Landmark-Based Conversational Indoor Navigation

Riccardo Bianco
ETH Zurich
rbianco@ethz.ch

Francesco Bondi
ETH Zurich
fbondi@ethz.ch

Roham Z. Nobari
ETH Zurich
rzendehdel@ethz.ch

Shaurya Panwar
UZH Zurich
shauryakishore.panwar@uzh.ch

Fatemeh Daneshmand
ZHAW Winterthur
dans@zhaw.ch

Abstract

Conversational indoor navigation aims to provide natural-language guidance for reaching destinations in indoor environments. However, existing approaches often rely on generic large language models that produce verbose or weakly grounded instructions, or on robot-centric navigation systems that use metric spatial references poorly aligned with human understanding.

In this work, we propose a language-driven indoor navigation system that generates concise, human-like instructions grounded in visible landmarks and structural cues. The system integrates path planning, semantic observations, and a lightweight on-device vision-language model to produce interpretable step-by-step navigation guidance from a single image and a natural-language query. By using landmarks and objects as reference points, the proposed approach avoids abstract distance-based descriptions and improves spatial clarity.

We evaluate the system through deployments in a real academic building and in a multi-floor residential environment with navigation tasks of increasing difficulty. In addition, a user study combining qualitative feedback and a quantitative performance metrics demonstrates that the proposed approach produces more effective and user-preferred indoor navigation instructions.

1. Introduction

Conversational indoor navigation aims to assist users in reaching destinations within complex indoor spaces through natural-language instructions. Unlike outdoor navigation, which benefits from standardized maps and reliable global positioning, indoor environments are characterized by intricate layouts, limited localization cues, and a stronger reliance on locally meaningful spatial references. As a result, effective indoor navigation requires guidance that is not only accurate, but also expressed in terms that align with how people naturally describe routes, such as landmarks and structural elements.

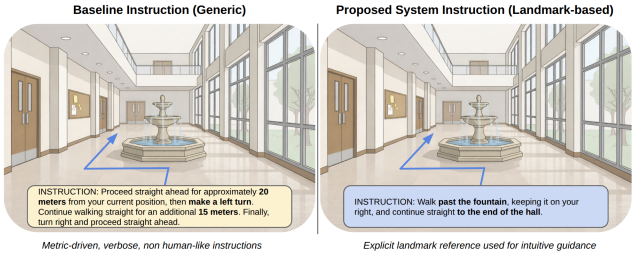


Figure 1. Comparison between baseline navigation instructions and our landmark-based, human-oriented guidance. The baseline approach (left) relies on metric and abstract descriptions, resulting in verbose and less intuitive instructions. In contrast, our method (right) explicitly references visible landmarks, producing concise and human-interpretable guidance.

Recent work has increasingly framed indoor navigation instruction generation as a controllable language modeling problem. While recent approaches demonstrate that modeling spatial context significantly improves instruction clarity [4, 14], they often fail to bridge the gap to efficient, real-world deployment. These methods typically operate on abstract path representations or rely on large, server-grade language models, lacking direct grounding in the realistic perceptual observations available on a user's device.

In this work, we focus on language-driven indoor navigation in pre-mapped environments, with an emphasis on human-oriented instruction generation. Rather than describing routes through abstract geometry, our approach expresses navigation in terms of landmarks, structural elements, and local spatial cues visible along the route. Figure 1 illustrates the qualitative difference between generic navigation instructions and the landmark-based guidance considered in this work.

To support controlled experimentation while preserving perceptual realism, we develop our system within the Habitat-Sim simulation framework [10], using semantically annotated indoor environments. Given a natural-language query (e.g., "How do I get to room HG E 3?") and an estimated user location, the system computes a feasible path to the target and extracts semantic observations along the planned trajectory. These observations are then used to generate navigation instructions that explicitly reference visible landmarks and spatial relations, reducing ambiguity and improving interpretability. This design is supported by recent evidence that grounding language generation in temporally ordered perceptual context leads to clearer and more consistent navigation instructions [12, 13].

We evaluate the proposed approach in two representative indoor settings: a multi-floor residential environment and a simulated reconstruction of a real academic building. A user study combining quantitative metrics and qualitative feedback demonstrates the effectiveness of landmark-grounded instructions for conversational indoor navigation.

The contributions of this work are twofold. First, we present a navigation pipeline that integrates user localization, path planning, semantic landmark extraction, and instruction generation using lightweight language models suitable for on-device deployment. Second, we validate the proposed system through experimental evaluation and user studies in realistic indoor environments, providing insights into the practical benefits and limitations of landmark-based conversational navigation.

