# Sign Language: Towards Sign Understanding for Robot Autonomy

Ayush Agrawal ‡∗ Joel Loo†∗ Nicky Zimmerman†∗ David Hsu†

***Abstract*— *Navigational signs* are common aids for human wayfinding and scene understanding, but are underutilized by robots. We argue that they benefit robot navigation and scene understanding, by directly encoding privileged information on actions, spatial regions, and relations. Interpreting signs in open-world settings remains a challenge owing to the complexity of scenes and signs, but recent advances in vision-language models (VLMs) make this feasible. To advance progress in this area, we introduce the task of *navigational sign understanding* which parses locations and associated directions from signs. We offer a benchmark for this task, proposing appropriate evaluation metrics and curating a test set capturing signs with varying complexity and design across diverse public spaces, from hospitals to shopping malls to transport hubs. We also provide a baseline approach using VLMs, and demonstrate their promise on navigational sign understanding. Code and dataset are available on Github [1].**

## I. Introduction

**Navigational signs** are ubiquitous elements of human environments designed to help humans navigate, and to aid spatial awareness and understanding in large, complex scenes [12], [32]. They typically use text and symbols to denote *locations* at different levels of abstraction (e.g. rooms, floors, buildings) and associate them with *directional* information. We argue that signs are rich, accessible sources of privileged information for robots. Signs facilitate *navigation* by conveying local, granular actions for reaching goals specified in natural language. As sources of discrete, symbolic information, signs align well with the growing shift toward topo-semantic navigation [11]. Signs provide information for rich, human-aligned *scene understanding*—a task that extends beyond scene mapping to capture abstractions (e.g., objects, regions) and relations among them [22]. Existing approaches to scene understanding are largely bottom-up: spatial abstractions and relations are inferred from direct sensor observations of the physical space. In contrast, signs enable a top-down process: they directly communicate human-specified spatial abstractions and relations, including for unseen regions and entities out of the robot's line-of-sight.

Despite their potential, navigational signs remain underutilized in robotics. Prior work has mainly focused on classifying fixed sets of sign categories, or on exploiting textual information in signs—as landmarks for localization [5], [37] or to annotate locations in semantic mapping [2]. We identify two key challenges limiting the use of navigational signs in the open world. First, parsing signs poses a challenge since they contain open-vocabulary text and open-set symbols, requiring semantic reasoning to interpret associations between elements of a sign (e.g. locations, directions). Second, detecting signs can be difficult as scenes contain diverse distractors which resemble but are semantically distinct from signs (e.g. displays, advertisements). Recent advances in vision-language models (VLMs) [36], [9] offer powerful open-world understanding and reasoning, providing a means to address both *sign* and *scene complexity*.

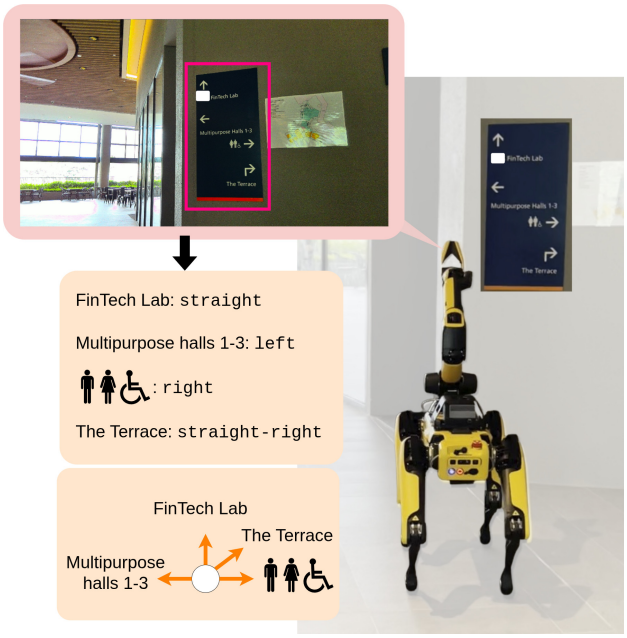


Fig. 1: **Sign understanding in the wild with Spot robot.** Navigational signs are aids for navigation and scene understanding. We introduce the *navigational sign understanding* task and design a baseline system to detect and parse signs on real hardware.

The main contribution of this paper is to introduce the task of **navigational sign understanding** and present a proof-of-concept system deployable on real hardware, enabling downstream applications in robot navigation and scene understanding. To support this task, we release an open-source benchmark combining a proposed set of evaluation metrics with a curated test set of signs from diverse public spaces (e.g. hospitals, malls, subway stations). We design a VLM-based baseline for detecting and interpreting signs. A key engineering challenge of this task is to select viewpoints that provide both clear visibility and alignment with the sign's intended viewing direction. To address this, we implement a ROS module integrating sign understanding with active viewpoint selection and demonstrate it on a quadruped robot.

