Object-Centric Grounding for Deployable and Interactive Vision-Language Navigation Agents

Haochen Zhang

CMU-RI-TR-25-86

Sept 2, 2025

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Ji Zhang, *Co-chair*
Wenshan Wang *Co-chair*
Yonatan Bisk
Bowen Li

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my parents and sister.

Abstract

Robots that operate in human-centric environments must integrate perception, reasoning, and action across multiple modalities to complete tasks according to user instructions. For these robots, being able to navigate according to a natural language instruction about the environment is an important capability, which requires 3D spatial reasoning, semantic scene understanding, and the ability to handle vague or ambiguous instructions. Additionally, the diverse and noisy nature of real-world environments motivates the need for vision-language navigation (VLN) systems that are robust and able to adaptively generalize.

This thesis makes two main contributions toward robust, interactive vision-language robotic systems by focusing on the underlying task of 3D object-centric grounding. First, we introduce IRef-VLA, a large-scale 3D benchmark with millions of referential statements and semantic relations, including imperfect language, to support the evaluation of models for 3D scene understanding. Second, we propose SORT3D, a modular framework for grounding object-referential language in 3D by leveraging large vision and language models, heuristic spatial reasoning, and 2D features, achieving zero-shot generalization to unseen environments and real-time operation on autonomous ground vehicle systems. Furthermore, we explore future directions for interactive, dialogue-enabled vision-language navigation by formulating the problem, exploring existing benchmarks and laying the groundwork for future work in this area. Together, these contributions advance general navigation systems that are capable of semantic scene understanding and communicating with human users in complex, real-world settings.

Acknowledgments

First and foremost, I'd like to thank my advisors, Dr. Wenshan Wang and Dr. Ji Zhang. Thank you for taking me in as a Master's student with little background in field robotics, trusting me to lead projects, and providing me with both research support and opportunities over the past two years. The knowledge and guidance you have provided me on how to be a researcher are never taken for granted.

I'd also like to thank the other members on my committee, Dr. Yonatan Bisk and Bowen Li for inspiring my work as a researcher, always taking time out of your busy schedules to discuss research or offer advice, and being so willing to help me learn. I look forward to learning more from you both in the future.

Thank you Pujith Kachana and Nader Zantout for being amazing teammates and collaborators for the past two years. Much of the work in this thesis is work that we did together and none of it would have happened without either of you. It's been a pleasure sharing this MSR journey with you.

Thank you also to the lab members of the Zhang Lab and AirLab who have given me advice, chatted with me about research, inspired ideas, and overall made my time in the lab more collaborative and more fun.

Finally, to my family, friends, and loved ones, thank you for the endless support, advice, and cheerleading. I truly could not have done this without any of you. You know who you are :). To my parents especially – I am forever grateful for the sacrifices you have made for me to get to where I am now and for being my rock through it all. I hope to make you proud.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

In the pursuit of generalizable and deployable robots that can operate with humans, robotic systems must be capable of understanding human environments and interacting with natural language. These systems must integrate perception, reasoning, and action, while aligning with user instructions and preferences. Advances in large language models (LLMs), vision-language models (VLMs), and other foundation models in recent years have led to a significant shift in progress towards leveraging internet-scale data to develop embodied agents [4, 54, 55]. These models have created new opportunities for robots that can interpret natural language instructions and operate in complex, real-world settings by aligning what they perceive with pretrained semantic knowledge. One domain of application is **Vision and Language Navigation (VLN)**, where an agent has to navigate according to a natural language instruction. This includes tasks such as object-goal navigation (e.g. "Go to the grand piano") and instruction following (e.g. "Go past the couch to the kitchen and stop near the fridge"). While such statements are trivial for adult humans to comprehend and follow, this task is surprisingly difficult for the commercially available robots we have today, such as the Amazon Astro[1] or Roomba[2].

