Benchmarking Interaction, Beyond Policy: a Reproducible Benchmark for Collaborative Instance Object Navigation

Edoardo Zorzi[1,2]
edoardo.zorzi@uniroma1.it
Francesco Taioli[3,*]
francesco.taioli@polito.it
Yiming Wang[4]
ywang@fbk.eu
Marco Cristani[2,5]
marco.cristani@univr.it
Alessandro Farinelli[2]
alessandro.farinelli@univr.it
Alberto Castellini[2]
alberto.castellini@univr.it
Loris Bazzani[2]
loris.bazzani@univr.it

[1] Sapienza University of Rome, Rome, Italy
[2] University of Verona, Verona, Italy
[3] Polytechnic of Turin, Turin, Italy
[4] Fondazione Bruno Kessler, Trento, Italy
[5] Reykjavik University, Reykjavik, Iceland

*Work done before joining Amazon.

Abstract

We propose *Question-Asking Navigation* (QAsk-Nav), the first reproducible benchmark for *Collaborative Instance Object Navigation* (CoIN) that enables an explicit, separate assessment of embodied navigation and collaborative question-asking. CoIN requires an embodied agent to locate a target object specified through free-form natural language (e.g., "Find the messy red bed") using only egocentric visual observations and interactive natural-language dialogue with a human (e.g., "Does the red bed have brown pillows?"). The dialogue serves as an active mean to resolve ambiguities of visually-similar object instances in cluttered environments. Existing CoIN benchmarks focus primarily on navigation success and do not offer an independent evaluation of collaborative question asking. To address this limitation, QAsk-Nav provides (i) a lightweight *question-asking protocol* scored independently of navigation, (ii) an enhanced *navigation protocol* with realistic, diverse, high-quality target descriptions, and (iii) an open-source *dataset*, that includes around 26,000 quality-checked reasoning and question-asking traces for training, evaluation and analysis of interactive capabilities of CoIN models. Using the proposed QAsk-Nav benchmark, we develop Light-CoNav, a lightweight unified model for collaborative navigation that is 3× smaller and 70× faster than existing modular methods, while outperforming state-of-the-art CoIN approaches in generalization to unseen objects and environments. Code, benchmark, and dataset are available at https://benchmarking-interaction.github.io/.

© 2026. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

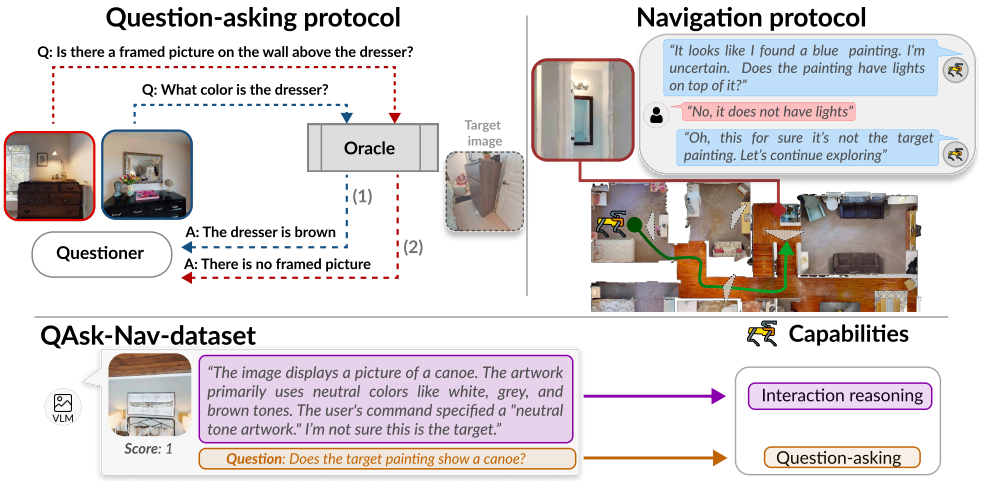


Figure 1: QAsk-Nav introduces two distinct protocols for question asking (top-left) and for navigation (top-right), supported by a novel dataset with structured reasoning, and question annotations (bottom).

1 Introduction

Open-vocabulary Instance Object Navigation (ION) [15, 22, 26, 45] requires an embodied agent to navigate complex environments to reach object instances specified via human-defined free-form natural language instructions. In real-world scenes, where multiple visually and semantically similar object instances are present (*e.g.*, multiple desk chairs *v.s.* dining chairs), ION becomes particularly challenging as the agent needs to disambiguate the target instance from *distractors* that are visually or semantically similar instances [15, 43, 53, 54]. An emerging paradigm in ION shifts focus toward *human-agent collaboration* [14, 45], introducing Collaborative instance Object Navigation (CoIN). This task necessitates agent-human interaction reasoning and uncertainty evaluation to determine the optimal timing and content of questions to the human, to resolve environmental or linguistic ambiguity, in order to find a specific instance of an object (e.g., "The messy red bed") in a cluttered environment among other distractors (e.g., "The brown bed" or "The tidy red bed").

Yet, the interactive nature of CoIN presents unique benchmarking challenges due to the intrinsic entanglement between navigation and interaction. While the first benchmark in [45] introduces the collaborative settings to existing simulation frameworks, the human-agent interaction is integrated within the navigation episodes during evaluation, leading to four main limitations: (i) the task performance is entangled with navigation policies, thus cannot isolate and measure solely the performance on human-agent interaction reasoning; (ii) the evaluation protocol is time-consuming as it requires the execution of the full navigation episode, introducing significant latency with up to 5 minutes per episode even the most modern embodied simulators [17, 20, 40]; (iii) the benchmark is not reproducible because the human responses are simulated and can differ at each run; (iv) goal instructions generated via early-generation Vision-Language Models (VLMs) often exhibit insufficient semantic richness, diversity and accuracy, limiting their utility for complex navigation tasks.

To address these limitations, we introduce *Question-Asking Navigation* (QAsk-Nav), the

first benchmark to support the development and evaluation of collaborative navigation agents in a *disentangled*, *lightweight* and *reproducible* manner, for both navigation and question-asking capabilities. As shown in Fig. 1, QAsk-Nav proposes three crucial elements: (i) a novel *question-asking evaluation protocol* that is disentangled from the navigation episodes, enabling fast and reproducible evaluation of interaction reasoning capabilities; (ii) an improved *navigation evaluation protocol* built on top of prior datasets [15, 45] with diverse, high-quality goal descriptions simulating realistic user commands, that greatly impacts the navigation performance of an embodied agent; (iii) a novel open-source *dataset* (QAsk-Nav-dataset) featuring a collection of 26,000 high-quality reasoning and interaction traces generated by leveraging modern large VLMs [2, 3, 36, 50] and text-guided image editing models [6, 47] following the question-asking protocol. Building on this benchmark, we introduce a *light-weight collaborative navigation agent* (Light-CoNav), a small VLM fine-tuned with QAsk-Nav-dataset for interaction reasoning on the CoIN task. Light-CoNav is substantially smaller (about 3× more compact), faster (by more than 70×), and more interpretable (with much smaller reasoning traces by around 10×) than prior state-of-the-art interaction models, while outperforming them both on navigation and question-asking tasks when evaluated on QAsk-Nav.

In summary, our main contributions are the following:

1. We propose a novel question-asking protocol disentangled from navigation, enabling lightweight, reproducible evaluation.
2. We enhance the navigation protocol with realistic, high-quality task descriptions that improve embodied agent navigation performance.
3. We create an open-source QAsk-Nav-dataset with high-quality reasoning and question-asking traces to develop CoIN agents with interactive capability.
4. We introduce Light-CoNav, a lightweight model for interaction reasoning on the CoIN task, outperforming complex modular architectures.

2 Related Work

Embodied agents have been evaluated on a variety of navigation tasks. In *object navigation* [1, 8, 15, 30, 32, 51, 54], the agent is tasked with reaching any object belonging to a given category. In contrast, *instance object navigation* [4, 15, 22, 26, 45], requires the agent to reach a specific object instance described in natural language. Similarly, *instance image navigation* [15, 18, 19, 43] challenges the agent to locate a target specified via an image. CoIN, as introduced in [45], extends these tasks with an agent collaborating with the human.

Benchmarks. Several benchmarks have been recently proposed for all these navigation tasks. OVON [54] uses realistic 3D scans from HM3D [37] to build a large object navigation benchmark with the goal of fixing the poor generalization issues of models trained on synthetically generated scenes [52]. GOAT-Bench [15] offers a comprehensive benchmark featuring three types of goal specifications: target images, object categories, and natural-language descriptions. PIN [4] and CoIN-Bench [45] focuses on testing navigation towards specific instances of objects in photo-realistic scenarios [37], including also distractor objects [17, 40]. Among these, CoIN-Bench [45] is the only benchmark that focuses on human-agent collaboration. However, prior works fail to decouple navigation from question-answering evaluation, and their reliance on simulator-heavy navigation episodes [40] introduces significant computational overhead. QAsk-Nav is the first to support the development and evaluation of collaborative navigation agents in a *disentangled*, *lightweight* and *repro-*

ducible manner, for both navigation and question-asking capabilities.

Interactive Navigation. Early work explored dialogue-guided navigation, where agents rely on human-provided guidance to complete tasks. For example, [12, 39, 48] introduce dialogue-based navigation benchmarks built from human-annotated interactions. Many existing approaches restrict interaction to predefined formats [27], limited communication channels [31, 34], or not free-form dialogue [13, 41, 42, 59]. TEACh, [35], proposes a dataset of human–to-human interactive dialogues for household tasks, while we introduce a framework/dataset for interactive QA during navigation. [38], while similar, focuses on a slightly different task and on developing a model; it does not provide a dataset and a full interaction evaluation environment. Ask-to-Clarify, [24], develops a similar disambiguation framework; however, unlike our work, it works by disambiguating instructions only at the start of an episode, while ours can do at different times, and entangles the QA module with a diffusion action module. [7] mainly focuses on developing a multimodal agent that can also act; we not only develop a model but also release a benchmark. Other works impose structural constraints on when or how interaction occurs: HANNA [33] allows interaction only at predefined locations, while in [57] clarification decisions rely on manually annotated disambiguation intents. More recent work explores richer interaction settings: ZIPON [8] studies personalized object retrieval, but the interaction is limited to ground-truth annotations. DialNav [14] explores a setting in which a navigator reaches a goal region with assistance from a remote guide, who must infer the navigator's position. AIUTA [45] introduces a free-form modular framework that focuses on human–agent interaction, combining VLMs and LLMs. However, AIUTA suffers from high latency due to the iterative procedure of its question-asking module, making real-world deployment challenging. We introduce Light-CoNav, a unified model that is substantially smaller, faster, and simpler than previous methods, demonstrating the complexities of previous modular approaches can be reduced.

3 The QAsk-Nav Benchmark

In this section, we will first provide the definition of the CoIN task (Sec. 3.1), followed by our proposed QAsk-Nav benchmark. Specifically, QAsk-Nav introduces three elements: (i) a question-asking evaluation protocol that is disentangled from the navigation episode (Sec. 3.2), (ii) an improved navigation evaluation protocol with high-quality task descriptions simulating realistic user commands (Sec. 3.3), and (iii) a modern dataset (QAsk-Nav-dataset) featuring a collection of diverse and high-quality reasoning and interaction traces (Sec. 3.4).

3.1 Task Definition

In the CoIN task as defined in [45], a navigation agent can collaborate with a user to reach a specific object instance expressed in free-form natural language, which can convey an arbitrary level of detail, *e.g.*, "the bed" or "the unmade bed near the open window". Human-agent interaction in CoIN is essential, as the agent must reason about visual observations and updated knowledge of the *target* to query the user. This allows it to effectively disambiguate the *target* from *distractors* that could be visually or semantically similar. More precisely, we define a *distractor* as an object instance belonging to the same semantic class as the target. For instance, if a specific bed is the target object in a navigation episode, any other bed in the environment is considered a distractor.



Figure 2: Examples from QAsk-Nav. Left: original image. Center: distractor with an altered sofa color and added mirror. Right: distractor with an altered plant and painting.

A CoIN navigation episode begins upon receiving a free-form natural language instruction describing the target D provided by the user. We assume the target object is present in the environment, which is continuous (not a discrete graph). At each time step t, the agent, parameterized by a policy π, receives an observation O_t and selects an action $a_t \in A =$ {`Forward 0.25m`, `Turn Right 15°`, `Turn Left 15°`, `Stop`, `Ask`}.. Navigation terminates when the agent selects the `Stop` action or when the maximum number of actions is reached. When the agent selects the `Ask` action, it generates a template-free, open-ended natural language question, denoted as $Q \in \mathcal{Q}$. The user's corresponding free-form natural language response, denoted $R \in \mathcal{R}$, is stored in the response set, forming the interaction context C, which can be leveraged to enable the agent to ask further relevant and disambiguating questions.

