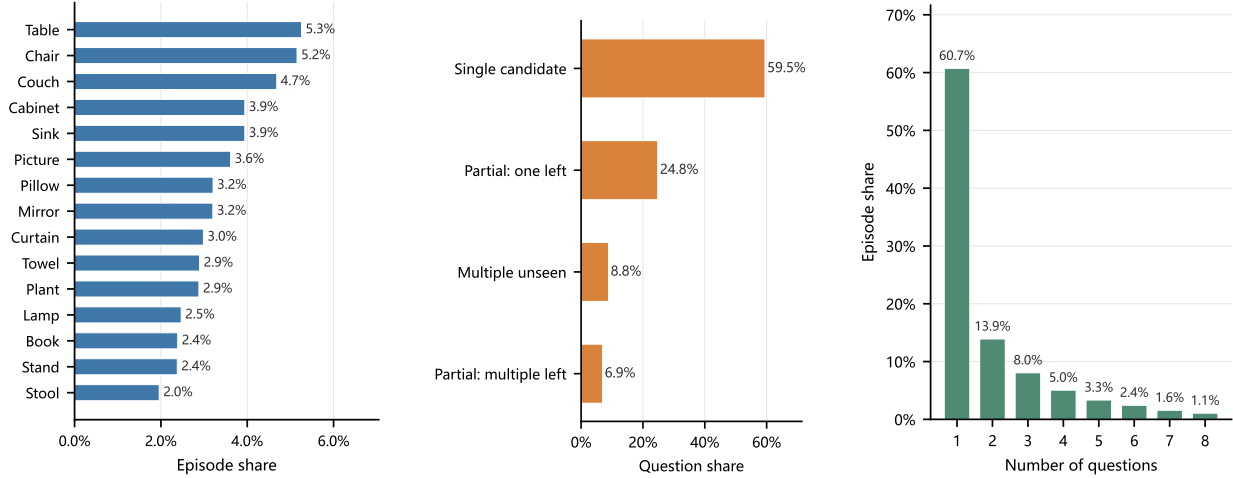
C.2 Navigation Trajectories

We generate IIGN episodes from all training scenes of VL-LN Bench and ObjectNav episodes from all training scenes of HM3D ObjectNav. The simulated agent uses 640×480 RGB-D observations with a 79° horizontal field of view. The camera is mounted at 1.25 m, and the depth range is 0–10 m. The agent radius is 0.18 m, with a 0.25 m forward step and a 30° turn angle.

During data collection, a frontier-based exploration policy repeatedly selects waypoints along the boundary of the explored free space, while an A* planner converts each waypoint into a collision-free action sequence. We record the observations, agent poses, actions, bounding boxes, and simulator instance IDs of all visible target-category objects. We retain only trajectories that satisfy two criteria: the target is visible in the final observation, and the trajectory contains no redundant exploration. The retained trajectories are then used to supervise both Observation-to-Evidence and Evidence-to-Decision learning.

C.3 Interaction Supervision

During navigation, we maintain a set of observed instances belonging to the target category and track whether each instance has been confirmed or ruled out through previous interactions. Whenever the agent observes one or more previously unseen candidate instances, we generate a question based on the number of newly observed candidates and the elimination history of previously observed ones. Representative question templates include:



(a) The 15 most frequent target categories.

(b) Candidate configuration.

(c) Clarification turns per episode.

Figure 6 Statistics of IIGN interaction supervision. (a) The 15 most frequent target categories, ranked by episode count; bar annotations report their shares of all 50,855 episodes. (b) The distribution of 118,309 clarification questions over candidate configurations. “Partial” denotes that some candidates have already been eliminated, while “one/multiple left” indicates whether the remaining uncertain set contains one or multiple candidates (two in this corpus). (c) The number of clarification turns per episode. For readability, only bins whose individual shares are at least 1% are shown; the omitted bins contain 1,961 episodes (3.86%).

- *Single candidate*: “Is this the target {category}?”
- *Multiple unseen candidates*: “Which of these {category} instances is the target?”
- *Partially eliminated candidates*: “After ruling out {category} instances {eliminated_instance_ids}, which of the remaining {category} instances is the target?” When only one candidate remains, we instead ask, “Is {remaining_instance_id} the target after ruling out {eliminated_instance_ids}?”

The automated oracle resolves each referenced candidate using Habitat’s ground-truth instance-segmentation mask and compares its simulator instance ID with the annotated target ID. It answers a binary question with **Yes** or **No**. For a multiple-candidate question, it returns the target’s referenced ID when present and **None of them** otherwise. Ground-truth segmentation is used only to generate the answer; the answer exposes no privileged mask, coordinate, pose, distance, or directional information. The question–answer pair is then appended to the dialogue history.

Figure 6 summarizes the IIGN interaction-supervision pool before final trajectory selection. It contains 50,855 candidate episodes and 118,309 question–answer pairs; approximately 40K IIGN trajectories from this pool are included in the final training mixture.