†Smart Systems Institute, National University of Singapore.
‡Northeastern University
∗ Equal contribution.
[1] https://github.com/AdaCompNUS/Sign-Understanding


Fig. 2: **Dataset to evaluate navigational sign understanding.** The dataset captures diverse signs across a varied range of scenes. These span across hospitals, transit hubs, malls, campuses, outdoor parks etc., and include scenes with multiple signs, distractors and varying conditions, e.g. illumination. Signs in the dataset reflect real-world complexity, including signs using diverse symbols, signs that are stylized or have unusual appearance, or signs that are information-dense. The dataset's human-annotated ground truth provides the bounding boxes for observed signs. Human readable signs are further annotated with the corresponding navigational cues.

## II. Related Work

Previous research tackled the classification of indoor and traffic signs, as well as the recognition of text in multiple domains. While navigational signs are commonplace and part of everyday life for humans, no prior work addresses the detection and parsing of navigational signs, and specifically the complexity of associating open-set vocabulary and directional cues.

### A. Traffic Sign Recognition

The importance of traffic sign recognition (TSR) for intelligent vehicles have been recognized in previous decades [7], [27]. Earlier approaches used hand-crafted features to detect and classify the sign [6], [33], yielding subpar performance. The increased interest in autonomous vehicles, advancements in deep learning and the availability of larger datasets boosted the research in this domain, resulting in more robust performance [30], [35]. The task of traffic sign recognition includes detecting and classifying traffic signs using a set of predefined categories. It does not extend to parsing compound navigational signs that include both open vocabulary text, symbols and directional arrows.

### B. Indoor Sign Detection

The problem of indoor sign detection (ISD) differs from TSR in two speacts. First, traffic signs are standardized while indoor signage is not. Second, the placement of traffic signs allows full visibility and the signs' appearance is easily segmented from the background, while indoor signs are often surrounded by clutter and might be partially occluded [1]. Similarly to TSR, the earlier classic approaches [16], [31] were surpassed by learning-based approaches [4], [8]. As in the case of TSR, ISD remained limited to the detection and classification of a close set of classes, lacking the ability to extract spatial relationships between places represented as symbols and texts.

### C. Text Spotting

Text spotting, which refers to detecting text in an image and then recognizing the characters, has also experienced a leap in performance with the introduction of neural network architectures [18], [34]. Recent models such as PaddleOCR [10] and TesseractOCR [26] reliably extract textual information from standardized signs, but this alone is not sufficient to facilitate navigational sign understanding.

The existing works are unable to reason about the spatial relationship between the detected place indicators and the world. The current research on sign recognition is also limited to a closed set of classes. To the best of our knowledge, the sign understanding task was not formally formulated yet and no attempt at a principled solution was made. Under the task of sign understanding, we aim to bring together the detection and recognition of symbols and text, with open set labels and association between the detected place indicators and their relative location.

## III. Task

We introduce the task of **navigational sign understanding**, which extracts *navigational cues* from signs. Given an input RGB image, we seek to first *detect* navigational signs, then to *recognize* the navigational cues in each sign. A navigational cue is a specific location indicated on the sign, and the direction associated with it. Specifically, the two subtasks of navigational sign understanding are:

**Navigational sign detection:** To identify all navigational signs in a given RGB image input, $I$. The output is $\mathcal{B} = \{B_1, \ldots, B_N\}$, where $B_n$ is a 2D bounding box of a navigational sign in $I$. Specifically, navigational signs should be signs containing *both* location and direction information, thus excluding signs only containing locational information like room identification signs on doors.

**Navigational sign recognition:** To extract the set of navigational cues in an input navigational sign. The input sign is represented as an image crop $I_n$. The set of navigational

cues for a given sign indicating $M$ locations is:

$$\mathcal{C}_n = (p_1, l_1, d_1), \ldots, (p_M, l_M, d_M) \tag{1}$$

While locations in a sign may be indicated in both text or symbol form, we express all locations as plaintext strings for ease of representation. $p_m$ is a single unique location indicated on the sign, specified as a string. $l_m \in \{\text{text}, \text{symbol}\}$ identifies $p_m$'s original form in the sign. $d_m \in \mathcal{D}$ is the indicated direction that is associated with $p_m$. Across signs, directions tend to be expressed mainly with eight cardinal directions. We specify the set of possible directions, $\mathcal{D}$, as $\{\texttt{None}, \texttt{straight}, \texttt{right}, \ldots, \texttt{straight-left}\}$, where `None` denotes purely locational cues not associated to any direction.

Navigational sign understanding combines detection and recognition. Detection results provide image crops of each identified sign, and recognition further extracts the set of cues from each crop.

## IV. Dataset

We curate a dataset to benchmark navigational sign understanding, and illustrate it in Fig. 2. We consider two key dimensions of variation in this dataset: *scene complexity* and *sign complexity*.

**Scene complexity.** Environmental factors such as illumination, background textures, visual clutter, and occlusion affect perception of signs. Complexity also arises from non-navigational signs (e.g., regulatory signs) or sign-like objects (e.g., digital advertisements with arrows or location names) that can mislead sign detection.

**Sign complexity.** Navigational signs vary in appearance, shape, and structure. Such signs may contain multiple navigational cues of different types—locational (indicating a place) and directional (linking a place with a direction)—some of which may be presented in stylized or ambiguous ways, presenting difficulty in parsing these cues. For example, a single arrow may represent several destinations, requiring spatial or semantic reasoning to disambiguate. Signs also differ in formatting, symbols, and languages.