The development of VLN systems that can be reliably deployed in human-centric environments, comes with a number of core challenges. First, these tasks often require precise spatial understanding and semantic reasoning in 3D. While large

[1]https://www.aboutamazon.com/news/devices/meet-astro-a-home-robot-unlike-any-other
[2]https://www.irobot.com/en_US/roomba.html

vision-language models (VLMs) have excelled at this in the 2D space [56], reasoning in 3D space is much more challenging due to noisier data, varying viewpoints, and inherent complexity of structure. Furthermore, the scale of available 2D data used for 2D vision-language models is much larger than what is available in 3D [12], while practical use cases often involve fine-grained objects in large, multi-room scenes. Second, VLN tasks often involves the subtask of **object-referential grounding**, which is defined as disambiguating the target object from language instructions that refer to relative relationships between objects to specify the target when multiple candidate objects exist (e.g. "Go to the chair closest to the door"). While such tasks are trivial for humans, they require real-time spatial and situated reasoning in a partially observable and dynamic environment, which is challenging for embodied agents. Finally, the dynamic and noisy nature of the real world brings uncertainty into the task. In particular, human instructions for grounding or navigation may be misaligned with the scene, or be ambiguous to the agent, necessitating iterative dialogue capabilities to seek clarification or ask for help.

This thesis addresses these challenges through three complementary contributions, focusing on object-centric grounding for downstream VLN tasks. First, in Chapter 2 we introduce IRef-VLA, to scale up existing 3D object-referential data in real-world environments. IRef-VLA is a **large-scale benchmark for referential grounding in 3D scenes**, combining over 11.5K scanned rooms with millions of semantic relations and referential statements, including deliberately ambiguous or imperfect language, relational scene graphs and navigable space annotations. This resource provides a foundation for evaluating models that reason about objects, relations, and navigable space in diverse real-world 3D environments.

Second, in Chapter 3 we present SORT3D, to tackle online object-centric spatial grounding in unseen environments towards creating VLN agents that are **deployable** in the real world. SORT3D is a modular **real-time framework for grounding object-referential language** in 3D with no training data. By leveraging rich 2D object attributes, heuristic spatial reasoning, and the sequential reasoning capabilities of LLMs, SORT3D achieves zero-shot generalization to unseen environments in both simulation and the real-world. We propose SORT3D as a practical and easily adaptable method for VLN, demonstrating the feasibility of language-guided object-goal navigation in real-world scenarios.

Finally, in Chapter 4 we explore the frontier of **interactive** and dialogue-based vision-language navigation, focusing on **ambiguity resolution**. While many existing methods including SORT3D assume complete and unambiguous instructions, real-world human guidance can be vague, incomplete, or misaligned. Interactive approaches where agents are able to communicate uncertainty and ask clarifying questions bring the potential for richer and more robust human-robot interactions, but introduce challenges in training data, evaluation, and modeling uncertainty. This thesis outlines the opportunities and initial directions for building dialogue-capable agents that adaptively handle ambiguity in navigation tasks.

Together, these contributions advance the development of robust and adaptive VLN systems that can perceive, reason, and act in human-centric environments.

Chapter 2

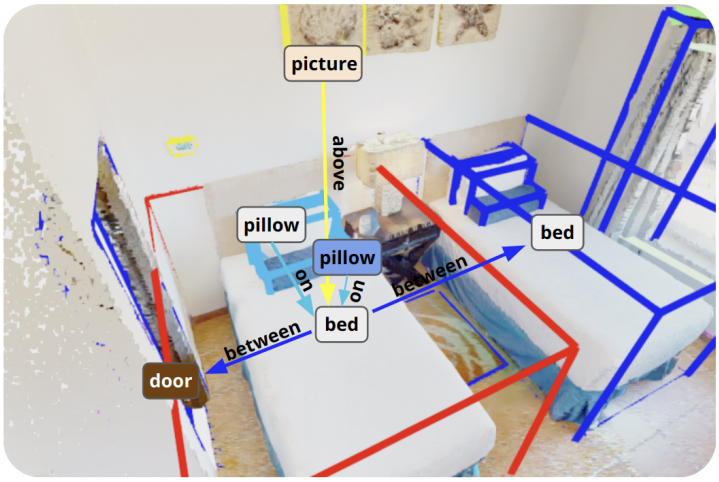
Diversifying Benchmarks for 3D Referential Grounding