2. Related Work

Our work builds on prior research in simulation-based indoor environments, visual localization, and navigation instruction generation. Simulation platforms such as Habitat-Sim [10] provide photorealistic indoor scenes with accurate geometry and sensor simulation, and are widely used for navigation and perception research. Large-scale datasets like HM3D [6] further enable realistic indoor modeling across diverse layouts and object distributions. In our system, we use Habitat-Sim populated with HM3D scenes as a controllable backend to generate navigation trajectories and dense perceptual observations, which we exploit to extract semantic information such as object visibility and spatial context along planned routes.

To anchor navigation in real-world scenarios, accurate user localization is required. Given a single RGB query image captured by the user, the pipeline estimates a 6-DoF camera pose by matching it against a pre-built sparse 3D reconstruction of the environment. The estimated pose initializes the user's position within the simulated scene and serves as the starting point for path planning. We employ a hierarchical image-based localization pipeline implemented using hloc [7] and COLMAP [11], following the methodology and evaluation setting introduced in the LaMAR benchmark [9].

Closest to our goal are works treating navigation as a controllable language-generation problem. *Controllable Navigation Instruction Generation* [14] uses Chain-of-Thought prompting to derive instructions from abstract route data, while the *Spatially-Aware Speaker* [4] emphasizes spatial consistency in generated text. However, these methods typically rely on abstract route representations or computationally intensive server-side models (e.g., GPT-4). In contrast to these formulations, our approach explicitly grounds generation in visual landmarks extracted from the Habitat-Sim reconstruction. By summarizing these rich visual cues into a structured context, we enable the use of lightweight on-device models without sacrificing the human-like quality of the instructions.

3. Method

3.1. System Overview

We propose a navigation pipeline takes as input a user-captured RGB image, and a natural-language navigation query, and outputs concise, human-like navigation instructions grounded in visible landmarks. Figure 2 illustrates an overview of the system. The core idea is to (i) localize the user in a simulated reconstruction of the environment, (ii) compute a feasible path to the requested destination inside Habitat-Sim, (iii) extract semantic cues along the path, and (iv) generate a human-interpretable description using a fine-tuned language model.

3.2. User Localization and Path Planning

Given a user-captured RGB image and a natural-language query, the system first determines whether the query corresponds to an indoor navigation request and extracts the target destination. The extracted goal is mapped to a normalized semantic representation compatible with the simulator, such as a room identifier or object category (e.g., `HG E 3` or `couch`). This normalization bridges free-form language and the discrete semantic labels used by the environment, accounting for lexical variation and synonyms (e.g., couch, sofa) in user queries.

Visual localization (single-image). Once a valid navigation goal is identified, we estimate the user's pose within the simulated environment using a visual localization module based on LaMAR, which builds on the hierarchical localization framework implemented in hloc [7] and COLMAP [11]. The module localizes a single query image against a pre-built sparse 3D reconstruction of the environment.

Let I_q denote the query image and $\mathbf{K} \in \mathbb{R}^{3\times3}$ its intrinsic calibration matrix. Camera intrinsics are inferred

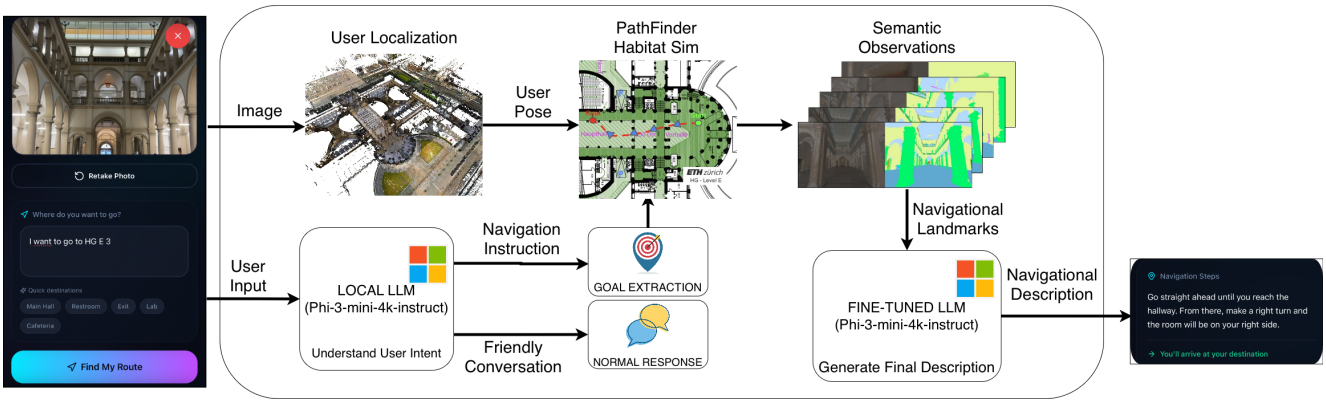
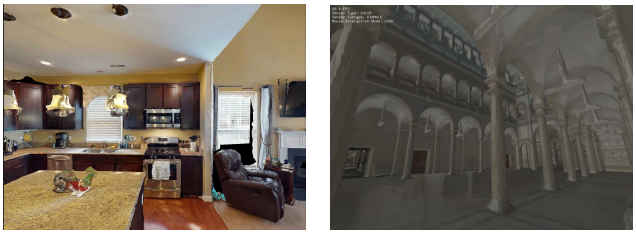