**Dataset structure.** Our dataset reflects scene and sign complexity across diverse environments. It comprises two splits to evaluate sign detection and sign recognition respectively. The **sign detection** split contains 160 RGB images of scenes ranging across hospitals, malls, and campuses, each with human-annotated bounding boxes of navigational signs, $B_i^{\text{gt}}$. The **sign recognition** split includes 205 cropped signs covering various aspects of sign complexity, including different stylizations, content formats, and variation in the use of symbols. Each navigational sign crop is annotated with navigational cues as tuples $[(p_1, d_1), \ldots, (p_T, d_T)]$, where $p$ is a location label and $d$ a direction category. We focus on English signs, noting that multilingual extensions are feasible with recent language models [10].

## V. Metrics

We describe metrics to benchmark the individual subtasks of navigational sign detection and recognition, and also for the overall task of navigational sign understanding.

**Detection metrics.** Detection is a specialized object detection task focusing on the category of signs that contain navigational information. To evaluate this, we employ standard object detection metrics, specifically COCO metrics [19] based on average precision (AP) and recall (AR).

**Recognition metrics.** Each sign in the dataset is annotated with ground-truth navigational cues, i.e. $\mathcal{C}_n$ for sign $n$. Predictions from sign understanding are matched to $\mathcal{C}_n$, and we report both aggregated and per-sign metrics. *Aggregated metrics* evaluate overall ability to parse cues, computed as precision and recall over all cues in the dataset. *Per-sign metrics* assess whether a system can fully parse individual signs, measured as the success rate of exactly recovering all cues from a sign.

Formally, let $\mathcal{G}$ be the set of ground-truth cues over all signs, where $s_n \in \mathcal{G}$ are cues for sign $n$, and $\mathcal{P}$ the predictions, with $p_n \in \mathcal{P}$ the predictions corresponding to sign $n$. Let $M_{\text{cue}}$ denote the total number of matched cues across the dataset, then:

$$\text{Precision} = \frac{M_{\text{cue}}}{\sum_{i=1}^{|\mathcal{P}|} |\mathcal{P}_i|}, \ \text{Recall} = \frac{M_{\text{cue}}}{\sum_{j=1}^{|\mathcal{G}|} |\mathcal{G}_j|} \tag{2}$$

The per-sign metric is defined as:

$$\text{Success Rate} = \frac{\sum_{i=1}^{|\mathcal{G}|} \mathbb{I}[s_i = p_i]}{|\mathcal{G}|} \tag{3}$$

where $s_i = p_i$ if there is a one-to-one match between predicted and ground-truth cues.

A match requires $l_t^{\text{gt}} = l_t$, $d_t^{\text{gt}} = d_t$, and $p_t^{\text{gt}}$ equivalent to $p_t$. For locations indicated with text, we define strict equivalence as an exact string match and relaxed equivalence as substring containment. We consider strict matching to be most representative of actual task performance since textual cues often encode specific locations (e.g., "Tower B"). For locations indicated as symbols, we consider two symbols equivalent if the cosine similarity of their language embeddings exceeds a threshold. Because text and symbol cues pose distinct challenges, we report precision, recall, and success rate separately for text and symbol cases, in addition to overall metrics.

**Overall metrics.** To assess overall performance of navigational sign understanding, we define a sign to be *understood* if it is both detected and if the extracted cues match the ground truth exactly. We further define **Precision**$_{\texttt{sign}}$ as the fraction of perfectly parsed signs over the predicted set of human readable signs, and **Recall**$_{\texttt{sign}}$ as fraction of perfectly parsed signs over the ground truth set of all human readable ground truth signs (obtained through Sec. VII-A).