2.1 Abstract

With the recent rise of large language models, vision-language models, and other general foundation models, there is growing potential for multimodal, multi-task robotics that can operate in diverse environments given natural language input. One such application is indoor navigation using natural language instructions. However, despite recent progress, this problem remains challenging due to the 3D spatial reasoning and semantic understanding required. Additionally, the language used may be imperfect or misaligned with the scene, further complicating the task. To address this challenge, we curate a benchmark dataset, IRef-VLA, for Interactive Referential Vision and Language-guided Action in 3D Scenes with imperfect references. IRef-VLA is the largest real-world dataset for the referential grounding task, consisting of over 11.5K scanned 3D rooms from existing datasets, 7.6M heuristically generated semantic relations, and 4.7M referential statements. Our dataset also contains semantic object and room annotations, scene graphs, navigable free space annotations, and is augmented with statements where the language has imperfections or ambiguities. We verify the generalizability of our dataset by evaluating with state-of-the-art models to obtain a performance baseline and also develop a graph-search baseline to demonstrate

the performance bound and generation of alternatives using scene-graph knowledge. With this benchmark, we aim to provide a resource for 3D scene understanding that aids the development of robust, interactive navigation systems. The dataset and all source code is publicly released[1].

2.2 Introduction



(a) Scene with scene graph

"The white bed that is between the other bed and the door frame"

(b) Referential statement

Figure 2.1: Sample region from the dataset visualized with (a) a scene graph and (b) a corresponding referential statement

.

Methods combining vision and language have been evolving rapidly with the advent of both Large Language Models (LLMs) [2, 65, 70] and Vision-Language Models (VLMs) [40, 48, 50] pre-trained on internet-scale data, tackling various 2D tasks such as Visual Question Answering (VQA) [7], image retrieval [35], and image captioning [48]. As we progress towards generalizable embodied intelligence, there is a need

[1]https://github.com/HaochenZ11/IRef-VLA

for methods that are capable of reasoning in 3D-space and interacting with humans. Particularly, being able to understand natural language and ground that language to the physical world are key skills for interactive robots. For example, humans are able to precisely refer to objects in a 3D scene, often using the statement of "least effort" [85] and making use of spatial relationships. An agent that can 1) similarly solve such a problem, 2) handle imperfect or ambiguous language, and 3) interact with humans to achieve the intended goal would be valuable in robotics fields such as indoor-navigation with applications as in-home assistants.

The pursuit of such agents that can identify and understand 3D scenes, consolidate visual input with language semantics, and display robust performance for real-world deployment, however, presents various challenges. First, the scene can have hundreds of objects, contain objects belonging to fine-grained classes, and have many similar objects [49]. Second, human referential language often involves spatial reasoning, implicit and explicit affordances, open-vocabulary language, and may even be incorrect or refer to something that does not exist, e.g. *"the remote on the table"* when the remote is actually on the sofa. Third, the scale of available vision-language data in the 3D space pales in comparison to the amount of 2D data, which was crucial to the success of 2D vision-language learning methods [11, 36]. Despite impressive recent advancements with foundation models, such problems remain difficult when applied to robotics as current methods fail to offer the accuracy and robustness needed for real-world deployment [28].

To advance the path towards more intelligent interaction in natural language navigation, we propose the IRef-VLA dataset as a benchmark for both the referential object-grounding task, and a novel extension of this task we call **referential grounding with imperfect references**. First, we provide the largest real-world dataset based on 3D scenes from a diverse set of existing indoor scans. Our dataset includes 1) segmented scene point clouds to enable learning directly from 3D visual information, 2) object-level attributes, semantic class labels, and affordances, 3) dense scene graphs with spatial relations as structural guidance, 4) heuristically-generated referential statements improving upon previous datasets, 5) traversable free space annotations allowing for references to areas and spaces, and 6) augmented "imperfect" referential statements to benchmark grounding with imperfect language. In particular, the inclusion of scene graphs, free space annotations, and imperfect statements distinguishes

our dataset from previous ones. Second, with the inclusion of imperfect language, we define the extended task of referential grounding with imperfect references, testing a model's capability to a) detect when a specific referenced object does not exist in the scene, and b) prompt interaction by generating valid alternative suggestions. A sample from our dataset is shown in Figure 2.3.

To validate our dataset, we train two state-of-the-art (SOTA) supervised referential grounding models on our dataset and demonstrate generalizability to test benchmarks. We also implement our own graph-search method that first determines whether an object is in the scene, then suggests alternatives if needed. We compare performance to an augmented SOTA model that classifies object existence. We release our dataset, source code for generation and baselines, and a dataset visualization tool publicly.