3.2 Question-asking Protocol

To decouple the interactive question-asking effectiveness from a CoIN navigation episode, we design a reinforcement learning-style [44] protocol (Fig. 1, top-left) that focuses solely on human-agent interaction inspired by [23]. The goal is to evaluate the agent's capability in asking clarifying questions about a target object when observing a candidate instance, given the initial description D, without running a full CoIN episode. Our protocol aims to mimic the situation during a CoIN episode where the agent faces a candidate object and needs to ask a question to the user. Specifically, we propose to have a *questioner* role representing the navigation agent who asks questions, and an *oracle* role, simulating the user answering the agent's questions (see Fig. 1 top-left). The oracle knows how the target object looks like since it has access to the image of the goal (not available to the questioner). The questioner is provided with the natural-language description D of the target object. To test multi-turn question-asking behavior for visual disambiguation, the protocol provides multiple images of distractors belonging to the same class as the target object. The distractors are presented to the questioner at each turn, who must determine whether they correspond to the target, or ask a clarifying question to the oracle. In the following, we describe in detail how the different components of the protocol are used to obtain a realistic benchmark.

Distractor Synthesis. A naive approach to generating distractors is to directly source observations from navigation episodes. However, due to the limited diversity of present instances, such an approach does not create challenging observations of distractors. We therefore opt to a generative way to generate diverse and challenging distractors exploiting advances in text-

guided image editing model, *i.e.*, `Gemini-3-pro-image` [11]. We source images from existing benchmarks, CoIN-Bench [45] and GOAT-Bench [15], and perform image editing of fine-grained attributes of the objects that are essential to uniquely identify an instance, including *color*, *salient feature*, and *contextual* elements. For each semantic category, we select 10 base images and obtain 5 variants of each by modifying these attributes, creating sets of images that differ from the original image in fine-grained details. This creates challenging, yet realistic scenarios where the distractors are very similar to the target (see Fig. 2). In Fig 8 (*Supp. Mat.*), we show (top) the distribution of the distractor categories (top 30), per split. We see that the categories are well distributed, especially in the train split, with most categories having at least 16-24 episodes (see paragraph *Interactive Episode*). This creates a balanced and well-rounded testing ground. To ensure data quality, we manually inspected around 75% of generated distractors for realism and diversity, discarding low-quality samples. More details in Sec 9 of *Supp. Mat.*

Target Descriptions. In addition to distractors, we synthetically generate the descriptions of the target object instances D. Different from prior benchmark [45], which only provide a brief description D, we aim to facilitate the study on how the specificity of the description influences the question-asking capabilities of the questioners and the navigation difficulty. To this end, we create six levels of descriptions with increasing specificity, ranging from minimal category-level cues, to highly detailed descriptions: (i) *category* (*e.g.*, 'bed'); (ii) *category and color* (*e.g.*, 'blue and dark brown bed'); (iii) *color-feature* which includes category, color, and a salient feature such as texture of the object (*e.g.*, 'blue patchwork bed'); (iv) *context*, which includes category plus surrounding context (*e.g.*, 'bed next to a wooden nightstand'); (v) *color-context*, that includes category, color, and context (*e.g.*, 'blue and dark brown bed next to a wooden nightstand'); (vi) *color-context-feature*, that includes category with color, context, and feature (*e.g.*, 'blue patchwork bed near a wooden nightstand'). We provide qualitative examples in Sec. 9 of the *Supp. Mat.*

Interactive Episode. Our protocol incorporates multi-turn episodes, featuring the persistent interaction required for effective human-agent collaboration in real-world scenarios. We construct the interactive episodes with the generated diverse distractor images and instructions D of multi-level specificity. Specifically, each episode incorporates the following data: (i) a target image, which represents the target instance; (ii) a natural-language description of the target; (iii) a sequence of distractors of the same category. In total, we collect data for $\sim$1400 episodes, which we split into a training and test set, comprising, respectively, data for $\sim$1,000 and $\sim$400 episodes (details in *Supp. Mat.* Sec. 9). As shown in Fig. 3, for each episode, the oracle is given access to the image of the target object instance, while the questioner has only access to the description D. We configure the episode to show distractors consecutively and finish with the target instance. At each step, the questioner is presented with an observation image, and it may either ask a question about the target or output a binary decision indicating whether the current observation matches the target. The episode continues if the agent correctly decides if the observation belongs to a distractor; otherwise, the episode terminates. Once the questioner outputs a decision for the final observation, the episode also terminates. As the episode always starts with distractors followed by the target, the agent may shortcut the task by always returning a negative decision. To ensure fairness, we randomize the number of observations per episode (with more than 4 candidates on average — Fig. 8, *Supp. Mat.*, bottom); we also measure the finish rate (`FR`), the ratio of completed episodes (requiring all correct decisions) (Sec. 5). The shortcut strategy would result in a `FR` of 0.

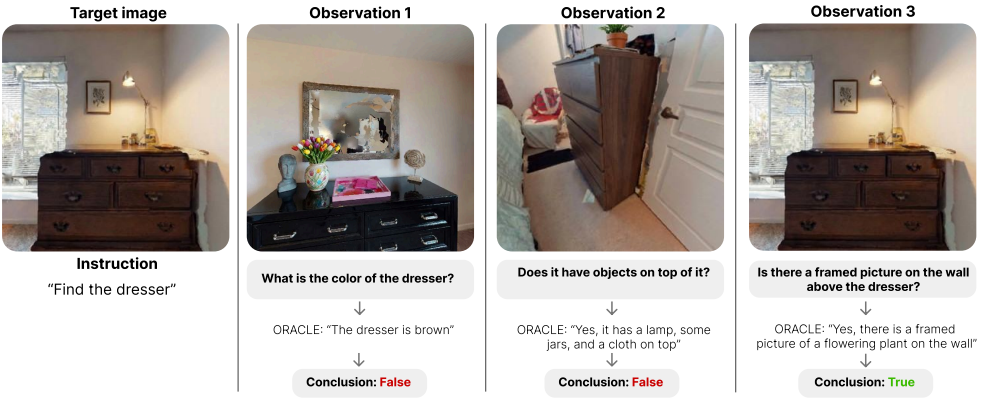


Figure 3: Episode from QAsk-Nav. Column 1: the target image available only to the oracle and the navigation instructions provided to the agent. Columns 2-3: different observations, including image, question, and answer from the oracle. Before each question and conclusion, the model produces a reasoning trace, motivating the decision. More detailed examples in *Supp. Mat.* Sec. 9.

3.3 Navigation Protocol

To evaluate the navigation capabilities of different models (Fig. 1, top-right), we propose a navigation protocol based on CoIN-Bench [45]. CoIN-Bench comprises ∼1,600 navigation episodes, picked from GOAT-Bench [15], based on the presence of distractor objects. As in GOAT-Bench, CoIN-Bench uses real-world 3D scene scans [37] within Habitat [40].

CoIN requires task instructions, *i.e.*, the description of an instance object to initiate the navigation episodes. However, constrained by the adopted VLM (`LLaVA 1.6 Mistral 7B` [25]), the quality of the instance description in CoIN-Bench is of limited quality, with wrong or hallucinated attributes (see Fig. 4). Moreover, to simulate the human user that provides responses to the navigation agent, CoIN-Bench leverages the same VLM `LLaVA 1.6 Mistral 7B` as the oracle, leading to limited interaction quality. To overcome these limitations, we instead leverage a commercial VLM (`Gemini-3-pro` [10]) to generate high-quality descriptions for each target object, and use a more advanced and larger VLM (`Qwen3-VL-30B-A3B` [2]) as an oracle that is integrated within the Habitat simulator [40]. In the following, we describe in detail the data split, and the target descriptions, and the episode execution during evaluation.

Data split. We follow the evaluation splits in prior works [15, 45]: *Val Seen*, *Val Seen Synonyms*, and *Val Unseen*. *Val Seen* contains the same objects and categories as the *Train* split, but different episodes (starting locations and distractors), the *Val Seen Synonyms* split uses the same object categories but different descriptions (using synonyms), to test adaptation to lexical variations, whereas the *Unseen* split, instead, contains completely novel objects. They comprise, respectively, 831, 359, and 459 episodes, for a total of 1,649 episodes (more information in the *Supp. Mat.*, Sec. 10).

Target Descriptions. We leverage commercial tools such as `Gemini-3-pro` [10] to ensure a faithful and complete description of each target object. We ensure the description quality with manual inspection using a subsampled set of descriptions, where human annotators are tasked to choose preferred, and to note hallucinations and imprecise details. Addition-

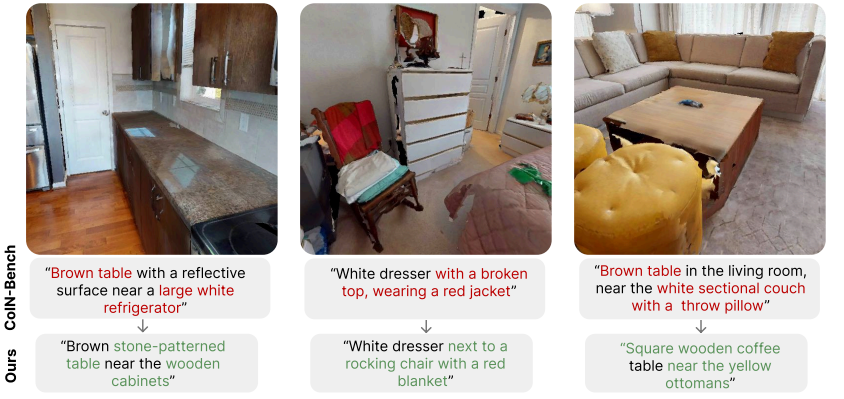


Figure 4: **Updated Task Annotations**. Comparison between CoIN-Bench [45] annotations and our QAsk-Nav annotations. Old annotations often contain confusing, nonsensical, or hallucinated descriptions, while new annotations are clearer and more accurate.

ally, we evaluate on the full set using a large language model (LLM) as a judge [16, 29, 58]. Our descriptions are preferred in more than 80% of the cases, with very few inaccurate details (see Sec. 5.2). This confirms that our descriptions are better in terms of precision and clarity, with reduced hallucinations. On average, our descriptions are shorter than those in CoIN-Bench: 9.66 words (std 1.81) compared to 17.53 words (std 9.46) in CoIN-Bench. Note that we use the type *(vi)* task descriptions in the navigation protocol (see Sec. 3.2).

Navigation Episode. The navigation agent starts from a random position in the environment, and is provided with a task description D of a target. The oracle is initialized with access to the target image. The agent navigates in the environment, taking, at each time-step t, an action a_t. When $a_t = \texttt{Ask}$, the question is fed to the oracle, which faithfully responds to it using the question Q contained in a_t, and the target image. A navigation is successful if the agent finds the target instance within the maximum number of steps.

3.4 QAsk-Nav-dataset

Due to the interactive nature in CoIN, it is challenging to create dataset of sufficient scale to support model training to enhance the interaction reasoning in such collaborative setting. In fact, prior work only addressed interaction reasoning at inference time. To bridge this critical gap, we propose QAsk-Nav-dataset (Fig. 1, bottom), the first dataset designed for training CoIN models to improve their reasoning and question-asking capabilities. The dataset is built by combining our protocols with modern generative models [2, 3, 11, 46, 47, 50].