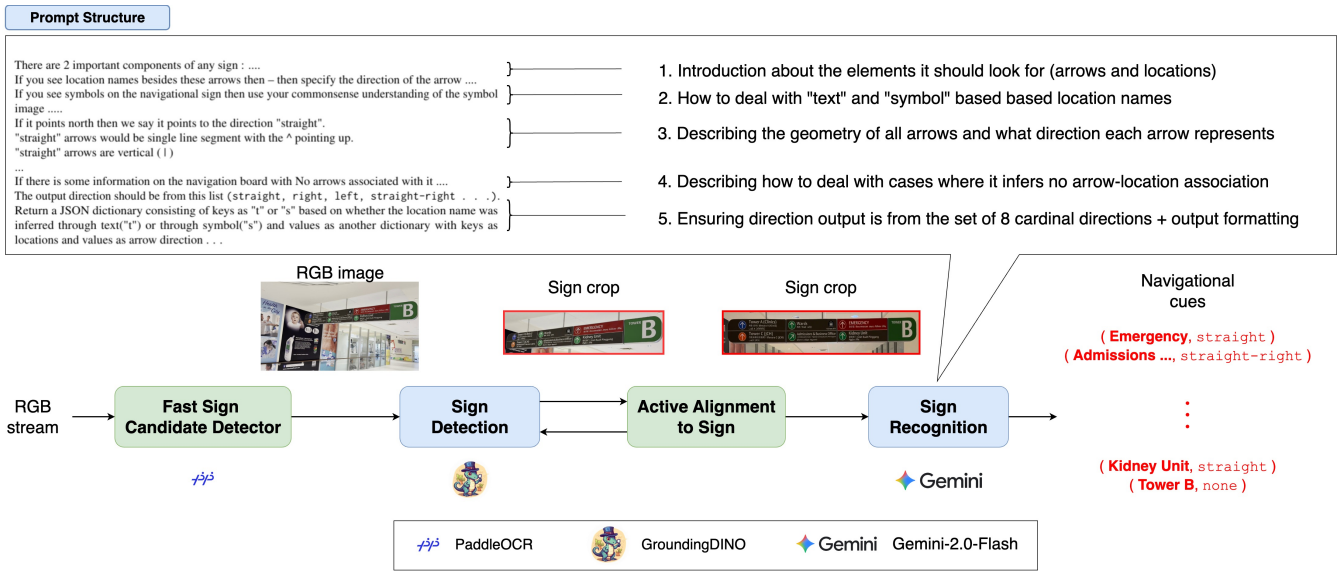


Fig. 3: **Baseline.** Blue denotes core modules required for sign understanding. Green modules are to integrate sign understanding into robot systems: an efficient, approximate sign filter, and an alignment module which physically servos the robot to align with the sign's canonical viewing direction and optimize the view.

## VI. Baseline

We introduce a baseline for navigational sign understanding (Fig. 3), that can be integrated on real robots for online operation. Our baseline relies on VLMs to detect navigational signs and extract navigational cues from them. For real robot integration, we add **(i)** an efficient filter to reduce calls to computationally expensive sign detection, **(ii)** a servoing step that aligns the robot (and sensor) with the sign head-on.

Alignment is important for two reasons. Firstly, signs are designed to be viewed head-on, and the directions contained within are expressed with respect to this canonical viewing direction. For a robot to interpret a sign and take actions informed by it, the view angle must be accounted for. Secondly, we find empirically that VLMs' results are unreliable at oblique viewing angles.

### A. Navigational Sign Understanding

We use an open-set object detector to perform navigational sign detection. Specifically, we prompt GroundingDINO [20] to identify "navigational signs", returning 2D bounding boxes which we convert into image crops of each sign. The image crops are passed to a VLM that is prompted to parse text and symbols from a sign, and reason about the relationships between the entities on the sign.

Given a sign crop, we query a VLM with the crop and a structured prompt to guide it in parsing the sign. Fig 3 highlights the structure of the prompt used. The prompt guides the VLM to **(i)** extract all location and place-related text from the image, **(ii)** extract all directional symbols from the image, **(iii)** associate locations and directions, **(iv)** output the associated directional cues as a list of tuples as described in Sec. III.

### B. Integrating into robot systems

Online sign understanding should be efficient and onboard where possible, and also ensure physical alignment of the robot with respect to the sign for optimal viewing.

As current open-set object detectors run at low frame rates on embedded hardware, we add a lightweight filter that selects frames likely to contain candidate navigational signs. Based on the heuristic that navigational signs likely contain text, we run a fast OCR model (PaddleOCR [10]) to select only frames with text to run the detector on. If specific text strings of interest are known beforehand (e.g. room names from a floor plan), we use fuzzy string matching to select frames containing these strings.

We aim to physically align the robot with the sign's viewing direction and ensure a close-up view that ensures the entire sign is visible for recognition. Using camera intrinsics, we apply a monocular depth estimation model (Metric3Dv2 [14]) to generate an aligned depth map from RGB input. We obtain the centroid and surface normal of the sign, by estimating a best-fit plane using least squares on the sign's depth crop. From this, we compute a target position along the normal that both guarantees full visibility of the sign and maximizes its coverage in the image.

## VII. Experimental Evaluation

We consider the following questions:

**Q1.** How accurately does our dataset capture the semantic information in navigational signs?
**Q2.** How effective is our baseline at *navigational sign detection*?
**Q3.** How effective is our baseline at *navigational sign recognition*?
**Q4.** How effective is our baseline at the overall task of *navigational sign understanding*?
**Q5.** What aspects of scene or sign complexity are challenging for the baseline?

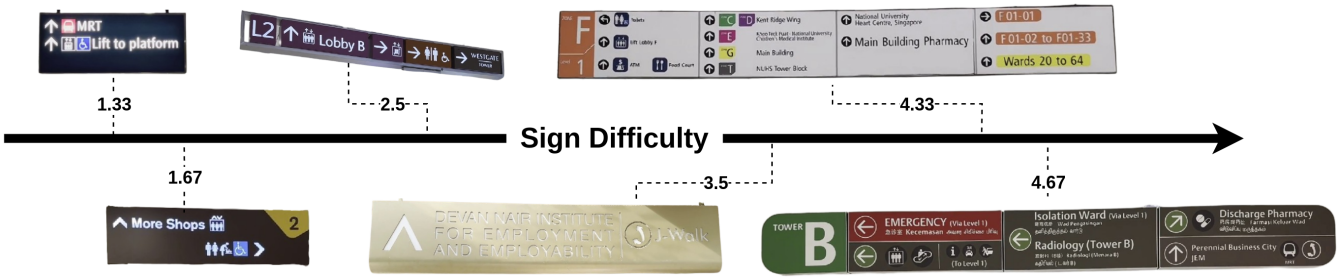


Fig. 4: **Human perspective.** Sign recognition difficulty score (increasing left to right) for different signs ranked by the participants in the user study. Content overload, ambiguous place-arrow associations and nonstandard style contribute to the perceived difficulty of recognizing navigational signs.