Figure 2. Overview of the proposed mixed-reality navigation system. Given a user image and a natural-language query through our web app, the system localizes the user, plans a path in Habitat-Sim, extracts semantic landmarks along the route, and generates grounded navigation instructions.



(a) Simulated house environment (b) Simulated HG building

Figure 3. **Simulated indoor environments used in this work.** (a) Multi-room residential environment used for development and pipeline evaluation. (b) Simulated reconstruction of the HG academic building in Habitat-Sim.

using COLMAP's camera inference from image metadata when available; otherwise, we assume a pinhole camera model with $f_x = f_y = 0.7\max(W, H)$ and principal point $(c_x, c_y) = (W/2, H/2)$ for an image of resolution $W \times H$.

The reference map consists of a set of images $\{I_j\}_{j\in\mathcal{M}}$ together with a sparse 3D point cloud $\mathcal{X} = \{\mathbf{X}_k \in \mathbb{R}^3\}_{k=1}^{N}$ obtained via COLMAP triangulation. The goal of localization is to estimate the 6-DoF query camera pose $\mathbf{T}_{wc} = (\mathbf{R}, \mathbf{t}) \in SE(3)$, which maps camera coordinates to the map frame.

Localization proceeds through the following stages:

1. **Local feature extraction.** We extract local keypoints and descriptors from the query image and all reference images using SuperPoint [2].
2. **Image retrieval.** To restrict matching to a small set of relevant candidates, we retrieve the top-K most similar reference images using NetVLAD [1], with $K = 10$ in our experiments.
3. **Local feature matching.** For each retrieved reference image, we establish 2D–2D correspondences with the query image using SuperGlue [8].
4. **2D–3D correspondence construction.** Matched reference keypoints that are associated with triangulated 3D points are lifted to 2D–3D correspondences using the pre-built reconstruction.
5. **Pose estimation.** Given the resulting 2D–3D correspondences and camera intrinsics $\mathbf{K}$, the query pose $\mathbf{T}_{wc}$ is estimated using robust Perspective-n-Point (PnP) with RANSAC [3].

The estimated pose $\mathbf{T}_{wc}$ localizes the user image within the map coordinate frame and serves as the starting state for subsequent path planning in the simulated environment.

Path planning in Habitat-Sim. Given the estimated start pose $\mathbf{T}_{wc}$ and the normalized goal specification, we compute a feasible path using Habitat-Sim's built-in planning module (*PathFinder*). The planner returns a collision-free trajectory represented as a sequence of waypoints $\mathcal{P} = \{p_1, \ldots, p_N\}$ in the simulator coordinate frame, which serves as the reference path for subsequent viewpoint sampling, landmark extraction, and instruction generation.

3.3. Landmark Extraction

Let the planned navigation path be $\mathcal{P} = \{p_1, \ldots, p_N\} \subset \mathbb{R}^3$. We densify $\mathcal{P}$ to obtain a sequence of viewpoints $\mathcal{V} = \{v_1, \ldots, v_M\}$, where each v_i consists of a 3D position and a viewing direction aligned with the path tangent.

At each viewpoint v_i, the simulator provides RGB, depth, and semantic observations. From the semantic image, we extract a set of visible object instances $\mathcal{O}_i$, each associated with a semantic label, a 3D bounding box, and pixel support. We compute simple visibility cues based on screen coverage and distance to the camera.

We construct a set of global semantic clusters $\mathcal{C} = \{c_1, \ldots, c_L\}$ by grouping all scene objects according to normalized semantic labels and spatial proximity, discarding

Algorithm 1 Landmark Extraction

Require: Path $\mathcal{P}$
Ensure: Landmark sets $\{\mathcal{L}_i\}$
Densify $\mathcal{P}$ to viewpoints $\{v_i\}$
for each v_i **do**
  Extract visible instances $\mathcal{O}_i$
  Assign instances to global clusters $\mathcal{C}$
  Compute saliency scores $s_{i,\ell}$
  $\mathcal{L}_i \leftarrow \{c_\ell \mid s_{i,\ell} \geq \tau\}$
end for
return $\{\mathcal{L}_i\}$

non-informative categories (e.g., walls and floors). This is necessary because semantic annotations may over-segment a single physical structure into multiple object instances with identical labels (e.g., individual steps of a staircase). By merging such instances, each cluster represents a persistent, human-interpretable semantic landmark.