QAsk-Nav-dataset contains high-quality reasoning traces together with their uncertainty scores and questions. Each sample of QAsk-Nav-dataset is a tuple (D, O, R, S, Q, C). D is a description of the target object instance of category c. O is a visual observation of an object, featuring either distractors or the target. R is a reasoning trace based on O and D. $S \in \{0, 1, 2\}$ is an uncertainty score, indicating the degree of the match of O with D, as motivated by the reasoning R. $S = 0$ means that the model is certain that O is not the target object, $S = 1$ means the model is unsure about the match, and $S = 2$ means that the model is certain that the observed object corresponds to the target object. Q is a question if $S = 1$, None otherwise. C is the context containing previous question-answer pairs that can be used

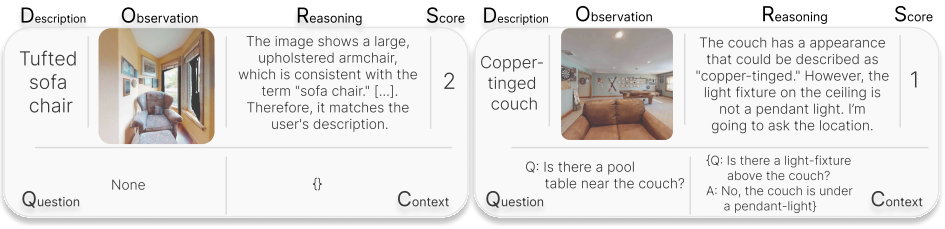


Figure 5: Two samples (D, O, R, S, Q, C) from QAsk-Nav-dataset. In the first example, the description D matches the observation O and thus the reasoning R and score S indicate a match (between D and O). In the second example, the description is ambiguous, and previous answers did not help: therefore R and S express uncertainty.

to inform what to ask. In total, it comprises approximately 26,000 reasoning traces, split into three partitions (following the standard protocol adopted by other benchmarks [15, 45]): *Train* (12.7k), *Val seen* (6.35k), and *Val unseen* (4.97k). See Fig. 5 for two examples.

We generate these samples starting from images O collected from our navigation protocol (target objects and distractor images), and from other navigation benchmarks, such as GOAT-Bench [15], and a mix of manually-curated and generated descriptions D. For each (D, O) pair, we prompt a commercial LLM (`Gemini-3-pro` [10]) to determine whether the description D matches with the observation O, that is, whether O is an observation of the target object described by D. We prompt it to generate a reasoning R and an uncertainty score S, plus a question Q if unsure. The final sample is created by merging manually verified (R, S, Q) and (D, O, C) with $C = \emptyset$. To increase sample diversity, we generate additional samples by running a large VLM as a questioner, by adopting our question-asking protocol, obtaining multiple reasoning traces and questions. From these rollouts (using the episode data from the training set), D is the starting description and O is the image distractor, while the reasoning R, the uncertainty score S and the question Q are obtained from the questioner's decision (the binary decisions, "not match" and "match", correspond to scores 0 and 2, respectively, and a question corresponds to a score of 1). When the navigation agent asks a question for the first time, the context is empty $C = \emptyset$, otherwise, the previous questions and answers are added to C. To ensure a quality dataset, we run a human analysis on thousands of samples, finding mostly high-quality reasoning traces, with an error rate below 0.5%. More details in Sec 11 of *Supp. Mat.*

4 Light-CoNav: A Light Collaborative Navigation Agent

Prior CoIN methods [14, 45] implement collaboration through multi-stage modular pipelines that combine multiple LLM and VLM components. In commonly used evaluations, interaction quality is mainly reflected through end-to-end navigation outcomes, making it difficult to isolate the contribution of each interaction module and often encouraging added complexity. Our benchmark, instead, provides separate protocols and metrics for navigation and question asking. Together with QAsk-Nav-dataset, it supports training a compact model and shows that strong collaborative performance can be achieved without complex modular architectures. We proved this by designing our navigation agent, called Light-CoNav, that is fine-tuned for improved reasoning and question-asking capabilities on QAsk-Nav-dataset.

Architecture and Navigation. Light-CoNav consists of a base VLM architecture composed of a single, lightweight, multimodal transformer ϕ [49], derived from `Qwen2.5-VL-7B-Instruct` [3], which layers on top of an existing navigation policy π[1]. The navigation policy π explores and navigates the environment, conditioned on the initial task command D. Once it observes an image O of an object belonging to the same category as the target object described by D, it passes the image to the VLM, which will output a uncertainty score S. Formally, $(R,S,Q) = \phi(p(D,O,C))$, with p the prompt that takes in input D, O and the context C (containing the previous questions and answers, empty at the start), output reasoning R, uncertainty score S and question Q. If the score is 2, it assigns this detection to the set of candidate objects, and π starts navigating towards it; if the score is 0, it discards this observation and π restarts the navigation; if the score is 1, it asks one or more questions Q to the user, possibly requesting more details and clarifications. In case of multiple questions, the questions and corresponding answers are added to the context C and passed to the model ϕ for the next interaction.

Training. We propose a two-stage training recipe to equip Light-CoNav with reasoning and question-answering capabilities. In the first stage, the base model is fine-tuned via *supervised fine-tuning* using *reasoning* samples from QAsk-Nav-dataset without questions and empty context C. This improves the model's ability to follow the given template and to produce high-quality reasoning traces and coherent uncertainty scores. Our tests showed that producing only a score S (without a reasoning trace R) decreases model performance. After this first learning phase, the model is capable of producing a reasoning and uncertainty score, *i.e.*, $(R,S,\emptyset) = \phi(p(D,O,\emptyset))$.

At this stage, the model has not yet been trained to generate questions. Therefore, we fine-tune it to acquire this skill by training on samples (D,O,R,S,Q,C) that include a question, interleaved with a small number of samples without questions (with a ratio of 10:1), to avoid excessively perturbing the model's output. Unlike before, the model is also trained with non-empty contexts C. After this phase, the model is also capable of producing questions Q, *i.e.*, $(R,S,Q) = \phi(p(D,O,C))$ (see *Supp. Mat.* Sec. 8.1)

5 Experimental Results

We conducted a thorough evaluation of the question-asking capabilities of various CoIN methods, following the protocol and episodes detailed in Sec. 3.2. Next, we assessed the navigation efficacy of the models using the evaluation framework defined in Sec. 3.3. The results are discussed in Sec. 5.1 and 5.2.

5.1 Results on Question-Asking

Baselines. To isolate the question-asking capabilities of CoIN methods, we compare the interactive module of AIUTA [45][2] and Light-CoNav, which we mark as AIUTA° and Light-CoNav°, respectively. In addition, we compare with the base `Qwen2.5-VL-7B` [3] to assess the impact of our fine-tuning and check the impact of the same number of parameters. Finally, we establish performance upper bounds using two significantly larger models,

[1] Light-CoNav performs only question asking to human, hence it can be used with many navigation policies as long as the policy is modular and employs an object detector. Here we use VLFM [53].

[2] To ensure reproducible offline evaluation, we replaced the original online LLM in the AIUTA self-questioner module with `Qwen3-13B` [3], providing similar performance with a total of 20B parameters.

`Qwen3-30B-VL-A3B` [2] and `Gemini-3-flash` [9], showing the capabilities of high-resource systems, typically inaccessible to navigation agents.

Metrics. We evaluate performance using: Success Rate (`SR` ↑), defined as the fraction of correct decisions per episode[3]; Finish Rate (`FR` ↑), the proportion of trajectories completed with all decisions correct; Number of Questions (`NQ` ↓), the average number of questions asked per observation; and Execution Time (`Time` ↓), the average time in seconds required to complete an episode. To mitigate potential biases inherent in individual VLMs, we employ a cross-model evaluation strategy using three distinct oracles: `Gemini-2.5-pro` [46], `Gemini-3-flash` [47], and `Qwen3-VL-30B-A3B-Instruct` [2]. Reported metrics represent the mean performance aggregated across these three oracles. Empirically, we did not observe significant differences between these models. Our recipe suggests that relatively large models with at least 30B parameters answer faithfully and similarly to the questions asked. We use the episode data from the test set, while each episode contains around four distractors on average (see Fig 8, bottom).

Results. Tab. 1 reports the comparison of different methods using the question-answering protocol of QAsk-Nav (for additional experiments, see Sec 7.2 of *Supp. Mat.*). We distinguish between navigation-ready models, those with the restricted parameters and memory efficiency required for robotic applications, and larger-scale, non-navigation-ready models. Light-CoNav° achieves a substantial improvement in `SR`, `FR`, and `NQ` compared to navigation-ready models, while reducing runtime. Notably, Light-CoNav° significantly outperforms both the base `Qwen2.5-VL-7B` [3] and state-of-the-art AIUTA°. This outcome underscores the effectiveness of our specialized training data. Despite being approximately 3× smaller in parameters, Light-CoNav° yields superior question-answering performance compared to the significantly larger AIUTA°. These results demonstrate that targeted supervision on QAsk-Nav-dataset effectively compensates for reduced model scale. When comparing our method with non-navigation-ready models of Tab. 1, we notice that Light-CoNav° outperforms `Qwen3-30B-VL-A3B` despite having 4× fewer parameters. We note that, while Light-CoNav° achieves far superior performance than all other methods, its `SR` reaches approximately 50% of that attained by `Gemini-3-flash` [9]. This performance gap highlights the potential for further scaling. In particular, `Gemini-3-flash` demonstrated significantly stronger contextual understanding, memory retention, and ability to ask salient follow-up questions grounded in prior interactions with the user. In contrast, smaller models performed substantially worse and did not consistently use previously provided contextual information correctly. Some failure cases include: repeatedly asking similar questions, requesting meaningless clarifications, and hallucinating about previous questions. This leads to incorrect predictions about the target and impedes the smaller model's ability to reach the correct goal even with the right questions. This highlights the need for better, smaller models. Rather than simply relying on parameter scaling, future improvements could leverage our dataset synthesis recipe to generate even more specialized training data, alongside architectural refinements, to bridge the margin between navigation-ready models and high-resource frontier models.

Next, we compare selected models on the six description types defined in the question-asking protocol (Section 3.2) in order to further analyze their behavior with increasing difficulty of navigation instructions. As shown in Tab. 2, Light-CoNav° consistently outperforms AIUTA° and `Qwen3-VL-30B-A3B` [2]—Qwen3-VL (30B)—in all settings except for the

[3] An episode is a complete rollout of the questioner, receiving sequential observations from the environment and asking questions about them.

Table 1: Performance comparison of methods on the QAsk-Nav benchmark. The top section shows navigation-ready models (parameter size under 20B), while the bottom section includes larger or closed-source baselines. The results are averaged across the different annotation types (see Sec. 3.2, *Target descriptions*)

Method	Navigation-ready	SR ↑	FR ↑	NQ ↓	Time ↓
Qwen2.5-VL-7B [8]	✔	0.0752	0.0034	2.7405	1.70
AIUTA° [15]	✔	0.3027	0.1993	1.1473	177.83
Light-CoNav° (ours)	✔	**0.4510**	**0.3317**	**0.1990**	**1.15**
Qwen3-VL-30B-A3B [8]	✗	0.3205	0.2755	**0.1174**	**1.58**
Gemini-3-flash [8]	✗	**0.9329**	**0.8943**	0.9564	13.20

Table 2: Success Rate (SR ↑) and Finish Rate (FR ↑) across different annotation types (see Sec. 3.2, *Target descriptions*) of QAsk-Nav.

	Cat		CatCol		ColFeat		Ctx		ColCtx		ColCtxFeat	
Method	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**
Qwen2.5-VL-7B[8]	0.02	0.0	0.04	0.0	0.13	0.01	0.05	0.01	0.08	0.0	0.13	0.01
AIUTA° [15]	**0.42**	**0.28**	0.34	0.22	0.30	0.19	0.32	0.22	0.24	0.15	0.20	0.13
Light-CoNav°	0.06	0.03	**0.45**	**0.31**	**0.58**	**0.44**	**0.43**	**0.32**	**0.61**	**0.42**	**0.68**	**0.54**
Qwen3-VL (30B) [8]	0.02	0.01	0.24	0.18	0.40	0.34	0.34	0.29	0.47	0.41	0.54	0.47
Gemini-3-flash [8]	**0.93**	**0.90**	**0.92**	**0.88**	**0.93**	**0.90**	**0.90**	**0.83**	**0.95**	**0.93**	**0.96**	**0.93**

category type, which is the least informative and most ambiguous description type. In the category case, Light-CoNav° and `Qwen3-VL-30B-A3B` tend immediate action, prioritizing direct decisions over proactive questioning. This behavior indicates that, under ambiguous category-level descriptions, models within this parameter range ($\leq$ 30B) tend to favor execution over information-seeking. These results suggest that current navigation-ready architectures have yet to fully optimize the threshold for uncertainty-driven communication, marking a clear opportunity for future work in robust ambiguity resolution. In all other description types, Light-CoNav° achieves substantially higher performance. This is particularly evident for the annotation color-context-feature, which most closely matches the annotations used in the navigation episodes. We also observe that Light-CoNav° performance increases monotonically with the amount of descriptive information (i.e., moving from the left to the right in Tab. 2), reaching its best results in the most detailed setting. In contrast, AIUTA° does not consistently benefit from richer descriptions and, in some cases, degrades as more detail is introduced. These results highlight the stronger visual-textual grounding and the more effective use of contextual information of Light-CoNav°.