### *A. Sign Understanding from a Human Perspective*

The task of sign understanding is efficiently performed by humans every day, and therefore we can safely use human performance as a safe heuristic to investigate **Q1**. Since unlike most learning approaches, human annotators can explain what factors contribute to the difficulty of sign detection and recognition – we asked them to enumerate their reasons for the same. This analysis helped us illuminate the challenges inherent in the task.

We conducted a study with ten participants. We tasked participants to rate the difficulty of detecting/recognizing signs and qualitatively describe factors affecting their ability to do so, based on fifteen signs sampled from the sign detection split. To verify correctness of ground-truth annotations for sign recognition, we also tasked participants to manually annotate six signs drawn the sign recognition split, deemed to be challenging (scored at difficulty $\geq 4$).

Fig. 4 shows signs from the survey, with average difficulty score given by participants. We find that sign detection is most affected by **(a)** sign placement and **(b)** sign size. For sign recognition, participants highlight **(a)** ambiguous place-arrow associations, **(b)** content overload and **(c)** nonstandard stylizations of symbols/arrows as key challenges in parsing. To understand if signs perceived as difficult by participants are also challenging for our baseline, we evaluate our baseline on the set of signs annotated by participants. We see lower recognition success rate ($\sim 15\%$) compared to the recognition performance in Sec. VII-C ($\sim 40\%$), indicating that VLMs indeed struggle with such signs.

We find that participants' annotations have a high degree of agreement with each other. The annotations proposed by the participants were compared against the ground truth annotation, and found to be matching in 95.5% of the labels. This showcases a high level of consensus between the annotations and the ground truth, indicating that the ground truth annotations captures the semantic information encoded in the navigational signs. This also gave us the consensus on the notion of "human readability" of a sign. We further use this in analyzing **Q4**.

### *B. Navigational Sign Detection Performance*

We address **Q2** by evaluating on the sign detection split of our dataset. Following the metrics in Sec. V, Tab. I reports overall detection performance, while Tab. II breaks down performance by sign size. Following COCO convention, we define signs in three sizes based on the bounding box area: Small (S) have $< 32^2$ pixels, Medium (M) have $> 32^2$ pixels and $< 96^2$ pixels and Large (L) has $> 96^2$ pixels. We compare detection performance of specialized object detectors (GroundingDINO [20]) with general VLMs (Gemini-2.0-Flash [29]).

TABLE I: **Evaluating navigational sign detection:** We evaluate detection with AP and AR at varying IoU thresholds.

| Metric | MaxDets | GroundingDINO | Gemini-2.0-Flash |
|---|---|---|---|
| AP@[IoU=0.50] (all) | 100 | 0.673 | 0.289 |
| AP@[IoU=0.75] (all) | 100 | 0.582 | 0.125 |
| AP@[IoU=0.25:0.75] (all) | 100 | 0.657 | 0.288 |
| AR@[IoU=0.25:0.75] (all) | 1 | 0.422 | 0.333 |
| AR@[IoU=0.25:0.75] (all) | 10 | 0.837 | 0.529 |
| AR@[IoU=0.25:0.75] (all) | 100 | **0.890** | 0.529 |

TABLE II: **Evaluating size based navigational sign detection** Evaluation of detection on the basis of navigational sign size.

| Metric | Size | MaxDets | GroundingDINO | Gemini-2.0-Flash |
|---|---|---|---|---|
| AP@[IoU=0.25:0.75] | S | 100 | 0.105 | 0.003 |
| AP@[IoU=0.25:0.75] | M | 100 | 0.190 | 0.039 |
| AP@[IoU=0.25:0.75] | L | 100 | 0.774 | 0.385 |
| AR@[IoU=0.25:0.75] | S | 100 | 0.513 | 0.021 |
| AR@[IoU=0.25:0.75] | M | 100 | 0.723 | 0.173 |
| AR@[IoU=0.25:0.75] | L | 100 | **0.949** | 0.637 |

Specialized object detectors outperform general VLMs on sign detection by a wide margin. Both models tend towards over-predicting signs, showing consistently higher recall than precision across all tests. Qualitatively, we find that models may have difficulty isolating *navigational* signs from the variety of other signs present in human environments. (See Fig. 6) We also note that sign size strongly impacts the models' ability to positively identify a sign, with precision jumping significantly from medium to large signs for both models. We surmise that navigational signs may only be accurately distinguished from non-navigational signs on the basis of their information content, resulting in the models' performance improving dramatically for close-up views where the signs' content is clearly visible. We provide more detailed analysis and examples in Sec. VII-E.