At each viewpoint, visible objects $\mathcal{O}_i$ are assigned to their corresponding clusters. For each cluster c_ℓ, we compute an aggregated saliency score

$$s_{i,\ell} = \sum_{o \in \mathcal{O}_i \cap c_\ell} w(o),$$

where $w(o)$ is proportional to the number of image pixels covered by instance o. Clusters with $s_{i,\ell} < \tau$ are discarded, yielding the landmark set $\mathcal{L}_i = \{c_\ell \mid s_{i,\ell} \geq \tau\}$.

The resulting sequence $\{\mathcal{L}_i\}_{i=1}^{M}$ provides a compact, perceptually grounded representation of the environment along the navigation trajectory.

3.4. Instruction Generation with a Lightweight LLM

The final navigation instructions are generated from an ordered sequence of viewpoints and their associated semantic landmarks using a lightweight language model. The model input includes (i) the target destination, (ii) a structured representation of the planned trajectory (e.g., turns and decision points), and (iii) a temporally ordered set of landmark cues extracted along the route.

We employ *Phi-3-mini-4k-instruct* as a lightweight language model and adapt it to the navigation task via parameter-efficient fine-tuning using LoRA. Figure 4 presents a qualitative comparison between instructions generated by three different models: (i) the base *Phi-3-mini-4k-instruct* model without task-specific adaptation, (ii) the same model after fine-tuning it via Low-Rank Approximation (LoRA) [5] for landmark-based navigation, and (iii) ground-truth instructions generated using a large proprietary language model (GPT-4o).

LoRA-based fine-tuning updates only a small subset of parameters, allowing the base model to acquire task-specific navigation behavior while preserving its general language capabilities. This approach enables a lightweight local model—requiring approximately 3 GB of GPU memory at inference time—to generate navigation instructions that are qualitatively comparable to those produced by substantially larger proprietary models, such as GPT-4o. In particular, the fine-tuned model exhibits similar fluency, landmark usage, and goal specification, while remaining suitable for fully on-device deployment.

Query: *Where can I find the plant in the office?*

Base model: Start by turning around. Proceed straight into the entryway, then turn right and walk into the kitchen. Continue straight into the living room and finally go up the stairs to reach the office.

Fine-tuned model: Start by turning around and walk straight through the lower bedroom into the entryway. Turn right into the kitchen and continue forward, passing through the living room. After passing the sofa, the stairs will be on your right. Go up the stairs to reach the office. Once inside, the plant is located on your right.

Ground truth: Turn around and walk straight ahead through the lower bedroom into the entryway. Turn right and continue into the kitchen, then proceed straight into the living room. Once you pass the sofa, the stairs will be on your right. Go upstairs to reach the office. The plant is located on your right.

Figure 4. **Qualitative comparison.** Example navigation instructions for a query in the simulated house environment. The base model primarily describes the route in terms of room transitions, without referencing objects or landmarks along the path, and stops at the target room without mentioning the goal. In contrast, the fine-tuned model generates more fluent, human-like instructions grounded in visible landmarks (e.g., the sofa) and explicitly indicates the goal's position, closely matching the ground-truth description.

Building on the strong performance of our landmark-based pipeline combined with a large proprietary language model, we automatically generate a training dataset by treating navigation instructions produced by the ChatGPT API as ground truth. This procedure yields 5,000 navigation examples from simulated episodes, each pairing a user-style query with instructions derived from semantic observations along the trajectory, without manual annotation.

As a result, the fine-tuned model produces step-by-step navigation instructions that remain tightly grounded in perceptual landmarks and spatial context, closely reflecting how humans naturally describe indoor routes, without relying on cloud-based inference or large-scale language models.

4. Results and Analysis

We evaluate the proposed navigation system in two stages. First, we assess the instruction generation pipeline in a controlled simulated house environment (see Figure 3a), where the starting position and target are known and no user localization is required. This setting allows us to analyze the quality of the generated instructions independently of localization and interface effects. Second, we report results on the HG building environment (see Figure 3b), which represents the full pipeline, including both localization and user interaction through a web-based application.

4.1. Evaluation in the Simulated House Environment

4.1.1. Experimental Setup

In the simulated house environment, navigation tasks are executed entirely within Habitat-Sim. Users issue natural-language queries through a textual interface and receive step-by-step navigation instructions generated by the system, as illustrated in Figure 8a. Since both the starting position and the target location are known, this setup isolates the instruction generation component and enables a focused evaluation of navigation language quality.