In order to access the design decision to use VLMs as oracles, we compared them against human responses. Tab. 3 shows results in terms of SR and FR, while results are averaged across the three VLMs (first row) and across human annotators (second row). We observe that the questioner achieves comparable performance when interacting with both VLM oracles and human participants. This validates that the VLM oracle is a reliable proxy to evaluate an agent's question-asking capabilities to humans (more details in the *Supp. Mat.*).

5.2 Results on Navigation

We adopt here the evaluation protocol for navigation discussed in Sec. 3.3.

Baseline. We compare Light-CoNav against state-of-the-art ION baselines, including end-to-end trained policies such as SenseAct-NN [15] and OVON [54], as well as modular zero-

Table 3: Human evaluation study of the question-asking protocol. We use `Gemini-3-flash` [9] as a questioner, and compare its result using three VLM oracles and human annotators.

Oracle	*Cat*		*CatCol*		*ColFeat*		*Ctx*		*ColCtx*		ColCtxFeat	
Human	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**
✗	0.93	0.90	0.92	0.88	0.93	0.90	0.90	0.83	0.95	0.93	0.96	0.93
✔	0.92	0.88	0.90	0.85	0.92	0.87	0.90	0.86	0.96	0.94	0.94	0.87

Table 4: Results of the navigation task on the CoIN benchmark with our QAsk-Nav annotations.

Method	Features	*Val Seen*			*Val Seen Synonyms*			*Val Unseen*		
	Interaction	**SR ↑**	**SPL ↑**	**NQ ↓**	**SR ↑**	**SPL ↑**	**NQ ↓**	**SR ↑**	**SPL ↑**	**NQ ↓**
SenseAct-NN [15]	✗	6.62	3.11	-	13.09	6.45	-	0.22	0.05	-
PSL [43]	✗	8.78	3.30	-	8.91	2.83	-	4.58	1.39	-
OVON [52]	✗	8.18	5.24	-	**15.88**	**11.35**	-	2.61	1.29	-
VLFM [53]	✗	0.36	0.28	-	0.00	0.00	-	0.00	0.00	-
AIUTA [45]	✔	10.46	4.96	1.31	14.89	9.11	**1.48**	7.83	4.38	1.37
Light-CoNav	✔	**12.20**	**6.32**	**1.29**	14.15	8.29	1.65	**9.63**	**4.55**	**1.35**

shot approaches like PSL [43] and VLFM [53]. For CoIN, we compare against AIUTA [45], the current state-of-the-art modular zero-shot method and the most direct competitor to our approach. Extensive descriptions of these baselines in the *Supp. Mat.* (Sec. 7).

Metrics and Implementation Details. We evaluate using Success Rate (`SR` ↑), Success weighted by Path Length (`SPL` ↑), and average number of questions (`NQ` ↓). Following [45], an episode is considered successful if the agent reaches within $\tau = 0.25$m of the target within $T = 500$ steps. We use the continuous environment provided by Habitat 3.0 [40], based on HM3D scans [37]. During simulation, the user is simulated by a VLM, `Qwen3-VL-30B-A3B` [2], which has exclusive access to the target image. Both the target image and the agent's observations O_t are represented at a resolution of 512×512.

Results. Tab. 4 shows the performance of all the methods on the instance object navigation episodes (see Sec. 3.3), separated as in the original Goat-bench split, as *Val Seen*, *Val Seen Synonyms*, and *Val Unseen*. Light-CoNav significantly outperforms all methods, especially in the *Val Seen* and *Val Unseen* splits, while using an interaction model that 70× faster to generate questions and uses 3× less parameters as showed in Tab. 6 (Time and Size columns). Specifically, the improvement on the *Val Unseen* split is important because it shows better adaptability to unseen scenes and objects than other methods, which is a key capability for CoIN agents. Interestingly, all methods show an improved performance on the *Val Seen Synonyms* split. We posit that the main reason for this is the nature of the trajectories contained in the split. In fact, *Val Seen Synonyms* only contains three goal categories (*shelf*, *armchair*, *kitchen cabinet*) with the latter having almost double the amount of the trajectories of the other two combined (see *Supp. Mat.*, Sec 10.2). Moreover, most of the *kitchen cabinet* trajectories contain similar distractors, like similar kitchen cabinets clamped together, lowering the difficulty and increasing the rate of false positives (a distractor very close to the goal). We include this split for fair comparison with previous works, even with these limitations in mind. In terms of the number of questions (`NQ`), our method asks almost the same number of questions as AIUTA. Tab. 6 also shows the impact of the improvement in annotation quality on the navigation `SR` performance of both AIUTA and Light-CoNav, with both models

Table 5: Comparison between CoIN-Bench [45] and our task descriptions evaluated by LLM-as-judge (`Gemini-3-pro`) and humans (on a random 50% subset of the data). Our task descriptions are highly preferred and exhibit a very low rate of problematic descriptions concerning hallucination or imprecise details (2-5%). In contrast, the CoIN-Bench descriptions are rarely preferred with a high problem rate (over 50%).

Evaluator	Task Descriptions	Preference Rate ↑	Problem Rate ↓
LLM as judge	CoIN-Bench	0.171	0.618
	Ours	**0.829**	**0.020**
Humans	CoIN-Bench	0.183	0.530
	Ours	**0.816**	**0.056**

Table 6: Impact of annotations quality, inference time, and parameter size. Our annotations improve the results across all splits. We report the `SR` ↑ across all splits, the average inference time during interaction (`Time` ↓), and model size (`Size` ↓).

Method	Ann. Type	*Val Seen*	*Val Seen Synonyms*	*Val Unseen*	Time ↓	Size ↓
AIUTA [45]	CoIN-Bench	7.42	14.38	6.67	185.79	20B
	Ours	**10.46** (1.41x)	**14.89** (1.04x)	**7.83** (1.17x)		
Light-CoNav	CoIN-Bench	9.14	11.64	6.43	**2.49**	**7B**
	Ours	**12.20** (1.33x)	**14.15** (1.33x)	**9.63** (1.50x)		

improving by around 1.30×. This motivates the adoption of the new annotations.

Quality of Our Decriptions. We assess here the quality of the new navigation descriptions against those provided by [45] (see Sec. 3.3, **Target Descriptions**). Tab. 5 shows the results in terms of SR, using a LLM-as-a-judge procedure (top 2 rows) and five human annotators (bottom two rows). For each human annotator, we sample a random 50% subset of the data. In both cases, we notice that our annotations are preferred over 80% of the time, while CoIN-Bench descriptions exhibit a high rate of problematic descriptions concerning imprecise details or hallucinations. Instead, the rate of problematic descriptions in our dataset is significantly lower, reducing from a very high value of 50-60% of COIN-Bench to only 2-5%. More details in *Supp. Mat.*

6 Conclusions

We introduced QAsk-Nav, the first benchmark for CoIN that proposes separate, standardized protocols to evaluates the agent's capability in navigation and question asking. We created QAsk-Nav-dataset, a large-scale collection of quality-checked reasoning and question-asking traces to support training, evaluation and analysis. The dataset will be made publicly available. Using this setting, we answered one concrete modeling design question, namely that strong collaborative behavior can be achieved without heavy modular pipelines, and that a lightweight interaction-reasoning model (like Light-CoNav) can be competitive while being substantially more efficient. More broadly, QAsk-Nav enables further model-design questions to be systematically studied: (i) How much explicit supervision from reasoning traces is actually needed to learn effective question asking? (ii) Does coupling question generation with calibrated uncertainty estimates improve ask-or-act decisions and robustness under noisy observations or missing detections?

Acknowledgments

We gratefully acknowledge Gianni Franchi for his valuable input and insightful discussions during the initial phases of the project. We are also extremely grateful to Google and LambdaLabs for providing computing credits through the Google Cloud Research Credits Program and the Lambda Research Grant Application. We also acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy). This work was also supported by the Ministero delle Imprese e del Made in Italy (IPCEI Cloud DM 27 giugno 2022 – IPCEI-CL-0000007) and European Union (Next Generation EU). Lastly, we acknowledge financial support from PNRR MUR project PE0000013-FAIR and PR Veneto FESR 2021-2027 24729- 002238 - SINERGHY.

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On Evaluation of Embodied Navigation Agents, 2018. URL https://arxiv.org/abs/1807.06757.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report, 2025. URL https://arxiv.org/abs/2511.21631.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report, 2025. URL https://arxiv.org/abs/2502.13923.

[4] Luca Barsellotti, Roberto Bigazzi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Personalized Instance-based Navigation Toward User-Specific Objects in Realistic Environments. In *NeurIPS*, 2024. URL https://openreview.net/forum?id=uKqn1Flsbp.

[5] 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, 2020. URL https://arxiv.org/abs/2006.13171.

[6] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning To Follow Image Editing Instructions. In *CVPR*, pages 18392–18402, June 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Brooks_InstructPix2Pix_Learning_To_Follow_Image_Editing_Instructions_CVPR_2023_paper.html.

[7] Xiaoyu Chen, Shenao Zhang, Pushi Zhang, Li Zhao, and Jianyu Chen. Asking before action: Gather information in embodied decision making with language models. *CoRR*, abs/2305.15695, 2023. doi: 10.48550/ARXIV.2305.15695. URL https://doi.org/10.48550/arXiv.2305.15695.

[8] Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. Think, Act, and Ask: Open-World Interactive Personalized Robot Navigation. In *ICRA*, pages 3296–3303, 2024. doi: 10.1109/ICRA57147.2024.10610178.

[9] DeepMind. Gemini 3 flash model card, Dec 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf. Model card published by Google DeepMind.

[10] DeepMind. Gemini 3 pro image card, Dec 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf. Model card published by Google DeepMind.

[11] DeepMind. Gemini 3 pro image card, Dec 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Image-Model-Card.pdf. Model card published by Google DeepMind.

[12] Qiaozi Gao, Govind Thattai, Suhaila Shakiah, Xiaofeng Gao, Shreyas Pansare, Vasu Sharma, Gaurav Sukhatme, Hangjie Shi, Bofei Yang, Desheng Zhang, Lucy Hu, Karthika Arumugam, Shui Hu, Matthew Wen, Dinakar Guthy, Shunan Chung, Rohan Khanna, Osman Ipek, Leslie Ball, Kate Bland, Heather Rocker, Michael Johnston, Reza Ghanadan, Dilek Hakkani-Tur, and Prem Natarajan. Alexa Arena: A User-Centric Interactive Platform for Embodied AI. In *NeurIPS*, volume 36, pages 19170–19194, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3d0758f0b95e19abc68c1c8070d36510-Paper-Datasets_and_Benchmarks.pdf.

[13] Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S. Sukhatme. DialFRED: Dialogue-Enabled Agents for Embodied Instruction Following. *IEEE Robotics and Automation Letters*, 7(4):10049–10056, 2022. doi: 10.1109/LRA.2022.3193254.

[14] Leekyeung Han, Hyunji Min, Gyeom Hwangbo, Jonghyun Choi, and Paul Hongsuck Seo. DialNav: Multi-turn Dialog Navigation with a Remote Guide. In *ICCV*, pages 8514–8523, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Han_DialNav_Multi-turn_Dialog_Navigation_with_a_Remote_Guide_ICCV_2025_paper.html.

[15] Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A Benchmark for Multi-Modal Lifelong Navigation. In *CVPR*, pages 16373–16383, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Khanna_GOAT-Bench_A_Benchmark_for_Multi-Modal_Lifelong_Navigation_CVPR_2024_paper.html.

[16] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing Fine-Grained Evaluation Capability in Language Models. In *ICLR*, 2024. URL https://openreview.net/forum?id=8euJaTveKw.

[17] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI, 2022. URL https://arxiv.org/abs/1712.05474.

[18] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-Specific Image Goal Navigation: Training Embodied Agents to Find Object Instances, 2022. URL https://arxiv.org/abs/2211.15876.

[19] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to Objects Specified by Images. In *ICCV*, pages 10916–10925, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Krantz_Navigating_to_Objects_Specified_by_Images_ICCV_2023_paper.html.

[20] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. iGibson 2.0: Object-Centric Simulation for Robot Learning of Everyday Household Tasks. In *CoRL*, volume 164, pages 455–465. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/li22b.html.

[21] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[22] Weijie Li, Xinhang Song, Yubing Bai, Sixian Zhang, and Shuqiang Jiang. ION: Instance-level Object Navigation. In *ACM MM*, MM '21, page 4343–4352. Association for Computing Machinery, 2021. ISBN 9781450386517. doi: 10.1145/3474085.3475575. URL https://doi.org/10.1145/3474085.3475575.