### *C. Navigational Sign Recognition Performance*

We address **Q3** by evaluating on the sign recognition split of our dataset, using the recognition metrics (See

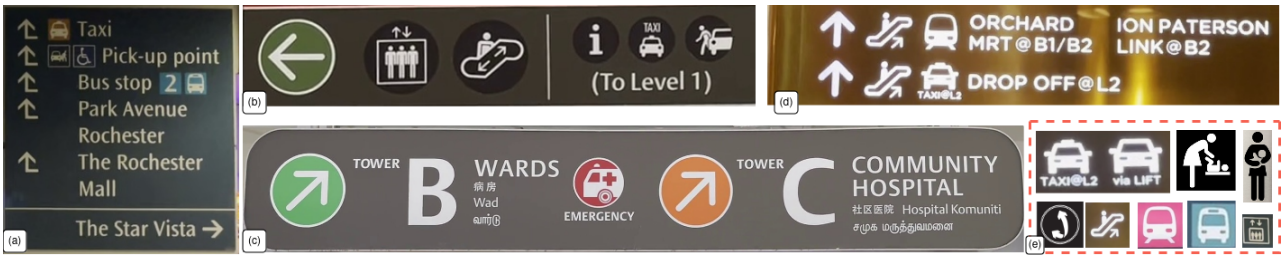


Fig. 5: **Sign Recognition.** Some of the common types of failure cases and complexities observed in the task of sign recognition


Fig. 6: **Sign Detection.** An example annotation obtained by GroundingDINO. Here we observe that the confidence is higher for larger signs whereas the confidence is lower for the smaller, more distant signs. As compared to other model, the coherence in selecting signs was more evident in GroundingDINO.

TABLE III: **Ablations for navigational sign recognition:** Evaluation of baseline's performance in parsing text and symbols, using precision/recall.

| Model | Precision | | | Recall | | |
|---|---|---|---|---|---|---|
| | Txt (E) | Txt (S) | Sym | Txt (E) | Txt (S) | Sym |
| GPT-4o | 0.79 | 0.80 | 0.82 | 0.59 | 0.60 | 0.70 |
| Gemini-2.0-Flash | 0.76 | 0.77 | 0.80 | 0.66 | 0.68 | 0.67 |

TABLE IV: **Evaluating navigational sign recognition:** Evaluation of the overall success rate of recognition at a per-sign-level. A sign is defined as fully recognized ("overall" columns) if **all** predicted cues (both symbol and text) match the ground truth cues. We also report the fraction of textual cues and symbolic cues correctly recognized at a per-sign level in the "Txt/Sym Success Rate" columns.

| Model | Txt/Sym Success Rate | | | Success Rate | |
|---|---|---|---|---|---|
| | Txt (E) | Txt (S) | Sym | Overall (E) | Overall (S) |
| GPT-4o | 44.6 | 44.6 | 63.3 | 39.3 | 39.3 |
| Gemini-2.0-Flash | 47.3 | 47.9 | 65.8 | 41.6 | 42.2 |

Sec. V). This split comprises curated sign crops observed with approximately head-on orientation, with legible text and symbols. Following metrics in Sec. V, Tab. III reports *aggregated metrics* and Tab. IV *per-sign metrics*. For matching symbols, we employ CLIP-based cosine similarity. For matching text, we show results for both exact (E) and substring (S) approaches. If multiple predictions are substrings of a single ground-truth navigational cue, we consider only the match with the longest substring. We compare both GPT-4o [15] and Gemini-2.0 Flash [29] VLMs in our evaluations.

TABLE V: **Ablations for navigational sign understanding:** Evaluation of different combinations of models for the combined task of detection and recognition. We report the precision at a sign level and recall at a sign level. The first model refers to the model that generates bounding boxes and the second model is used to parse those crops. We only allow matching of bounding boxes which have a minimum of 0.5 IoU with any of the ground truth bounding box

| Models | | | |
|---|---|---|---|
| Detection | Recognition | $\text{Precision}_{\texttt{sign}}$ | $\text{Recall}_{\texttt{sign}}$ |
| Gemini-2.0-Flash | GPT-4o | 31.0 | 24.2 |
| Gemini-2.0-Flash | Gemini-2.0-Flash | 39.2 | 30.3 |
| GroundingDINO | Gemini-2.0-Flash | 39.3 | 39.0 |
| GroundingDINO | GPT-4o | 35.3 | 35.1 |

Strong performance on aggregated precision and recall indicates that existing VLMs can capably parse locational information (text and symbols) and perform the semantic reasoning needed to associate them with the right directions. Gemini-2.0-Flash generally outperforms GPT-4o on sign recognition, fully predicting signs accurately more often (Tab. IV). We found GPT-4o to be more conservative and infer only subsets of cues in the sign, leading to significantly lower recall than Gemini-2.0-Flash. Qualitatively, we find that VLMs can often generate lexically correct answers, but tend to struggle with parsing text that requires contextual understanding—e.g. text spread across multiple lines and hence requiring analysis of semantic continuity to understand. However, inferring symbolic locations does not require such contextual awareness, and thus we observe a performance gap between inferring textual and symbolic cues at both *aggregated* and *per-sign level*.