2.3 Related Work

Object Referential Datasets

The referential object-grounding task has been defined and explored in 3D datasets such as ReferIt3D [3], ScanRefer [13], and SceneVerse [33]. While ReferIt3D and ScanRefer establish a benchmark for referential grounding on one set of scenes providing both synthetically generated (Sr3D) and human-uttered (Nr3D) statements [3], ambiguities exist in the synthetic statements and the human utterances can be subjective and unintuitive, e.g. using the word "*comfy*" or using clock bearings to describe objects. SceneVerse scales the data up by curating a much larger dataset and generating statements synthetically, then using an LLM for rephrasing, though both templated and LLM-rephrased statements are often unnatural and lack explicit references to attributes like size, color, and shape which humans often use for object reference. As a result, models trained on SceneVerse still performed poorly on the Nr3D benchmark [33]. More recently, SpatialRGPT [16] proposes a data pipeline to further push the scale of 3D-grounded referential statements using existing large VLMs, although it is limited to 2D images and does not directly contain 3D data.

Semantic Scene Graph Datasets

Generating scene graphs from 3D scenes has also been explored in 3DSSG [71], Hydra [29], HOV-SG [74], and ConceptGraphs [23]. 3DSSG focuses on predicting scene graphs automatically, resulting in generated graphs that can miss relations or generate redundant ones, which requires more processing to disambiguate objects given their relations. In Hydra, a system is developed to build 3D scene graphs in real-time but does not include explicit language-grounding. While HOV-SG and ConceptGraphs both build open-vocabulary scene graphs, they are designed for referring to an object mainly using region references rather than fine-grained inter-object relations.

Instruction-Following Datasets

Multiple works have also explored language-guided navigation through instruction-following statements, often specifying a series of steps to move between regions in a large scene. Common datasets including Room Across Room [37] and Room-2-Room [6] focus on generating distinct steps to navigate between rooms but do not explicitly focus on decomposing the task into disambiguating objects explicitly through spatial relations, making it difficult to be robust to scene changes or imperfect language.

Referential Object Grounding

A number of papers have explored the task of learning referential object grounding, mainly on either the ReferIt3D benchmark or the ScanRefer task. These include BUTD-DETR [31], MVT [28], ViL3DRel [13], 3D-VisTA [84], and GPS trained on SceneVerse [33]. Despite massively upscaling data, GPS still only achieves an accuracy of 64.9% on Nr3D [33] and has low zero-shot generalization capabilities to new scenes and language. These models are also incapable of handling ambiguities in language input, and with the exception of GPS, cannot handle open-vocabulary input, making them unideal for real-world deployment.

Language Interaction in Embodied Agents

Some works [47, 60, 80], have explored the task of interactive visual grounding and ambiguity resolution, however, the formulation is either limited to simple input statements in 2D images, or the evaluation of ambiguous statements is limited to small amounts of human-annotated data on few scenes due to the cost and lack of such data. Other work in embodied, interactive agents has focused on multi-turn natural language dialogue. The TEACh benchmark [44] offers a human-generated dataset of task-driven dialogues for language grounding, dialogue understanding and task reasoning. [58] demonstrates the benefits of language feedback for improving real-world robotics tasks, although it is limited to one-way communication as the agent cannot pose questions to the human user. Extending this to navigation, [57] presents an instruction-following navigation system that uses large pretrained models, showing the effectiveness of large-scale data for language-guided navigation tasks. Building on these advancements, we aim to enhance interactive navigation in 3D scenes by improving the scale and quality of 3D language data with a focus on language scenarios that prompt further interaction for spatial reasoning.

2.4 Task Formulation

2.4.1 Referential Grounding with Imperfect References Task

We define the task of referential grounding with imperfect references as an augmented version of referential object-grounding which involves identifying objects without assuming a perfect match between references and scene objects. For a given statement, a referred object is only returned if it exists, otherwise the expected response is a) an explicit indication that the object was not found, and b) a suggested alternative object. We see this as an initial step in interactive referential grounding, which can facilitate navigation by clarifying uncertainties about the intended goal. Examples of imperfect grounding references are shown in Figure 2.2.