[23] Yu-Sheng Li, Chien-Hui Tseng, Chian-Yun Huang, and Wei-Yun Ma. Guess What: A Question Answering Game via On-demand Knowledge Validation. In *IJCNLP*, pages 57–60, November 2017. URL https://aclanthology.org/I17-3015/.

[24] Xingyao Lin, Xinghao Zhu, Tianyi Lu, Sicheng Xie, Hui Zhang, Xipeng Qiu, Zuxuan Wu, and Yu-Gang Jiang. Ask-to-clarify: Resolving instruction ambiguity through multi-turn dialogue. *CoRR*, abs/2509.15061, 2025. doi: 10.48550/ARXIV.2509.15061. URL https://doi.org/10.48550/arXiv.2509.15061.

[25] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

[26] Qianyi Liu, Siqi Zhang, Yanyuan Qiao, Junyou Zhu, Xiang Li, Longteng Guo, Qunbo Wang, Xingjian He, Qi Wu, and Jing Liu. GroundingMate: Aiding Object Grounding for Goal-Oriented Vision-and-Language Navigation. In *WACV*, pages 1775–1784, February 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Liu_GroundingMate_Aiding_Object_Grounding_for_Goal-Oriented_Vision-and-Language_Navigation_WACV_2025_paper.html.

[27] Qianyi Liu, Siqi Zhang, Yanyuan Qiao, Junyou Zhu, Xiang Li, Longteng Guo, Qunbo Wang, Xingjian He, Qi Wu, and Jing Liu. GroundingMate: Aiding Object Grounding for Goal-Oriented Vision-and-Language Navigation. In *WACV*, pages 1775–1784, February 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Liu_GroundingMate_Aiding_Object_Grounding_for_Goal-Oriented_Vision-and-Language_Navigation_WACV_2025_paper.html.

[28] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv preprint arXiv:2303.05499*, 2023.

[29] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In *EMNLP*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

[30] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. ZSON: Zero-Shot Object-Goal Navigation using Multimodal Goal Embeddings. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/d0b8f0c8f79d3a621af945cafb669f4b-Abstract-Conference.html.

[31] Arjun Majumdar, Fei Xia, Brian Ichter, Dhruv Batra, and Leonidas Guibas. FindThis: Language-Driven Object Disambiguation in Indoor Environments. In *CoRL*, volume 229, pages 1335–1347. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/majumdar23a.html.

[32] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. THDA: Treasure Hunt Data Augmentation

for Semantic Navigation. In *ICCV*, pages 15354–15363. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01509. URL https://doi.org/10.1109/ICCV48922.2021.01509.

[33] Khanh Nguyen and Hal Daumé III. "Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning". In *EMNLP-IJCNLP*, pages 684–695. Association for Computational Linguistics, November 2019. doi: 10.18653/v1/D19-1063. URL https://aclanthology.org/D19-1063.

[34] Khanh X Nguyen, Yonatan Bisk, and Hal Daumé Iii. A Framework for Learning to Request Rich and Contextually Useful Information from Humans. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 16553–16568. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nguyen22a.html.

[35] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gökhan Tür, and Dilek Hakkani-Tür. Teach: Task-driven embodied agents that chat. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 2017–2025. AAAI Press, 2022. doi: 10.1609/AAAI.V36I2.20097. URL https://doi.org/10.1609/aaai.v36i2.20097.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, volume 139, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[37] 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 *NeurIPS: Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=-v4OuqNs5P.

[38] Ram Ramrakhya, Matthew Chang, Xavier Puig, Ruta Desai, Zsolt Kira, and Roozbeh Mottaghi. Grounding multimodal llms to embodied agents that ask for help with reinforcement learning. *CoRR*, abs/2504.00907, 2025. doi: 10.48550/ARXIV.2504.00907. URL https://doi.org/10.48550/arXiv.2504.00907.

[39] Homero Roman Roman, Yonatan Bisk, Jesse Thomason, Asli Celikyilmaz, and Jianfeng Gao. RMM: A Recursive Mental Model for Dialogue Navigation. In *EMNLP Findings*, pages 1732–1745, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.157. URL https://aclanthology.org/2020.findings-emnlp.157/.

[40] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and

Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *ICCV*, October 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Savva_Habitat_A_Platform_for_Embodied_AI_Research_ICCV_2019_paper.html.

[41] Ying Shen, Daniel Bis, Cynthia Lu, and Ismini Lourentzou. ELBA: Learning by Asking for Embodied Visual Navigation and Task Completion. In *WACV*, pages 5177–5186, February 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Shen_ELBA_Learning_by_Asking_for_Embodied_Visual_Navigation_and_Task_WACV_2025_paper.html.

[42] Kunal Pratap Singh, Luca Weihs, Alvaro Herrasti, Jonghyun Choi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Ask4Help: Learning to Leverage an Expert for Embodied Tasks. In *NeurIPS*, volume 35, pages 16221–16232. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/674ad201bc8fa74b3c9979230aa0c63b-Paper-Conference.pdf.

[43] Xinyu Sun, Lizhao Liu, Hongyan Zhi, Ronghe Qiu, and Junwei Liang. Prioritized Semantic Learning for Zero-Shot Instance Navigation. In *ECCV*, pages 161–178, Cham, 2025. URL https://doi.org/10.1007/978-3-031-73254-6_10.

[44] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[45] 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 *ICCV*, pages 18781–18792, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Taioli_Collaborative_Instance_Object_Navigation_Leveraging_Uncertainty-Awareness_to_Minimize_Human-Agent_Dialogues_ICCV_2025_paper.html.

[46] Gemini Team. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. URL https://arxiv.org/abs/2507.06261.

[47] Gemini Team. Gemini: A Family of Highly Capable Multimodal Models, 2025. URL https://arxiv.org/abs/2312.11805.

[48] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-Dialog Navigation. In *CoRL*, volume 100, pages 394–406. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/thomason20a.html.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NeurIPS*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[50] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

[51] Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. MultiON: Benchmarking Semantic Map Memory using Multi-Object Navigation. In *NeurIPS*, volume 33, pages 9700–9712, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6e01383fd96a17ae51cc3e15447e7533-Paper.pdf.

[52] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin S Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander Clegg, John M Turner, Zsolt Kira, Manolis Savva, Angel X Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, and Chris Paxton. HomeRobot: Open-Vocabulary Mobile Manipulation. In *CoRL*, 2023. URL https://openreview.net/forum?id=b-cto-fetlz.

[53] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation. In *ICRA*, pages 42–48. IEEE, 2024. URL https://doi.org/10.1109/ICRA57147.2024.10610712.

[54] Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. HM3D-OVON: A Dataset and Benchmark for Open-Vocabulary Object Goal Navigation. In *IROS*, pages 5543–5550. IEEE, 2024. URL https://doi.org/10.1109/IROS58592.2024.10802709.

[55] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11941–11952. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01100. URL https://doi.org/10.1109/ICCV51070.2023.01100.

[56] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster Segment Anything: Towards Lightweight SAM for Mobile Applications. *arXiv preprint arXiv:2306.14289*, 2023.

[57] Michael JQ Zhang and Eunsol Choi. "Clarify When Necessary: Resolving Ambiguity Through Interaction with LMs". In *NAACL*, pages 5526–5543, April 2025.

ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.306. URL https://aclanthology.org/2025.findings-naacl.306/.

[58] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *NeurIPS*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

[59] Yi Zhu, Yue Weng, Fengda Zhu, Xiaodan Liang, Qixiang Ye, Yutong Lu, and Jianbin Jiao. Self-Motivated Communication Agent for Real-World Vision-Dialog Navigation. In *ICCV*, pages 1574–1583, 2021. doi: 10.1109/ICCV48922.2021.00162.

Supplementary material

This supplementary material provides additional analyses of our proposed benchmark, offering extended empirical evaluations, data quality validations, and detailed implementation information.

We begin by expanding on the experimental details and analyses presented in the main manuscript (Sec. 7). This includes a detailed breakdown of the navigation baselines (Sec. 7.1) and an extensive evaluation of question-asking protocols, specifically focusing on the performance gap between human oracles and VLMs (Sec. 7.2). Furthermore, we provide additional ablation studies to justify the necessity of fine-tuning (Sec. 8).

To validate the integrity of our benchmark, we provide a thorough assessment of our data generation and human-centric evaluations. We detail the question-asking protocol and visualize the synthesized imagery to demonstrate its high fidelity (Sec. 9). This is complemented by a user study centered on navigation episodes, which evaluates the quality and linguistic characteristics of our proposed descriptions (Sec. 10).

Moreover, to ensure full transparency and reproducibility, we describe the dataset collection process in more detail (Sec.11), and report the complete set of prompts used across all experiments (Sec. 12).

7 Experimental Results

7.1 Details of the Navigation Methods

In our navigation experiments, we experimented with the following state-of-the-art methods: SenseAct-NN [15], OVON [54], PSL [43], VLFM [53], and AIUTA [45]. The details of each of these methods are described below.

SenseAct-NN [15] introduces a recurrent policy trained end-to-end with Reinforcement Learning, integrating visual observations and textual descriptions via encoders from a pre-trained vision-language model (*i.e.*, CLIP [36]), alongside GPS and compass data.

OVON [54] is a transformer-based object-navigation policy that builds a large multidimensional state representation at each timestep by combining the current visual input, the language goal, and the previous action through two SigLIP encoders [55] and an action embedding layer.

PSL [43] proposes a zero-shot policy that is pre-trained on the different tasks of image navigation, then applied to object navigation, which also uses text, using a CLIP encoder and a particular semantic perception module.

VLFM [53] is a fully modular zero-shot policy that builds a semantic *frontier map*, assigning to each possible frontier a value indicating the likelihood of finding the object in that area using BLIP-2 [21]. For object detection, it uses Grounding-DINO [28] and Mobile SAM [56]. VLFM is the navigation module that we exploited in Light-CoNav given its practical simplicity and good navigation performance.

AIUTA [45] is the COIN method that is introduced together with COIN-Bench. AIUTA follows a modular zero-shot design, built on top of VLFM, using two large models (a VLM and an LLM) that dialogue with each other every time a potential target is detected to reduce uncertainty on visual understanding, and understand how to interact with the user under the collaborative setting. AIUTA is most related to Light-CoNav. Light-CoNav presents

Table 7: Human evaluation study of the question-asking protocol. We use `Gemini-3-flash` [9] as a questioner, and compare its result using three VLM oracles and human annotators.

Oracle	*Cat*		*CatCol*		*ColFeat*		*Ctx*		*ColCtx*		ColCtxFeat	
Human	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**	**SR↑**	**FR↑**
✗	0.93	0.90	0.92	0.88	0.93	0.90	0.90	0.83	0.95	0.93	0.96	0.93
✔	0.92	0.88	0.90	0.85	0.92	0.87	0.90	0.86	0.96	0.94	0.94	0.87

a smaller, lighter single-module question-asking VLM, that outperforms the larger, heavier multi-module AIUTA.

7.2 Human Evaluation and Results on Question-Asking

VLM-based *v.s.* Human Evaluation. For ease of reference, we report the same data as in Tab 3 in Tab 7. Our main evaluation protocol includes VLM-based oracles to make experimentation fast, scalable and reproducible. However, a fair question arises: *How does the model behave when a real human takes the role of the oracle?* In addition, we want to investigate whether the proposed scalable way to run the evaluation without costly human intervention is a viable and consistent way to test for question-asking capabilities, without introducing errors and biases in the tests. To answer these questions, we tested the state-of-the-art model `Gemini-3-flash` as a questioner, and have human annotators as oracles. Tab. 7 shows the results of the VLM-based oracle vs. the human oracle. Two annotators, by interacting with the system, generated 578 different trajectories, with an average of 2.8 answered questions per episode, for a total of more than 1.6K answers. Each annotator had the target image (not known to the questioner) and the question asked. We instructed each annotator to answer faithfully based only on these elements.

Under different target description types, we observe consistently similar results in terms of SR and FR obtained by both oracle types. Moreover, we find that, on average, a human response can take from 4 seconds up to 15, whereas a VLM takes as little as 2 seconds, with less variability. Moreover, using a VLM oracle allows us to run multiple evaluations in parallel, which further scales up the evaluation, which is not feasible with human evaluation. This demonstrates that our VLM-based oracle can serve as a viable alternative to simulate costly human responses, significantly scaling up experimentation and speeding up the process tenfold. Fig. 11 shows the tool used to collect the human answers.