The 15 most frequent categories account for 50.97% of episodes, while the remaining 147 categories form a long tail. The most frequent individual category accounts for only 5.26%, indicating that the corpus is not dominated by a single object class. Candidate configurations are more concentrated: single-candidate questions constitute 59.47% of all questions, and cases with one candidate remaining after partial elimination contribute another 24.77%. Thus, 84.24% of interactions are binary confirmations. The remaining 15.76% involve multiple candidates or rejection of all visible candidates: 8.84% contain multiple previously unseen candidates, and 6.91% retain multiple candidates after partial elimination.

Dialogue length is similarly skewed toward concise interactions: 60.73% of episodes contain one clarification turn and 13.90% contain two, while the mean is 2.33 turns per episode. The retained multi-turn tail nevertheless exposes the model to repeated clarification and evidence updates rather than only isolated single-turn confirmations.

D Evaluation Protocol

D.1 Habitat Agent Configuration

We use the same simulated embodiment and action space for all three benchmarks: HM3D ObjectNav, VL-LN, and CoIN. In Habitat, the robot is modeled as a cylindrical agent with a height of 1.25 m and a radius of 0.18 m. It is equipped with co-located RGB and depth sensors mounted at 1.25 m. Both sensors produce 640×480 observations with a 79° horizontal field of view, and the depth sensor has a measurement range of 0–10 m. GPS and compass sensors are also enabled in the task configuration.

The discrete action space allows the agent to stop, move forward, turn left or right, and look up or down. A forward action moves the agent by 0.25 m, a turning action changes its heading by 30° , and a look action changes the camera elevation by 30° . Sliding along obstacles is enabled, and each episode is limited to 500 simulator steps. An episode is successful when the agent issues **STOP** at a robot-to-target Euclidean distance of at most 0.25 m. Any instance of the instructed category is valid in ObjectNav; only the annotated target instance is valid in IIGN.

D.2 Baseline Selection and Adaptation

We select five representative baselines that cover the two dominant paradigms in goal-oriented navigation: modular systems and end-to-end learned policies. The modular baselines isolate the effects of geometric exploration, semantic guidance, and explicit interaction, whereas the end-to-end baselines provide direct comparisons with learned goal-oriented policies and navigation foundation models. All results are reproduced under the common Habitat configuration described in Appendix D.1, rather than quoted from the original papers.

Modular methods. FBE is a geometry-only frontier-based exploration baseline derived from the VLFM implementation [36]. We remove VLFM’s language-grounded value map while retaining its mapping, target-detection, and local-navigation components. Frontier selection is therefore based only on geometric exploration cues, providing a language-agnostic reference for measuring the benefit of semantic guidance. VLFM [36] is a representative zero-shot modular ObjectNav method. It builds an occupancy map from depth observations, extracts frontiers at the boundary between explored and unexplored space, and scores the observed regions against the target category with a pre-trained vision-language model. The resulting language-grounded value map guides the agent toward the most promising frontier, while a pre-trained detector and local policy handle target localization and motion.

AIUTA [24] complements these exploration-oriented baselines with an explicit interaction mechanism. It is a training-free modular method that operates independently of the underlying navigation policy. Its Self-Questioner uses LLM–VLM self-dialogue and uncertainty estimation to refine the description of a candidate instance, after which an Interaction Trigger decides whether to ask the user, continue navigation, or stop. We include AIUTA as a representative modular baseline for interaction-driven instance disambiguation in IIGN.

End-to-end methods. MTU3D [44] is designed directly for goal-oriented embodied navigation. It constructs an online query-based spatial memory from RGB-D observations and jointly represents grounded objects and unexplored frontiers, thereby coupling visual grounding, frontier selection, and trajectory prediction in a unified learned framework. Because its original model already supports category-, language-, and image-specified navigation goals, we evaluate the released pretrained weights directly without task-specific fine-tuning.

InternVLA-N1 [25] is an open navigation foundation model and serves as the strongest learned baseline in our comparison. It adopts a dual-system architecture in which a vision-language System 2 predicts mid-term pixel goals or latent plans and a diffusion-based System 1 executes short-horizon motion. The released model, however, is primarily trained for vision-and-language navigation and does not natively cover category-level ObjectNav or interactive instance disambiguation. We therefore initialize its System 2 from the official weights and fine-tune only this high-level component with our ObjectNav and IIGN navigation data, aligning its goal inputs and outputs with the two task formulations. In simulation, we do not use InternVLA-N1’s learned System 1; instead, its predicted pixel goals are executed by Habitat’s shortest-path follower.

For evaluation, every method uses the same sensor configuration, robot embodiment, episode budget, and

success threshold, while consuming the subset of RGB-D inputs required by its native architecture. For the direct comparison between InternVLA-N1 and our method, InternVLA-N1’s System 2 and GG-Nav serve as the respective high-level decision modules, while both share the same Habitat shortest-path follower as System 1. This controlled low-level execution isolates differences in high-level goal-oriented reasoning and decision making. On IIGN, the non-interactive baselines receive the same underspecified initial instruction but cannot ask clarification questions; success is awarded only for reaching the annotated target instance rather than an arbitrary instance of the same category. This setting quantifies how far standard ObjectNav policies can progress without interaction or explicit instance disambiguation. AIUTA and InternVLA-N1 are allowed to query the same interaction oracle, thereby holding oracle access fixed across the interactive methods.