We compare the proposed navigation pipeline against multiple baselines and model configurations. Specifically, we evaluate: (i) a baseline approach that directly uses a large proprietary language model to generate navigation instructions without explicit path or semantic grounding; (ii) our full pipeline using a lightweight local language model; (iii) our pipeline using a large proprietary language model as the instruction generator; and (iv) our pipeline using a lightweight local language model fine-tuned for the navigation task. All evaluations are performed on the same set of navigation trajectories.

4.1.2. Evaluation Metrics

To assess the quality of the generated navigation instructions, we adopt three human-centric evaluation metrics. Each metric is rated on a 5-point Likert scale, where higher scores indicate better performance.

- **Presence and Usefulness of Reference Objects.** This metric evaluates whether the objects mentioned in the instructions are present in the environment, visible to the user, and useful as reference points for navigation. Higher scores correspond to instructions that consistently reference distinctive and relevant landmarks.
- **Spatial and Directional Correctness.** This metric measures the accuracy of spatial and directional references (e.g., left, right, in front of, behind) with respect to the user's position and orientation. Instructions with precise and unambiguous spatial cues receive higher scores.
- **Naturalness of Human Language.** This metric evaluates how closely the instructions resemble natural human language, considering grammatical correctness, verb choice, and idiomatic expressions. Higher scores indicate fluent, human-like navigation instructions.

4.1.3. Results

Qualitative examples of the pipeline in action in two different environments, showing egocentric views sampled along the planned trajectory together with their corresponding semantic segmentations, the input query, and the generated navigation instructions, are reported in Figure 5.

Figure 6 reveals a clear gap between unstructured language-only baselines and pipeline-based approaches. The baseline, which lacks explicit path and semantic grounding, performs poorly in spatial correctness and reference object usage, frequently producing ambiguous instructions. This confirms that reliable indoor navigation requires explicit geometric and perceptual constraints.

Among pipeline variants, both the cloud-based and fine-tuned local models achieve the highest scores. Notably, the fine-tuned lightweight model closely matches the cloud-based performance, indicating that task-specific fine-tuning is more impactful than model scale. Overall, grounding instruction generation in navigation trajectories and perceptual landmarks consistently improves spatial accuracy and language naturalness.

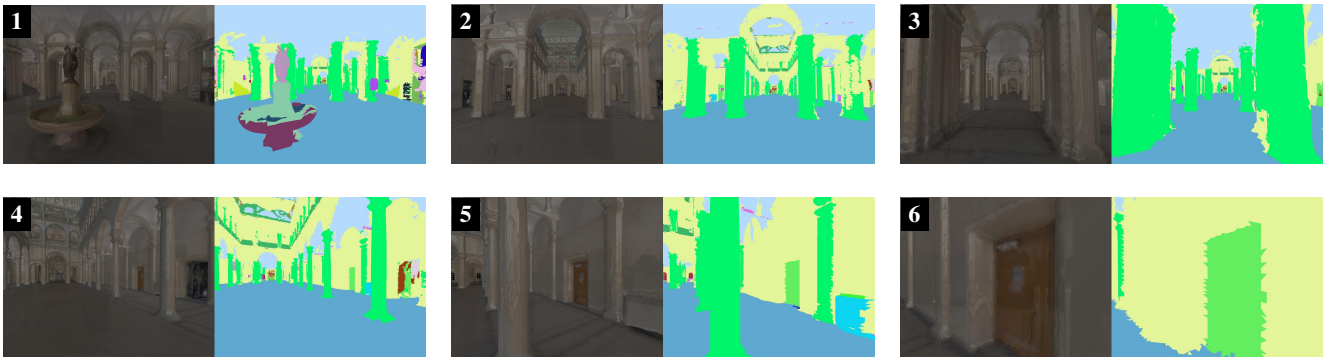
4.2. Evaluation in the HG Building Environment

We finally evaluate the system in the HG building environment, which represents the complete application. Unlike the simulated house setting, this scenario exercises the full pipeline, including user localization from a real image. Figure 7 shows instruction quality in the HG building environment, where the full pipeline is active. Despite additional uncertainty introduced by user localization, our pipeline-based approaches consistently outperform the unstructured baseline across all metrics.

The baseline exhibits a marked drop in spatial and directional correctness, confirming that language-only navigation degrades under realistic conditions. In contrast, landmark-grounded methods remain robust, with the fine-tuned lightweight model closely matching the cloud-based model. This suggests that explicit perceptual grounding and task-specific adaptation are more critical than model scale in realistic indoor navigation.