On the other hand, `Gemini-3-flash` [9] exhibits superior question-asking capabilities across all types, reaching 96% on the most informative type (color-context-feature, last column). It is worth noting that this evaluation functions as an upper bound for CoIN, since `Gemini-3-flash` is estimated to be very large (order of 1 trillion parameters), thus making the model non-navigation-ready.

Additional evaluations The trajectories in our Question-asking protocol mimic the way distractors are seen in the navigation trajectories: the agent explores the environment, detecting objects in the same class as the target object, discarding observations matching distractors (i.e. objects similar to the target object, but with different characteristics); after detecting the target object, it navigates towards it. Therefore, usually, the target object's detection is always the last observation: thus, the trajectories in our Question-asking protocol always show the target (correct) image in the last position. This is a bias that could be exploited by the

Table 8: Additional evaluation on our QA benchmark, comparing trajectories with the target always in the last position (rand. pos., ✗), and new trajectories with shuffled positions (rand. pos., ✔)

Method	Rand. pos.	SR ↑	FR ↑	NQ ↓	Time ↓
Gemini-3-flash	✗	0.93	0.89	**0.96**	**13.2**
Gemini-3-flash	✔	**0.96**	**0.94**	1.99	31.6
Light-CoNav° (ours)	✗	0.45	0.33	0.20	**1.15**
Light-CoNav° (ours)	✔	**0.52**	**0.39**	0.20	3.99

questioner models. To avoid any 'shortcutting', we randomized the number of distractors for each trajectory. Therefore, the only way a questioner can exploit this system is by always returning *false*, which would incur in a success rate of $(n-1)/n$ for each trajectory, where n is the number of distractors contained in the trajectory. However, our main metric is the `Finish Rate`, which is only positive if *all the distractors and true target are detected* (i.e., such a strategy would incur a `Finish Rate` of 0). We can see, in the main text, that no model attains a `Finish Rate` of 0, therefore, such exploits do not happen. However, we still performed additional evaluations on 250 new validation trajectories in Tab. 8 to verify, again, that placing the true target as the final distractor does not introduce exploitable patterns or affect our conclusions. We shuffled distractor order so that the correct target can appear in any position; moreover, we included target-free trajectories containing only distractors to assess this additional scenario. Tab. 8 shows similar and slightly better results when comparing models with *randomized position* or without it. This indicates that our design choice does not impact the models' results.

8 Light-CoNav

8.1 Motivation of First Training Stage

During the first training stage that we described in the main paper, Light-CoNav learns the capability to perform reasoning R and score S for its uncertainty. We show here an experiment to motivate the need of this first training stage. In particular, Fig. 6 shows the results, in terms of *mean absolute error* between the score S from the model and the ground truth score, comparing our first-stage finetuned model as in the paper (`Sft`), a model finetuned using only S (`Sft no-reas.`), the pre-trained base model (`Base`), the base model prompted to return only S (`Base no-reas.`), a baseline always returning $S=0$ (`Fixed=0`), and a baseline picking scores based on their frequencies on the train set (`Freq.`). We observe that returning both R and S (`Sft`) achieves the best performance across all splits of QAsk-Nav-dataset and models after fine-tuning. Because the dataset is unbalanced, returning always 0 is achieves a good performance, beating the untuned model. However, finetuning is always better, even in unseen splits. This is particularly meaningful for the val unseen split, where categories and images are completely new.

Having evaluated the strength of the first training stage against possible alternatives, we proceed with the second phase, where we train Light-CoNav also on data points containing a question and a non-empty context. The results of the second training stage are reported in Tab. 1 of the main paper.

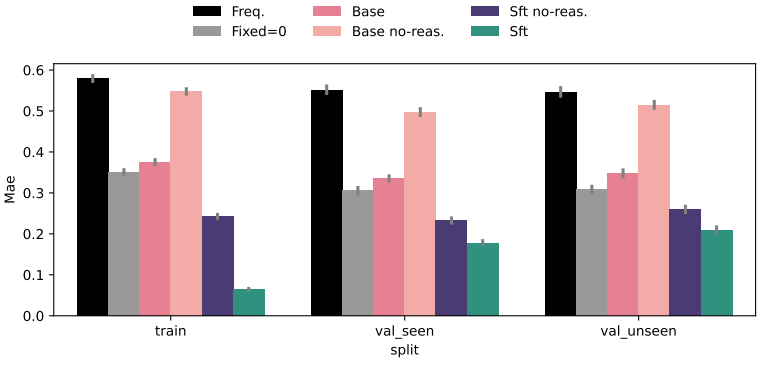


Figure 6: Ablation of different models' performance after the first training phase on QAsk-Nav-dataset. Performance is measured with the *mean absolute error* between the output score and the ground truth score in the dataset. After this phase, a model can only return R and S.

8.2 Qualitative Examples

Fig. 7 reports a qualitative example of Light-CoNav navigating a hard environment with different distractors. For visualization purposes, we report a short reasoning R, omit the score (which can be easily inferred from the conclusion), and append the question Q after the reasoning. In practice, at each detection, the model returns (R, S, Q) following the prompt in Sec. 12.

The goal is to find a 'blue painting' (D). At the start (1), it immediately detects a painting; however, it correctly concludes that the painting is not the target blue painting, being pink $(S = 0)$. Next, after moving in the environment, it detects another painting (2). This time, too, it correctly concludes that this is not the target painting $(S = 0)$. The third observation is much harder: the agent mistakenly sees a mirror, reflecting blue light, as a possible painting, and stops the navigation to ask a question $(S = 1)$. However, the answer of the user clears its doubts by helping it localizing the true target painting, which is correctly detected at the end (4) $(S = 2)$.

In addition to this example, we also present more qualitative episodes in the last 60 seconds of the accompanying video, showcasing the interactive dynamics of Light-CoNav.

9 Question-asking Protocol

9.1 Object Category Distribution in Episode Data

Fig. 8 reports statistics of the question-asking episode data. On the left, we report the object category distribution (top) and the length of episodes, as measured in terms of the number of observations (bottom), for the training episodes. On the right, we report the same statistics for the validation episodes. For the category distribution, we report data of the top-30 most frequent categories for best visibility. We can see that the object categories are well-distributed, especially for the training episodes, with the top-30 categories each having more than 16 episodes. In terms of episode length, the average validation episode is slightly

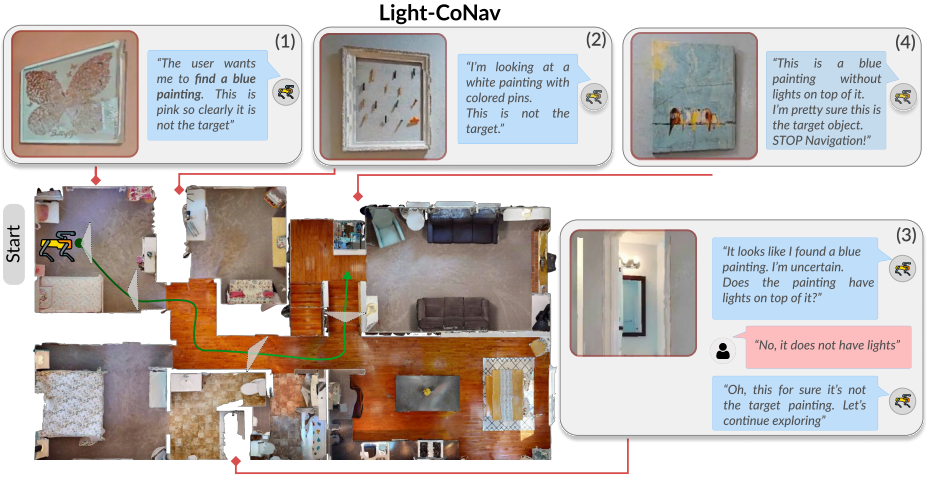


Figure 7: A CoIN episode. Light-CoNav is looking for a 'blue painting' (D). In the first two detections, (1) and (2), it correctly evaluates the observations, concluding that they are not the target object. (3) is a difficult observation, as it looks like a painting. However, asking a question to the user disambiguates it. (4) is the target object.

longer than the average training episode at ~ 4.8 instead of around ~ 4.1.

9.2 Augmentation of Images and Descriptions with Editing

Relying solely on images obtained from navigation episodes is limiting, because they can be of low quality, have bad viewpoints, or be very different from each other. Therefore, we start from images of common household objects and use a state-of-the-art commercial image generation model (`Gemini-3-pro-image` [🍌]) to edit the original images to differ in a small number of details. For each original image, we generate five additional images, changing 1 to 5 details. These details include color, shape, texture, presence, and position of objects. Moreover, we add or remove objects from the images. Fig. 9 and Fig. 10 show 32 examples of images generated using our protocol. On the left, we show the original image; on the right, three synthetic images derived from it. The results are of high visual quality with coherent semantics. The changed details are mostly color and surrounding objects, but also involve texture and shape. In addition, we generate six different descriptions for the images, differing in terms of information density and specificity. See Fig. 16 and Fig. 17 for three examples of different description types.

To ensure high-quality and diverse data, we manually inspected around 75% of generated distractors for realism and diversity, discarding low-quality samples. In particular, for around 75% of the starting pictures, we checked all five synthetic images generated from them, discarding those that were too similar to the others, that had unrealistic details added (for example, objects added in improbable positions), and those with generic artifacts.

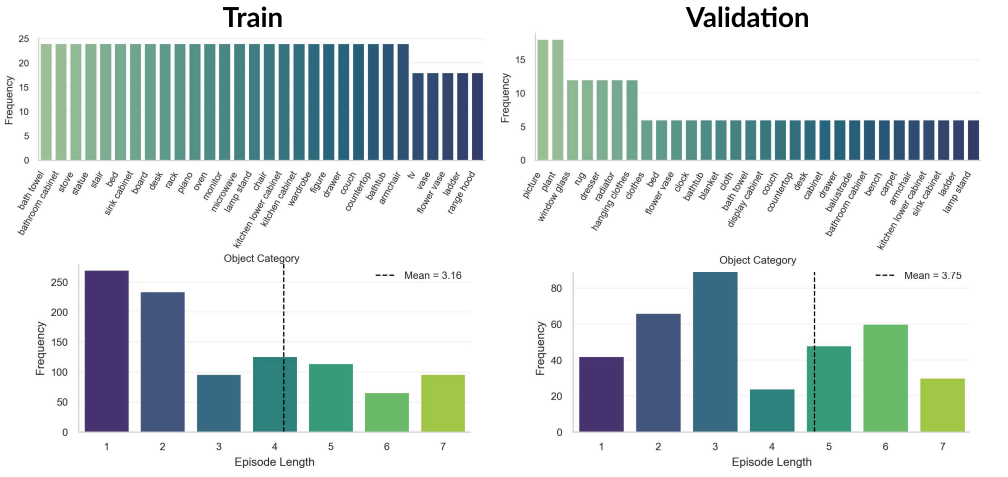


Figure 8: Categories distribution (top 30) of the target image, and lengths of episodes (in number of distractors; dotted line is the mean length), for the question-asking protocol (left: *Train* split, right: *Validation* split). The categories are well distributed, with the top 30 having at least 16 (for the train split) or 6 (for the validation split) examples.

10 Navigation Protocol

10.1 Human Evaluation of Task Descriptions

In our navigation protocol, we proposed a recipe to generate high-quality and diverse task descriptions for navigation. Therefore, we assess the quality of the proposed task descriptions when comparing them to CoIN-Bench's ones [45]. We set up a preference test with two choices (CoIN-Bench [45] *v.s.* ours, order randomized) on 50% of the data samples and ask a evaluator which option they prefer. For this task, we use both an LLM-as-judge evaluator (`Gemini-3-pro`) and 5 human evaluators.

We instruct the evaluator to focus on *precision*, *conciseness*, and *clarity*; moreover, we ask the evaluator to check the annotations for *critical problems* concerning imprecise details and hallucinations via additional check boxes. The evaluation tool we developed is shown in Fig. 12. We measure the `preference rate` (↑) defined as the ratio between the number of preferences expressed for one or the other description and the total number of expressed preferences; and the `problem rate` (↓) defined as the ratio between the number of problematic descriptions with hallucinations or imprecise details and the number of annotations of that type.

We analyze in Fig. 13 the distribution of the length of the task descriptions in terms of words. As shown clearly, our task descriptions are shorter with many fewer outliers, compared to the ones of COIN-Bench. The average length of CoIN-Bench descriptions is around ~ 17, with a standard deviation of ~ 9. In contrast, our annotations are ~ 9 words long with a smaller standard deviation of ~ 2.