We differentiate this task from the original referential object-grounding task in ReferIt3D [3] and embodied tasks such as ObjectNav [9], ObjectGoal [5], and AreaGoal [5]. Compared to the standalone object referential grounding task in

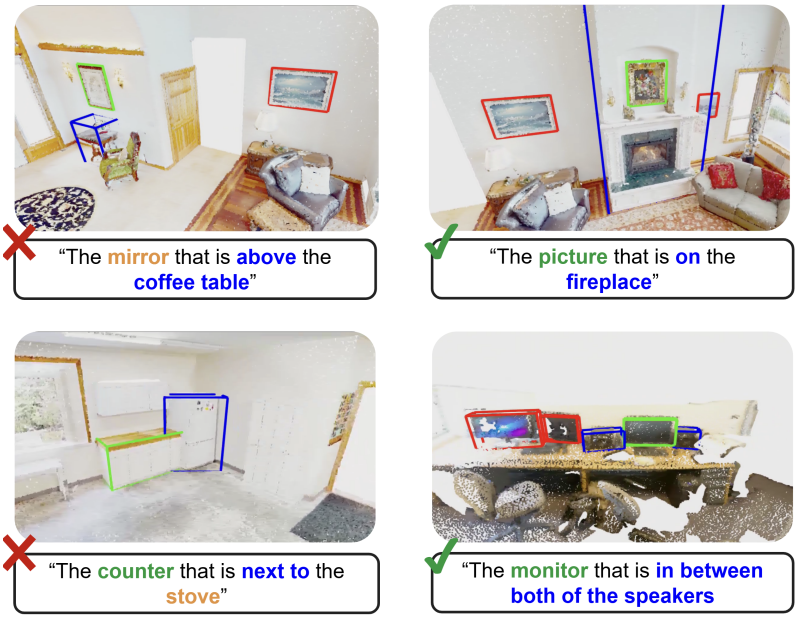


Figure 2.2: Examples of imperfect references (left) and correct references (right)

benchmarks such as ReferIt3D [3] and ScanRefer [13], our task aims to be a stepping stone towards multi-turn interactive navigation. Instead of assuming the referred object is always present and that only a single retrieval attempt is allowed, our task accommodates imperfect references and allows for multi-turn interactions.

In contrast to existing embodied navigation tasks like Object-Goal Navigation (ObjectNav) [9], which evaluate how an agent navigates to a goal, our task focuses on the nuances of 3D language grounding, independent of agent actions or planning. Conversely, the ObjectGoal and AreaGoal tasks are navigation-focused and the statements involve simple references like "find chair," while our task addresses the challenge of grounding complex spatial relations and detailed classes from more complex statements.

In general, we find that current formulations for related tasks are limited by their reliance on simple references, assumptions of reference correctness, and single-shot design. These constraints are unrealistic given the imperfect and dynamic nature of real-world scenes and instructions, highlighting the need for tasks focused on robust grounding in such scenarios.

2.4.2 Metrics

For the grounding and search subtask, we use binary classification metrics—true positive (TP), false positive (FP), true negative (TN), and false negative (FN)—to assess how well the model can identify object existence based on a referential statement. In particular, we note that false positives can be regarded as a more significant error in practical settings, where the method identifies a referenced object in the scene when in reality it does not exist in that specification, effectively "hallucinating" the object's existence.

To quantitatively assess the quality of retrieved object alternatives, we use a heuristic scoring system as a naive metric. We calculate a similarity score, $score_{sim}$, based on how well each suggestion matches aspects of the referential statement, such as object classes, attributes, and spatial relations. Aspects are weighted by importance, with object class and relation given higher weights as these are closer aligned to human intent compared to attributes. In general, these weights can be tuned through a grid search or learned through a neural network with human feedback. The score is then normalized by the maximum possible match score. For a given imperfect referential statement S with n total aspects $A(S) = \{a_1, ...a_n\}$ ordered by class then attributes and n varying between statements, a selected alternative S' with m total aspects $A(S') = \{a'_1, ...a'_m\}$, and λ_i as the weight on the i-th aspect of S:

$$score_{sim} = \frac{\Sigma_i^n \lambda_i * 1\{a_i \in A(S')\}}{\Sigma_i^n \lambda_i} \tag{2.1}$$

These heuristics provide a preliminary comparison metric for retrieved alternatives. However, such suggestions depend heavily on original user intent and preferences. Thus, human-labeled scores may better quantify quality, though this approach may be limited in scale.