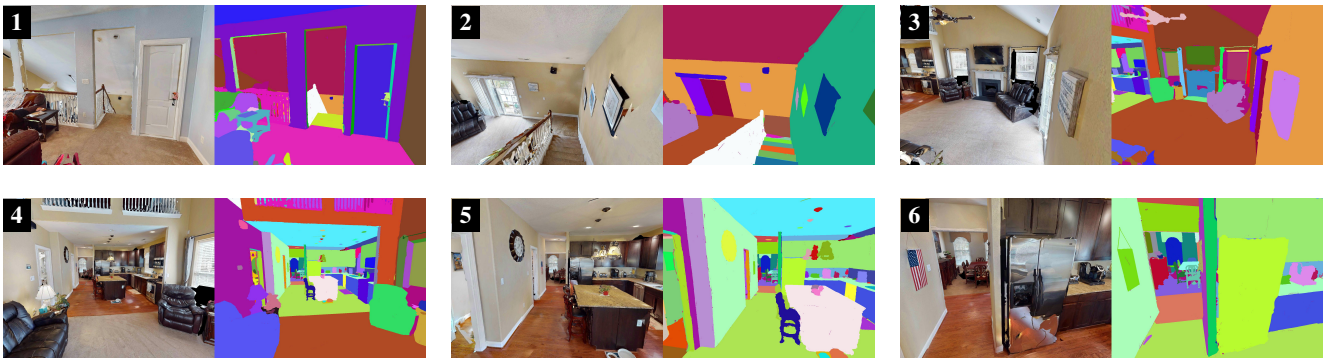
4.2.1. Experimental Setup

Users interact with the system through the web-based mobile interface by submitting a natural-language navigation query together with a single RGB image of their surroundings. The image is used to estimate the user's starting location in the simulated reconstruction of the HG building, after which navigation instructions are generated and displayed to the user through the web-based interface. An example view of the interface is shown in Figure 8.



Input: *"How do I get to room HG E 3?"*
Output: "Walk straight past the fountain, then turn slightly right when reaching the hall. Room HG E 3 should be in front of you."

(a) HG academic building



Input: *"Where is the fridge?"*
Output: "Start by moving forward in the office. Continue straight ahead and take the stairs down to the living room. Once in the living room, turn left and keep going straight, passing the couch on your left, until you reach the kitchen. You will find the refrigerator in front of you."

(b) Multi-floor house environment

Figure 5. **Qualitative examples of the proposed system in two indoor environments.** (a) Navigation in the HG academic building and (b) navigation in a multi-floor residential environment. In each case, six egocentric views sampled along the planned trajectory are shown in temporal order. For each view, the corresponding semantic segmentation is displayed on the right, with different colors indicating distinct objects and structural elements. The user query and the generated landmark-grounded navigation instructions are reported below each example.

4.2.2. End-to-End Latency Analysis

We analyze the end-to-end latency of the proposed system to assess its suitability for interactive mixed-reality navigation. All experiments were conducted on a laptop equipped with an Intel Core i9 CPU and an NVIDIA RTX 4070 GPU with 8 GB of VRAM. Table 1 reports the average execution time over 50 runs for the complete pipeline, including both the core navigation components and the localization module.

The results show that the overall end-to-end latency remains within an acceptable range for interactive use. Visual localization constitutes the dominant computational cost, accounting for the majority of the runtime, while all other stages introduce comparatively minor overhead. In particular, instruction generation with the lightweight on-device language model contributes only a small fraction of the total latency.

Although cloud-based language models can offer fast inference under ideal network conditions, our system is designed around a local lightweight model to satisfy practical deployment constraints. Local inference avoids reliance on network connectivity, provides more stable and predictable response times, and preserves user privacy by keeping egocentric visual inputs and navigation queries entirely on-

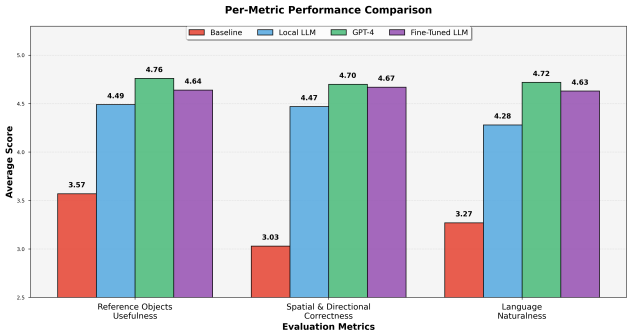


Figure 6. Evaluation in the simulated house environment. Average scores for the three instruction quality metrics, comparing the different configurations. Higher scores indicate better performance.

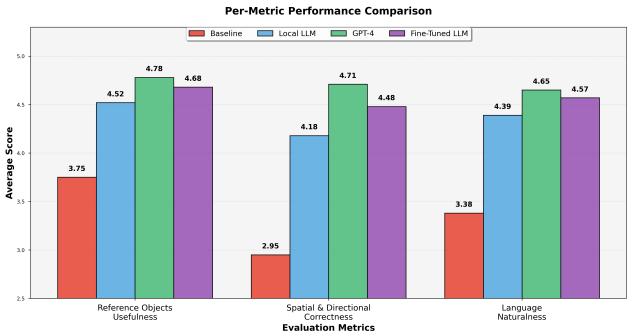


Figure 7. Evaluation in the HG building. Average scores for the three instruction quality metrics, comparing the different configurations. Higher scores indicate better performance.

device. These considerations are especially important for indoor navigation in personal or domestic environments.