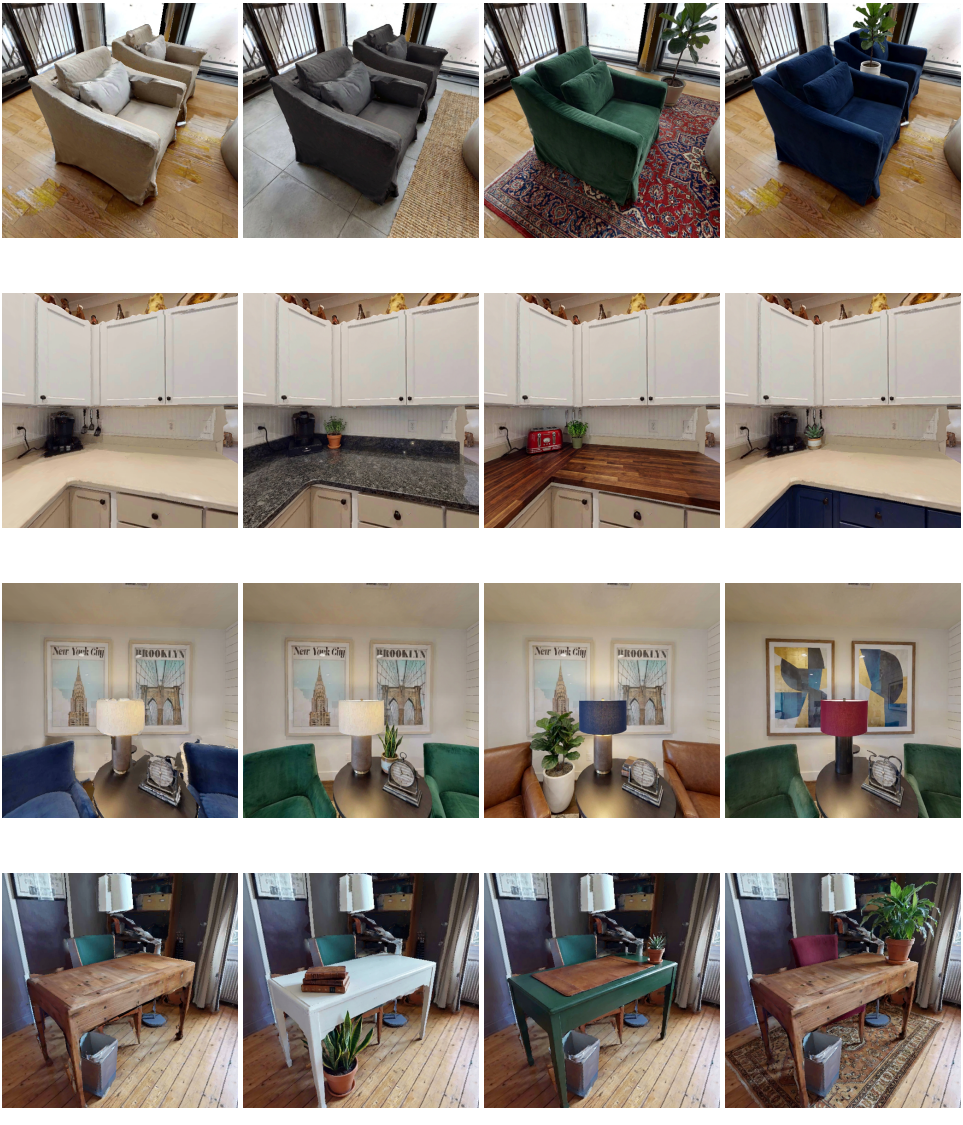


Figure 9: Variability of images used in our question-asking protocol. The first image is the original image; the next three are variations obtained using a strong generative model (`Gemini-3-pro-image` [🎞])

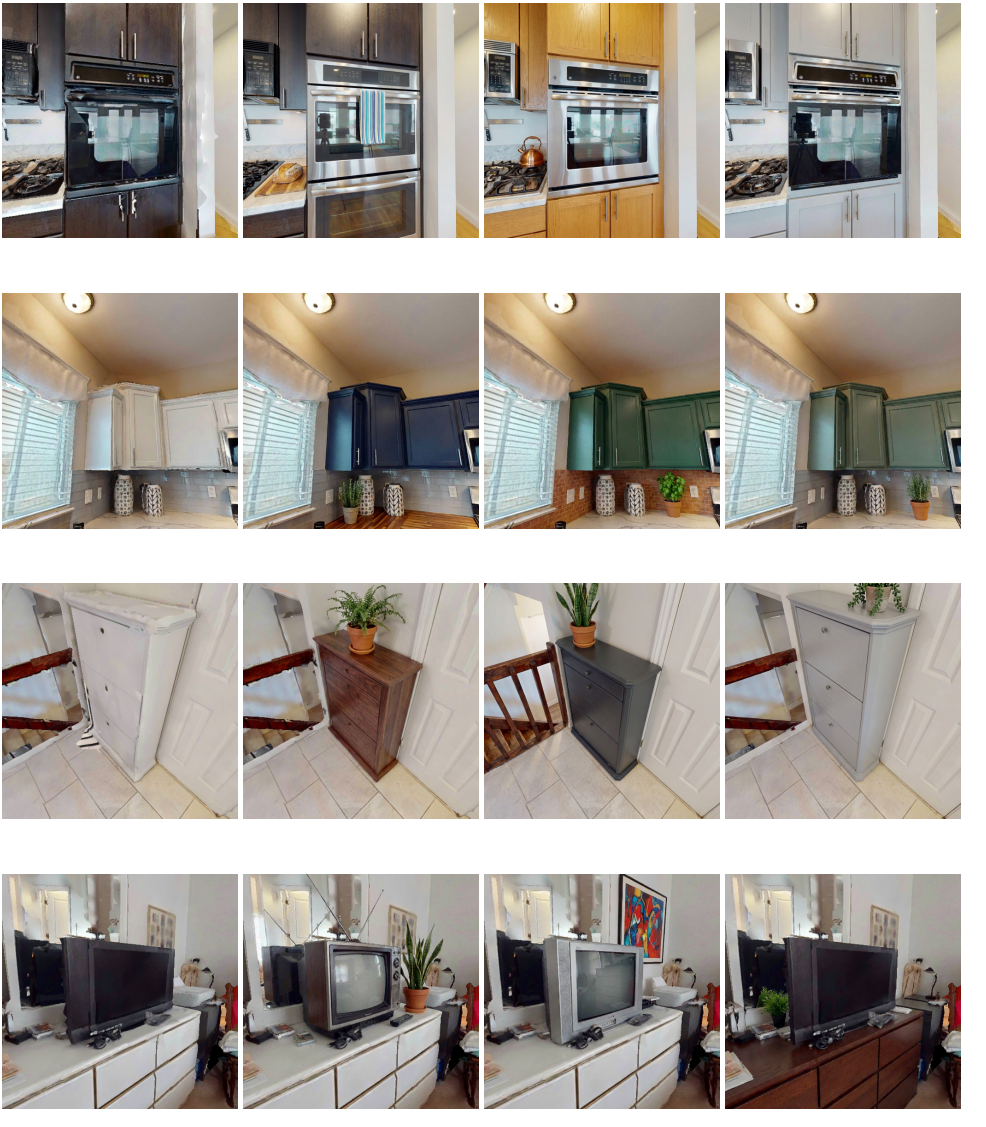


Figure 10: Variability of images used in our question-asking protocol (cont'd).

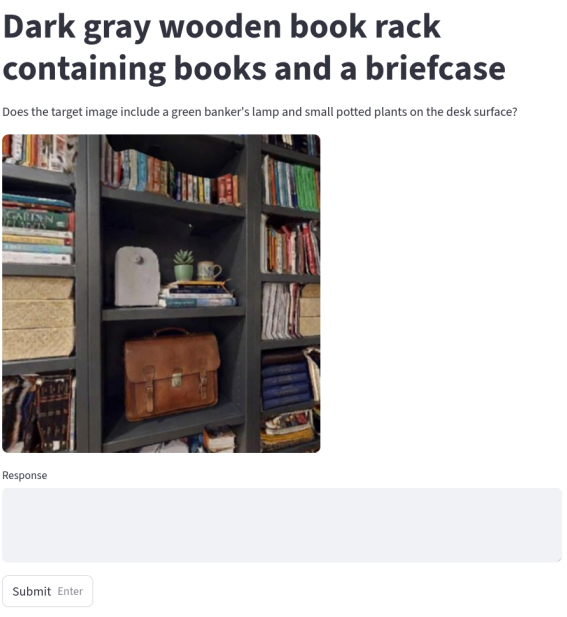


Figure 11: Tool for the human oracle tests. The target image is presented to the user, along with the task description (the title), and the current question asked by the questioner (below the title). The user is instructed to answer by writing the answer in the text box below.

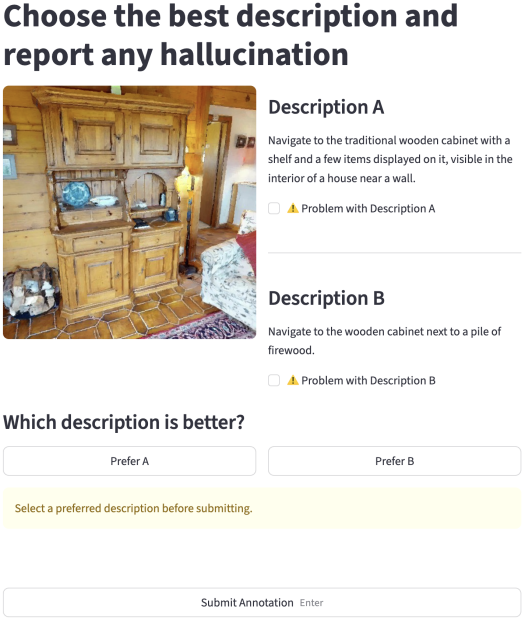


Figure 12: Evaluation tool for preference testing in our human evaluation. Two choices are presented to the user. The users are instructed to click on the checkbox "Problem with Description X" if they find hallucinations and clear mistakes.

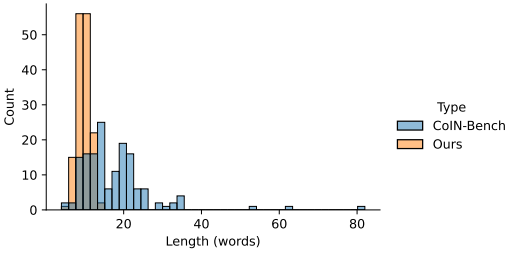


Figure 13: Length of descriptions, in words. The CoIN-Bench descriptions exhibit extreme outliers and, in general, are longer and wordier than our descriptions.

10.2 Episode Data

Here we provide more information about the navigation episodes. In Fig. 14, we report the distribution of object categories for all goal instance objects of our navigation episodes, divided into the three splits. The data split follows prior benchmarks' practice to maintain compatibility. We can see that the *val seen synonyms* split only contains three categories of objects, whereas the *train* and *val unseen* split contain at least 9 different categories.

11 QAsk-Nav-dataset

11.1 Data Collection

We describe here how we built QAsk-Nav-dataset, which was done in two independent phases to increase diversity.

In the first phase, we collected around 500 instance object images from both our navigation episodes and other related datasets (GOAT-Bench [15] and CoIN-Bench [45]) to have a diverse set of objects and scenarios. We created three splits sourced from the original navigation datasets [15, 45]: (i) *Train* contains images sourced from train episodes; (ii) *Val seen* has images from *Val seen* and *Val seen synonyms* episodes, and (iii) *Val unseen* contains images sourced from validation episodes including objects not present in other splits. Human annotators are tasked to compose two high-quality descriptions for each image.

We then group all images and descriptions by object category, creating a pool of images and associated descriptions. In addition, we used `Gemini-3-flash` [9] to write 20 additional descriptions for each object category (not based on any image), and add these descriptions to the pool of descriptions for that object category. Then, we prompted `Gemini-3-pro` to write high-quality reasoning R, scores S, and questions Q given the description D and image O, as described in the main manuscript in Sec. 3.4. By doing so, we obtain approximately 24,000 (D, O, R, S, Q, C) samples (note: the context C is empty here and used during navigation). In this phase, the dataset is highly imbalanced toward entries with scores 0 and 2, with very few questions. Later, in the second phase, we collect many more questions and entries with a score of 1.