2.5 Dataset Creation

2.5.1 Overview

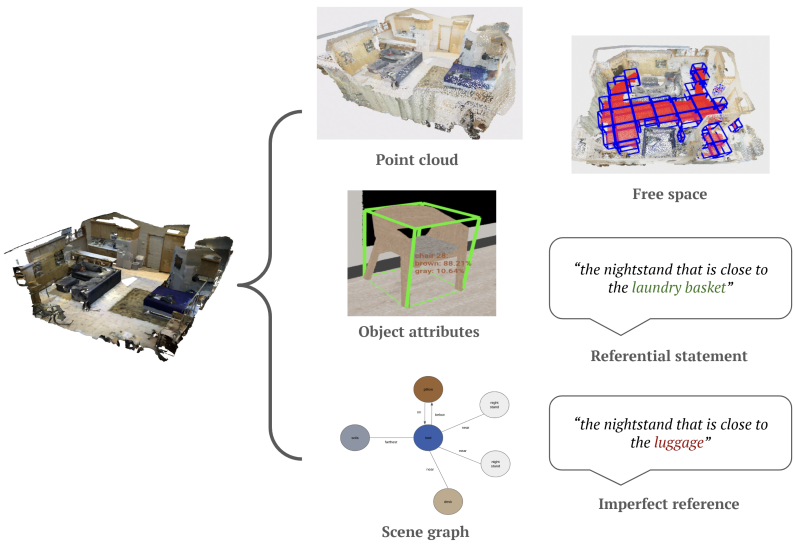


Figure 2.3: Overview of an individual data sample

To advance robust interactive navigation agents, we introduce IRef-VLA, a synthetically-generated public benchmark dataset. It combines 3D scans from five real-world datasets: ScanNet [18], Matterport3D [10], Habitat-Matterport 3D (HM3D) [49], 3RScan [34], and ARKitScenes [8], as well as Unity-generated scenes. Figure 2.4 shows the distribution of regions from each source. Each scene includes:

- Scene point cloud
- List of objects with semantic class labels, bounding box, and color(s)
- List of traversable free spaces
- List of regions with semantic labels and bounding boxes
- Scene graph of spatial relations split by room
- Language statements with ground-truth annotation

Key features of our dataset are large-scale scene graphs enabling identification of similar objects, traversable free space annotations, and imperfect language statements. In total, our dataset comprises 7,635 scenes with over 11.5K regions, 286K objects

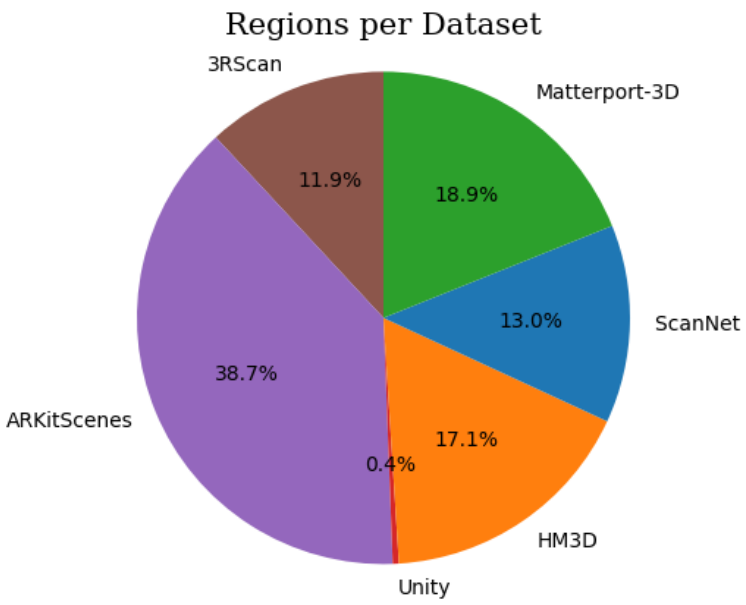


Figure 2.4: Breakdown of regions from each data source

across 477 classes, 7.6 million inter-object spatial relations, and 4.7 million referential statements. Figure 2.5 shows the spatial relations per dataset, with further details on the data curation process in Figure 2.6.

2.5.2 3D Scan Processing

To generate point cloud files, we used scene-level point clouds from PLY files for ARKitScenes, Matterport3D, and ScanNet. For HM3D, Unity, and 3RScan, point clouds were uniformly sampled from meshes, with colors derived from textures. Regions and objects were identified using semantic information from the original meshes. ARKitScenes, 3RScan, and ScanNet have single-room scenes, while Matterport3D and HM3D provide region segmentations, and Unity scenes are custom-segmented. Each object is labeled with an open-vocabulary class name, mapped to NYU40 [24] and NYUv2 [62] schemas with the provided mappings[2]. The dominant three colors were obtained for each object based on the point cloud and a color clustering algorithm.