From a system perspective, localization emerges as the primary bottleneck and represents the most promising target for future optimization.

5. Conclusion

This work presented a system for a natural-language indoor navigation assistant that combines user localization, path planning, semantic observations, and lightweight language models. By leveraging a simulated reconstruction of a real academic building, the proposed pipeline generates navigation instructions based on perceptually meaningful landmarks and spatial structure. The results show that integrating semantic cues with explicit path information leads to navigation instructions that are clearer and more intuitive than simple baseline approaches.

Through qualitative and quantitative evaluation, including a user study, we observe that the system provides consistent guidance across different navigation scenarios, with improvements in perceived clarity, spatial correctness, and overall usability. Importantly, our findings indicate that

Table 1. Average execution time (50 runs) per pipeline stage for the proposed system using a lightweight local language model.

Stage	Time (s)
Core navigation pipeline	
User intent classification	0.8778
Goal pose extraction	0.0003
Path planning	0.3791
Landmark extraction	0.0002
Instruction generation	1.7301
Total core pipeline time	2.9875
Web application component	
Visual localization	9.7610
Total end-to-end time	**12.7485**

compact local language models can already support effective instruction generation. In particular, a small on-device model such as Phi-3-mini achieves strong performance, while task-specific fine-tuning further narrows the gap with larger proprietary models, highlighting a favorable trade-off between model size, deployability, and instruction quality.

Despite these promising results, the current system relies on a web-based interface, where the user captures a single image and queries the system for navigation instructions. Future work will focus on extending the pipeline to incorporate real-time egocentric sensory inputs, such as wearable cameras or smart glasses providing a continuous video stream with real-time localization. In parallel, reducing end-to-end system latency will be a key objective, as it is expected to significantly improve system responsiveness and overall user experience. More broadly, this work suggests that mixed-reality systems provide a practical and effective framework for developing and evaluating indoor human-like navigation assistants driven by natural language.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[2] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 3

[3] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[4] Rishab Gopinathan, Matej Masek, Rawan Abu-Khalaf, and David Suter. Spatially-aware speaker for vision-and-language navigation instruction generation. *arXiv preprint arXiv:2409.05583*, 2024. 1, 2

[5] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4

[6] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *NeurIPS Datasets and Benchmarks Track*, 2021. 2

[7] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[8] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[9] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Lamar: Benchmarking localization and mapping for augmented reality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

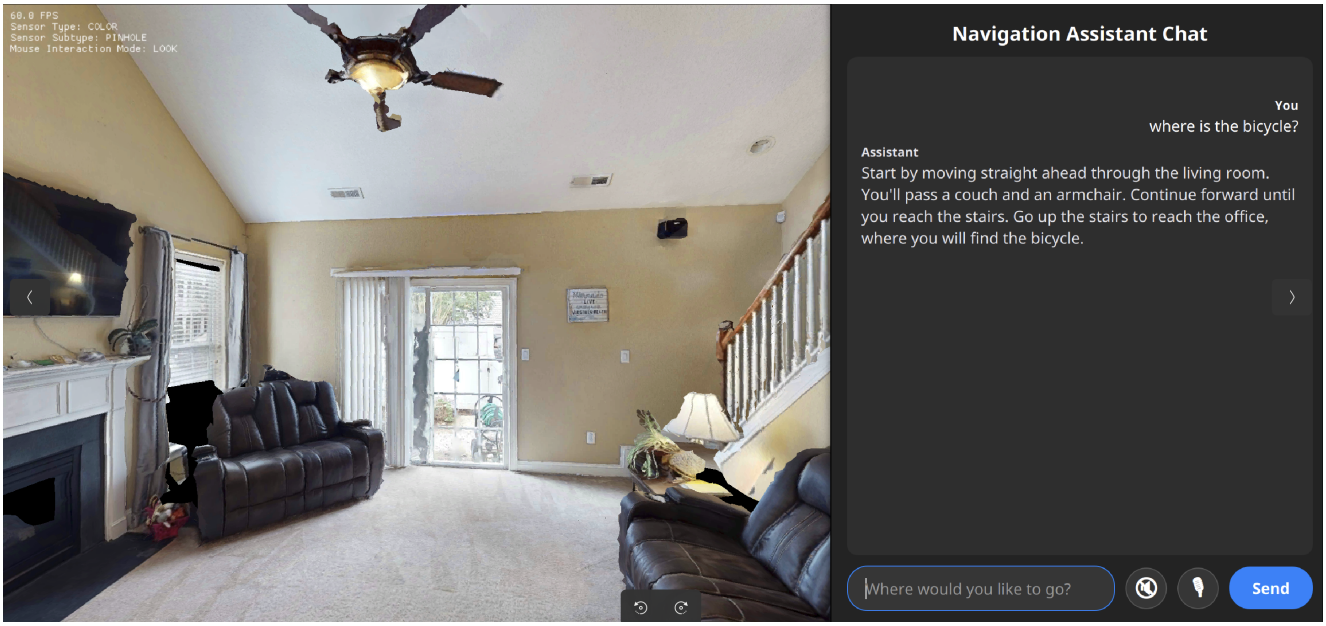
[10] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[11] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