In the second phase, we collected a set of samples following our question-asking protocol, *i.e.*, we run `Gemini-3-flash` [9] as questioner using another instance of `Gemini-3-flas` as oracle. During each interaction, the questioner is presented with a description D and an observation O (an image of an object). If it thinks D matches O, then it returns a

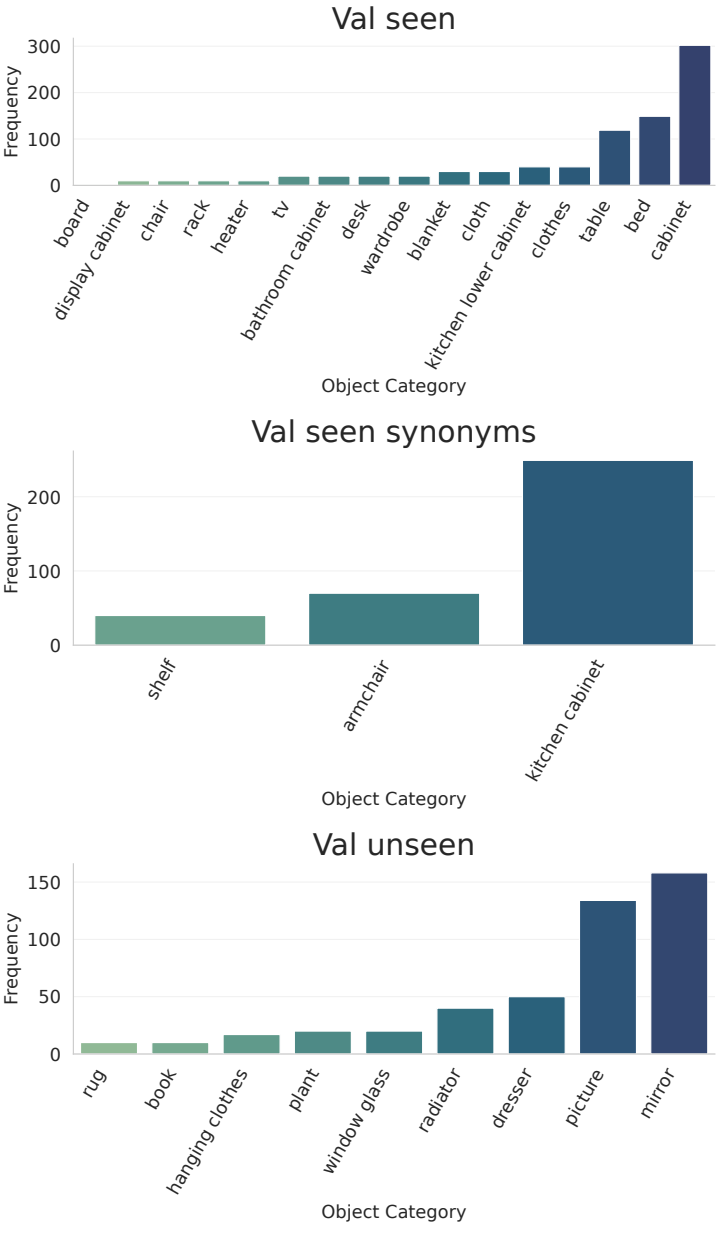


Figure 14: Distribution of the target instance objects categories, per split, for the navigation episodes.

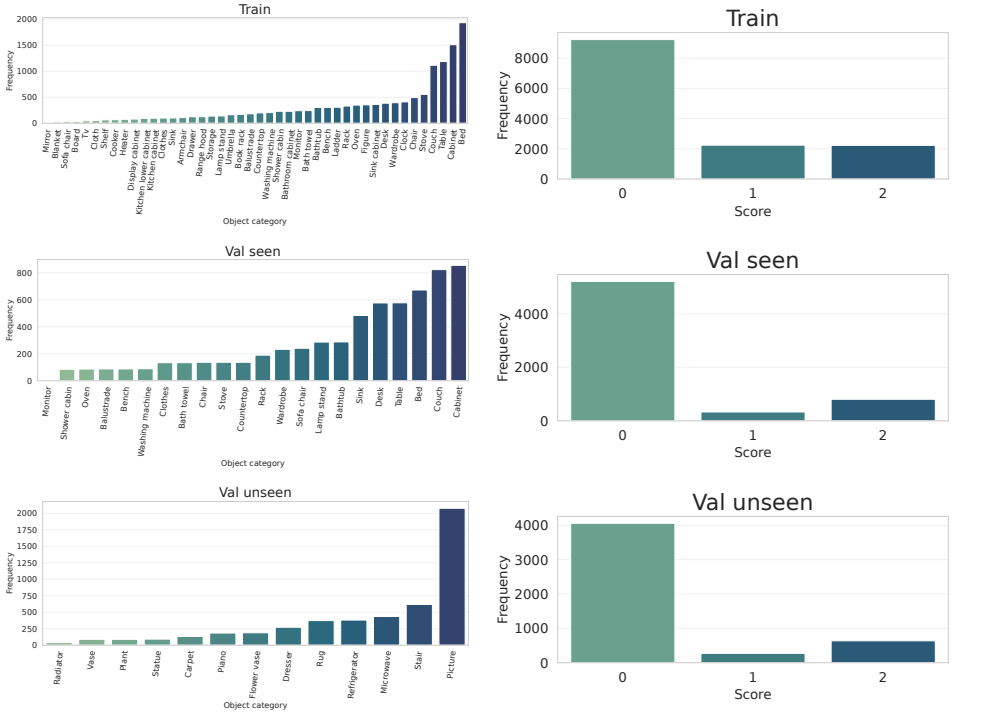


Figure 15: Distribution of the object categories in QAsk-Nav-dataset (left) and distribution of scores (right), divided by split.

positive conclusion; if it thinks D does not match O, it returns a negative conclusion; if it is uncertain, it can ask a question. Every time it produces a reasoning R, motivating the decision. We map these interactions to the following (R,S,Q) tuples, respectively: $(R,S=2,\text{None}),(R,S=0,\text{None}),(R,S=1,\text{Question})$. If there were no previous questions and answers, then $C=\emptyset$; otherwise, C will contain the previous question-answer pairs. Therefore, each interaction maps directly to an entry of our dataset: (D,O,R,S,Q,C). In this way, we collected around 4,000 interactions from the training question-asking episode data and included them in the *Train* split of QAsk-Nav-dataset. Finally, we combined those samples with the 24,000 samples from the previous phase to obtain the 26,000 samples of QAsk-Nav-dataset.

To ensure a high-quality dataset, we conducted a human analysis of thousands of samples (around 1.1K), involving three people. More precisely, our goal was to check whether the produced reasonings were: i) correct (stated the correct conclusion); ii) clear and concise (followed a clear and logical sequence of statements, not a convoluted one); iii) did not 'hallucinate' objects and/or features not present in the image. We found a very low error rate, under 1%, with errors mainly related to ambiguous statements (when a clear conclusion was harder than usual). We then removed the bad samples from the dataset.

11.2 Dataset Statistics

In Fig. 15, we show information about QAsk-Nav-dataset. In particular, for each split (*Train*, *Val Seen* and *Val Unseen*), we report (left) the object categories distribution, and (right) the distribution of scores. We can see that the dataset contains, predominantly, $S = 0$. This is expected as the images, in general, are quite different than each other, and therefore most image-description pairs do not match. The *training* is more balanced, having as many $S = 1$ (and therefore questions) as $S = 2$. This is thanks to the second data collection phase, where we use rollouts of a strong *questioner* on our question-asking protocol to produce high-quality questions and reasoning traces. For this reason, the training set contains more questions (and $S = 1$) than the other two splits.

12 Prompts

We describe here how Light-CoNav is prompted to achieve the navigation goals. When a candidate object is detected from the current image, the observation O and the initial task description D are passed to Light-CoNav using the following prompt (`USER_TASK` is substituted with the description D and the image is prepended to this text):

```
You are a robot navigating an enclosed space. Your goal is to navigate
to the correct object based on the user's commands. You were given the
following task by the user '{USER_TASK}'.

Currently, you are facing a scene represented by the given image. Reason
 about what you are seeing, comparing what you know about the task (
given the user commands) and the given scene.
For example, if the description is 'Black leather sofa near a lampstand'
 your reasoning process will be 'I'm currently observing a brown sofa
which is different than black, making it unlikely to be the target sofa.
 Moreover, there is no lampstand near it, only a rug and a window' etc.

If there are distortions or artifacts, do not focus on them, focus on
the object at hand. At the end of the reasoning process, evaluate how
well the provided image aligns with the user's task.

Assign a confidence score based on the following scale:
- 0: You are certain the image **DOES NOT** match the task.
- 1: You are unsure whether the image matches the task or not.
- 2: You are certain the image **DOES** match the task.

Provide a concise reasoning (under 100 words). If you are unsure (score
1), ask an informative question to the user: this question should help
you dispel any doubt about the current observation. Otherwise, just
return ''.
Use your reasoning output to ask the best question possible. Prefer
asking questions at the start, rather than blindly coming to a
conclusion, if the description is very short and does not include many
details. The image might have distortions or digital artifacts: *NEVER*
mention them in the question. Consider the information contained in the
previous questions and answers, which are provided here.
```

Followed by either

> There are no previous questions or answers.

or

> 1. Question1 <|answer|>Answer1<|answer|>
> 2. Question2 <|answer|>Answer2<|answer|>
> ...

These two prompts are used, respectively, when the context is empty and when there are previous questions and answers in the context.
Lastly, the prompt includes the format to use:

> Strictly follow this output format: <motivation>Your reasoning here</motivation> <score>0, 1, or 2</score> <question>Your question or None (if score is not 1)</question>

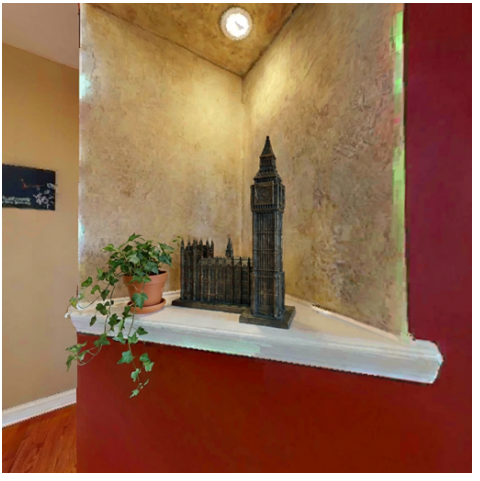
Therefore, we pass (D, O, C) to Light-CoNav using this prompt and Light-CoNav returns, in turn, the output (R, S, Q).

13 Limitations

Here we lay out the main limitations and possible future extensions of our work:

1. We always assume the real object is present in the environment;
2. The navigation model may fail when it cannot get a clear view of the target;
3. The QA evaluation environment, while using images with different points of view, does not explicitly include occluded images or same-object viewpoint variations

We plan to extend this work in the future, mainly by improving the diversity and scope of the images contained in the dataset, and to improve the navigation model by testing different backbones, less reliant on good, unobstructed object detections.

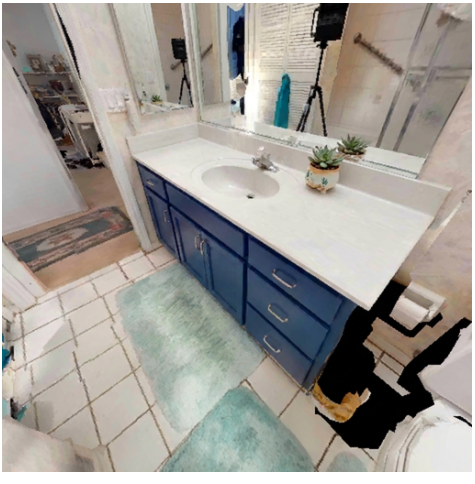


Category	Statue
Category-Color	Dark bronze statue
Color-Feature	Dark bronze statue of Big Ben
Context	Statue standing on a shelf next to a potted plant
Color-Context	Dark bronze statue standing on a shelf next to a potted plant
Color-Context-Feature	Dark bronze Big Ben statue standing on a shelf next to a potted plant



Category	Picture
Category-Color	Black framed poster
Color-Feature	Black framed poster with yellow text
Context	Poster hanging on a wall above a bicycle print
Color-Context	Black poster hanging on a brown wall above a print
Color-Context-Feature	Black framed poster with yellow text on a brown wall above a print

Figure 16: Different types of generated descriptions.



Category	Bathroom cabinet
Category-Color	Dark blue cabinet
Color-Feature	Dark blue cabinet with silver handles
Context	Cabinet underneath a white sink
Color-Context	Dark blue cabinet underneath a white sink
Color-Context-Feature	Dark blue cabinet with silver handles underneath a white sink

Figure 17: Different types of generated descriptions (cont'd).