### *D. Navigational Sign Understanding Performance*

We evaluate the full baseline (Sec. VI), which both detects signs and extracts navigational cues, to address **Q4**. Test inputs are drawn from the sign detection dataset split. We highlight that the dataset only annotates signs with legible text and symbols, and our metrics are computed based on this ground-truth. Tab. V presents *overall metrics* for variants of the baseline using different models for the sub-tasks of detection and recognition. Consistent with above results, GroundingDINO continues to show strong performance on detecting navigational signs, while Gemini-2.0-Flash exhibits better performance in extracting navigational cues. We note that Gemini-2.0-Flash exhibits poorer recognition performance on signs cropped from the detection dataset split


(a) (b) (c) (d) (e) (f)

Fig. 7: **Demo of plug-in module on Spot. (a, b)**: Spot explores the environment. **(c)**: Fast detector triggers sign detection, and overrides exploration upon positive detection. **(d, e)**: Plug-in module servos to face sign head-on, and arm makes adjustments to fully view sign. **(f)**: Successful parsing results obtained; Spot resumes exploration after.

compared to evaluations on the curated recognition dataset split, likely owing to more variation in viewing angles and distance, and bounding box mis-detections. This highlights the need for good viewpoint selection for in-the-wild sign understanding.

### E. Discussion

We address **Q5** with a qualitative analysis of the common failure modes of navigational sign detection and recognition.

For *navigational sign detection*, the main failure mode lies in semantically understanding whether the sign contains navigational cues. Empirically, we observe existing models like GroundingDINO perform well at identifying signs in general. However, distinguishing signs containing *navigational cues* requires understanding of a sign's semantic content. Thus, detecting navigational signs is a challenge when the content of the sign is unclear (e.g. due to distance) or when the content itself is semantically ambiguous (e.g. large advertisement containing arrows and text).

For *navigational sign recognition*, failure modes include:

- **Arrow association.** There is no guarantee of a one-to-one mapping between an arrow and location on a navigation sign. While it easy for us humans to infer such associations, models struggle to infer the intended direction from subtle context or commonsense cues (e.g. Fig. 5(b) includes multiple symbols associated with a single arrow).
- **Symbols with arrows.** Confusion arises when arrows appear as part of a symbol (e.g. elevator or escalator icons).
- **Multi-line names.** Spatial formatting leads models to split or merge location names incorrectly (e.g. Fig. 5(a) "Park Avenue Rochester" parsed as multiple entities).
- **Overcrowded signs.** Dense layouts with small text or symbols increase misclassification and incorrect text–direction associations.
- **Ambiguous symbols.** Even for humans, certain symbols (e.g. Fig. 5(e) car park vs. taxi pickup) are difficult to disambiguate, and models show similar confusions.
- **Combinatorial understanding.** Correct interpretation often requires reasoning over text–symbol combinations (e.g. Fig. 5(a) "Wheelchair Accessible Car Pickup Point"), which goes beyond element-wise parsing.

## VIII. Applications

We provide a plug-in module for online sign understanding on real robot hardware. We conclude by discussing potential downstream applications using semantic information from signs, which our plug-in sign understanding module enables.

### A. Sign Understanding on Real Robots

We demonstrate the plug-in sign understanding module (Sec. VI) on a Spot robot. Spot explores an environment (Fig. 7) while running the module onboard a Jetson Orin, using RGB input from the gripper camera and querying the Gemini-2.0-Flash VLM over 4G. A fast detector filters the RGB stream at 10Hz to identify candidate signs, which are then passed to the sign detector. Upon detecting a sign, the module overrides Spot's navigation policy, and iteratively servos toward the sign to optimize angle and distance, with the arm making fine adjustments to optimize the view when close to the sign. The entire process completes within 20s.

### B. Downstream Applications

Textual cues have been explored for localization [37], [5], mapping [2], and navigation [3], but prior work largely treated text as landmarks for map matching, overlooking the spatial information in directional signs. Talbot et al. [28] highlighted their potential but offered no method to parse signs. Liang et al. [17] obtained navigation policies from navigational signs, but made strong assumptions in parsing signs. In contrast, we propose a robust pipeline for sign understanding and a plug-in module for downstream applications.

Navigational signs contain information about places and their relative direction, which makes them useful for localizing in prebuilt maps that contain semantic and textual information. The place labels and associated directions are a constellation of abstract objects [23] that can be matched against the map to localize the robot. Similarly, for SLAM, navigational signs are good candidates for loop closures. Navigational signs are visually salient features, and their

spatial semantic information can be used to verify the partially constructed map and the pose estimation. Navigational signs extend scene understanding beyond line-of-sight, and their compact semantic content can be easily integrated into representation such as scene graphs [13], [21], assisting with object-goal navigation and more efficient exploration. The recent focus on topological and topo-semantic navigation [21], [24], as well as language goal specification [25], highlights the significance of navigational signs, due to their topo-semantic nature and their similarity to natural language. Further improvement in sign understanding can support navigational signs as a policy for navigation, increasing the impact and adoption of works such as Liang et al. [17].

Overall, navigational signs are a rich, underutilized source of semantic information that can greatly expand scene understanding capabilities of robots in human-oriented environments.

## REFERENCES


[1] J.L. Almeida, F.C. Flores, M.N. Roecker, M.A. Braga, and Y.M. Costa. An indoor sign dataset (isd): An overview and baseline evaluation. *VISIGRAPP (4: VISAPP)*, pages 505–512, 2019.