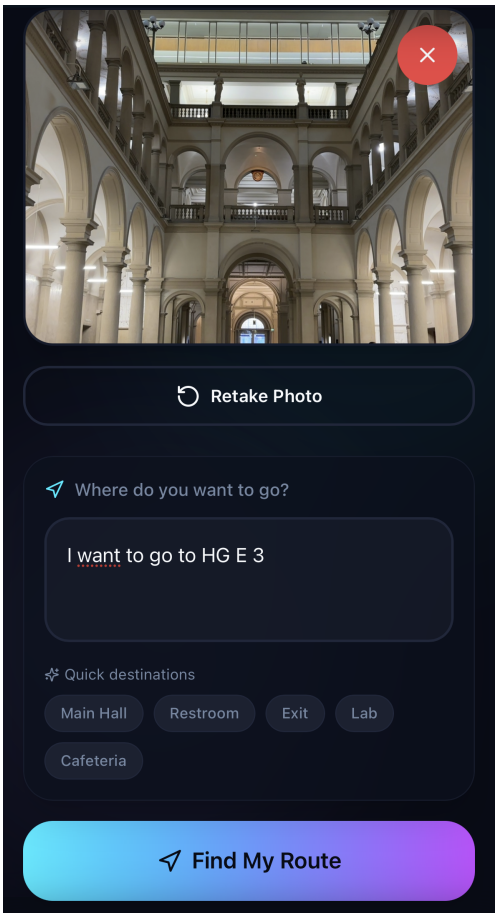
[12] Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025. 2

[13] Yuan Zhang, Chen Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2023. 2

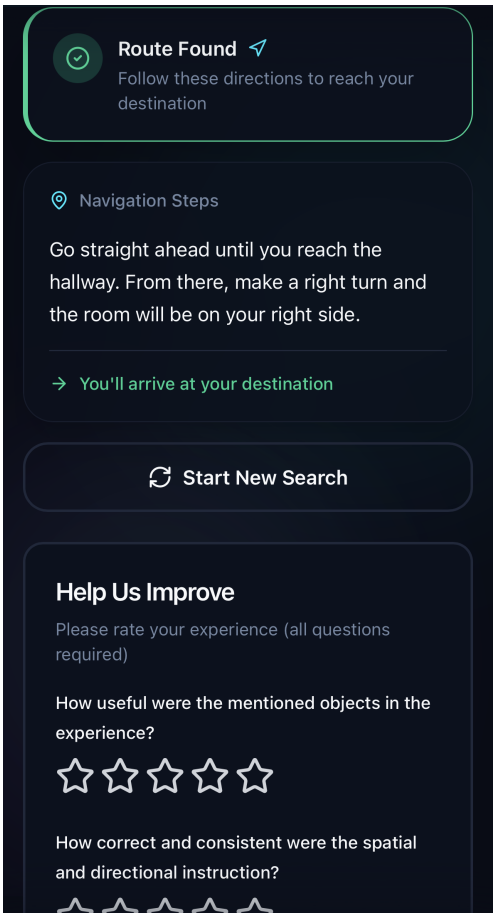
[14] Yifan Zhang, Ziyu Wang, Yuxuan Liu, Yifan Xu, and Xirui Li. Controllable navigation instruction generation with chain-of-thought prompting. *arXiv preprint arXiv:2407.07433*, 2024. 1, 2



(a) Experimental setup in the simulated house environment. The Habitat-Sim rendering is shown on the left, while the graphical interface for issuing navigation queries and visualizing instructions is shown on the right.



(b) Web interface for image submission and natural-language navigation queries.



(c) Web interface displaying the generated landmark-grounded navigation instructions.

Figure 8. **User interface and evaluation setup.** (**a**) Simulated house environment and experimental GUI used for controlled evaluation in Habitat-Sim. (**b**) Web-based mobile interface for submitting a user image and navigation request. (**c**) Visualization of the generated step-by-step navigation instructions grounded in semantic landmarks.