D.3 Stage-wise Failure Attribution

Each failed episode is assigned to one stage: exploration, grounding, tracking, reaching, or stopping. Attribution is fully automated from the simulator’s ground-truth instance-segmentation masks, the model’s grounding/tracking records, and the final robot pose; no manual labels are used. Let d_{final} denote the straight-line robot-to-target distance at termination. We apply the following rules in order, assigning each failed episode to the first rule that it satisfies:

1. **Exploration:** the target’s ground-truth instance ID is absent from every instance-segmentation observation; equivalently, the target never enters the camera view.
2. **Grounding:** the target appears in at least one observation, but none of the model’s grounding records is associated with its ground-truth instance ID.
3. **Tracking:** the target is correctly grounded at least once, but its established track ID is not recovered when the target next appears, or that ID is subsequently assigned to a different instance.
4. **Reaching:** the correct target association is established and retained, but $d_{\text{final}} \geq 1\text{ m}$; the target neighborhood is defined as the set of robot poses less than 1 m from the target.
5. **Stopping:** the final stopping position is inside the target neighborhood but outside the success region, i.e., $0.25 < d_{\text{final}} < 1\text{ m}$.

E Real-world Experiment Details

E.1 System Architecture and Execution

Our platform is a Unitree Go2 equipped with an Intel RealSense D455 RGB-D camera mounted at 0.7 m and tilted downward by 15° . A ROS 2 client time-synchronizes the 640×480 RGB and aligned-depth streams with a 0.1 s tolerance and associates each observation with the nearest odometry sample. Each RGB-D observation is sent over HTTP to GG-Nav on a remote RTX 4090 server, while low-level control runs onboard the robot.

For each high-level decision, GG-Nav uses eight uniformly sampled historical frames and all dialogue frames. It also samples at most seven observations collected since the preceding decision, first grounding and tracking instances in these intermediate frames and then deciding from this evidence and the current frame. The model emits a discrete action, a pixel goal, or a clarification question. Movement commands are returned to the robot client, whereas an IIGN question is handled on the server: it explicitly references the current or an intermediate frame, the corresponding image is displayed with its grounding boxes and instance IDs, and the human answer is attached to that frame in the subsequent history. Because the robot uses a fixed downward-tilted camera, a look-down request triggers a second inference stage on the fixed camera stream rather than a physical camera motion.

For a pixel goal, the server denormalizes the predicted coordinate to the 640×480 image, back-projects it using aligned depth and camera intrinsics, compensates for camera pitch, and limits the resulting robot-frame waypoint to 1 m. The client transforms this local waypoint into the world frame using the time-aligned odometry and tracks it with a receding-horizon MPC controller (20 steps at 0.1 s, with $|v| \leq 0.3\text{ m/s}$ and $|w| \leq 0.3\text{ rad/s}$). Discrete actions are converted into pose increments from the current odometry—0.25 m forward, 30° left/right, or stop—and executed by a PID-mode pose controller with maximum linear and

angular speeds of 0.6 m/s and 0.5 rad/s, respectively. After a turn, the next server query is delayed until the yaw error is below 5° and angular speed below 0.08 rad/s, or until a 10 s timeout. Both controllers ultimately publish $(v, 0, w)$ commands through the Unitree Sport API.

E.2 Scenarios and Evaluation

We evaluate four real-world settings: ObjectNav and three IIGN difficulty levels. The IIGN levels are defined by the navigation distance and the number of same-category candidate instances requiring disambiguation:

- **Easy:** navigation distance < 50 m and fewer than five candidate instances.
- **Medium:** navigation distance < 50 m and at least five candidate instances.
- **Hard:** navigation distance > 50 m and at least five candidate instances.

An episode is successful only when the robot issues **STOP** within 0.25 m of the intended target instance while facing that instance. The final robot-to-target distance is measured manually in the laboratory, and the requirement that the robot face the target is verified by visual inspection at termination.

F Limitations and Future Work

First, the current interaction policy supports only target-specific clarification questions about observed candidate instances. Real-world interaction permits a broader and potentially more efficient set of information-seeking behaviors. For example, an agent could improve exploration by asking for route guidance, such as “How can I get to the target instance?”, or improve disambiguation through a compositional question that combines room context and nearby attributes, such as “Is the target sofa in the living room and next to a red decoration?” Future work will extend GG-Nav from candidate verification to open-ended question planning, allowing it to decide when to request navigational guidance and which combination of semantic and spatial cues would most effectively reduce target uncertainty.

Second, the current target vocabulary and object scale are constrained by the Habitat-based task setup and its available object annotations. Consequently, our experiments primarily cover relatively large, common household objects, such as beds, tables, and mirrors, and we do not yet claim reliable support for small everyday objects such as phones or forks. Extending GG-Nav to these targets will require simulation and real-world training data with substantially greater coverage of small objects, finer-grained instance annotations, and higher-resolution or active close-range perception for robust grounding and tracking.