[2] C. Case, B. Suresh, A. Coates, and A.Y. Ng. Autonomous sign reading for semantic mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[3] C. Chen, L. Lu, L. Yang, Y. Zhang, Y. Chen, R. Jia, and J. Pan. Signage-Aware Exploration in Open World using Venue Maps. *IEEE Robotics and Automation Letters (RA-L)*, 2025.

[4] S.A. Cheraghi, G. Fusco, and J.M. Coughlan. Real-time sign detection for accessible indoor navigation. In *Journal on technology and persons with disabilities:... Annual International Technology and Persons with Disabilities Conference*, volume 9, page 125, 2021.

[5] L. Cui, C. Rong, J. Huang, A. Rosendo, and L. Kneip. Monte-Carlo Localization in Underground Parking Lots Using Parking Slot Numbers. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[6] A. De la Escalera, J.M. Armingol, and M. Mata. Traffic sign recognition and analysis for intelligent vehicles. *Journal on Image and Vision Computing (IVC)*, 21(3):247–258, 2003.

[7] A. De La Escalera, L.E. Moreno, M.A. Salichs, and J.M. Armingol. Road traffic sign detection and classification. *IEEE transactions on industrial electronics*, 44(6):848–859, 1997.

[8] F.C.T. De Vera, A.C. De Vivar, and J.J.R. Balbin. Indoor Signage and Arrow-Icon-Text Detection and Localization with YOLOv5. In *8th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, 2024.

[9] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J.S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[10] Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang, et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020.

[11] S. Garg, K. Rana, M. Hosseinzadeh, L. Mares, N. Sünderhauf, F. Dayoub, and I. Reid. Robohop: Segment-based topological map representation for open-world visual navigation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[12] D. Gibson. *The wayfinding handbook: Information design for public places*. Princeton Architectural Press, 2009.

[13] Q. Gu, A. Kuwajerwala, S. Morin, K.M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[14] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen. Metric3D v2: A Versatile Monocular Geometric Foundation Model for Zero-Shot Metric Depth and Surface Normal Estimation. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 46:10579–10596, 2024.

[15] A. Hurst, A. Lerer, A.P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[16] D. Kunene, H. Vadapalli, and J. Cronje. Indoor sign recognition for the blind. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, 2016.

[17] C. Liang, R.A. Knepper, and F.T. Pokorny. No map, no problem: A local sensing approach for navigation in human-made spaces using signs. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6148–6155. IEEE, 2020.

[18] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai. Real-time scene text detection with differentiable binarization. In *Proc. of the Conference on Advancements of Artificial Intelligence (AAAI)*, 2020.

[19] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 740–755, 2014.

[20] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2024.

[21] J. Loo, Z. Wu, and D. Hsu. Open scene graphs for open-world object-goal navigation. *Intl. Journal of Robotics Research (IJRR)*, 2025.

[22] R. Mascaro and M. Chli. Scene representations for robotic spatial perception. *Annual Review of Control, Robotics, and Autonomous Systems*, 8, 11 2024.

[23] A. Ranganathan and F. Dellaert. Semantic modeling of places using objects. In *Proc. of Robotics: Science and Systems (RSS)*, 2007.

[24] D. Shah and S. Levine. Viking: Vision-based kilometer-scale navigation with geographic hints. *arXiv preprint arXiv:2202.11271*, 2022.

[25] D. Shah, B. Osiński, S. Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Proc. of the Conf. on Robot Learning (CoRL)*, pages 492–504. PMLR, 2023.

[26] R. Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[27] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[28] B. Talbot, F. Dayoub, P. Corke, and G. Wyeth. Robot navigation in unseen spaces using an abstract map. *IEEE Transactions on Cognitive and Developmental Systems*, 13(4):791–805, 2020.

[29] G. Team, R. Anil, S. Borgeaud, J.B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A.M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[30] J. Wang, Y. Chen, Z. Dong, and M. Gao. Improved yolov5 network for real-time multi-scale traffic sign detection. *Neural Computing and Applications*, 35(10):7853–7865, 2023.

[31] S. Wang, X. Yang, and Y. Tian. Detecting signage and doors for blind navigation and wayfinding. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 2:81–93, 2013.

[32] W. Wang. *Please follow the signs:: Considering existing navigational aids in indoor navigation services*. PhD thesis, Technische Universität Wien, 2023.

[33] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma. A hybrid vehicle detection method based on viola-jones and hog+ svm from uav images. *Sensors*, 16:1325, 2016.

[34] Z. Yang, Z. Li, X. Jiang, Y. Gong, Z. Yuan, D. Zhao, and C. Yuan. Focal and global knowledge distillation for detectors. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[35] J. Zhang, X. Zou, L.D. Kuang, J. Wang, R.S. Sherratt, and X. Yu. Cctsdb 2021: a more comprehensive traffic sign detection benchmark. *Human-centric Computing and Information Sciences*, 2022.

[36] J. Zhang, J. Huang, S. Jin, and S. Lu. Vision-language models for vision tasks: A survey. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 2024.

[37] N. Zimmerman, L. Wiesmann, T. Guadagnino, T. Läbe, J. Behley, and C. Stachniss. Robust Onboard Localization in Changing Environments Exploiting Text Spotting. *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.