To provide extra navigation targets, each scan was also processed to generate the horizontally traversable free space. Separate traversable regions in a room are

[2]For the Unity scenes, the ground-truth semantic labels were cleaned then manually mapped to the class schemas by five data annotators. A validation round was done to standardize the labels.

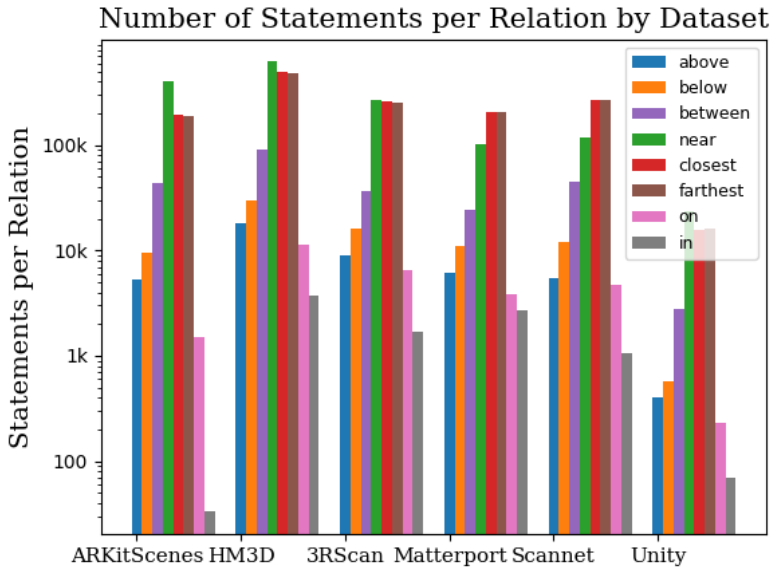


Figure 2.5: Number of statements per relation type from each dataset processed

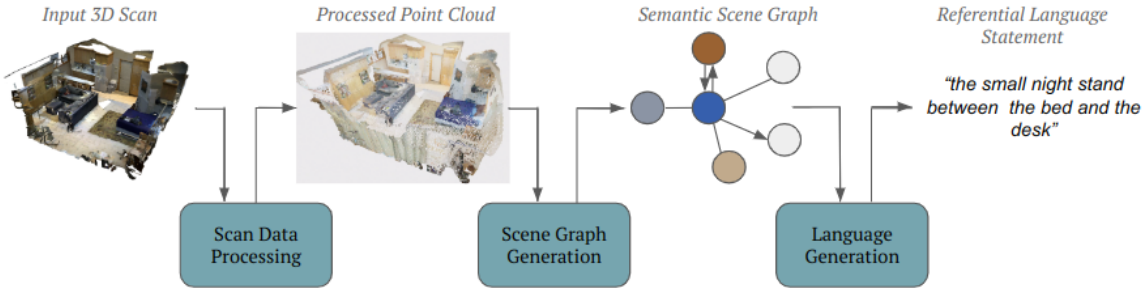


Figure 2.6: Data processing pipeline consisting of: 3D Scan Processing, Scene Graph Generation, and Language Generation

combined into sub-regions, for which spatial relations with other objects in the scene are generated to create unambiguous references to these spaces (e.g. "the space near the table").

2.5.3 Scene Graph Generation

Eight different types of semantic spatial relations were heuristically calculated based on the yawed object bounding boxes to generate a scene graph of relations. Relations

Table 2.1: Summary of semantic relationship types in IRef-VLA

Relation	**Definition**	**Synonyms**	**Properties**
Above	Target is above the anchor	Over	
Below	Target is below the anchor	Under, Beneath, Underneath	
Closest	Target is the closest object of a certain class to the anchor	Nearest	Inter-class
Farthest	Target is the farthest object from a certain class to the anchor	Most distant, Farthest away	Inter-class
Between	Target is between two anchors	In the middle of, In-between	Ternary
Near	Target is within a threshold distance of the anchor	Next to, Close to, Adjacent to, Beside	Symmetric
In	Target is inside the anchor	Inside, Within	
On	Target is above and in contact with the anchor in the Z-axis	On top of	Contact

are generated exhaustively for every pair or triple of objects within a region, then filtered based on the semantic classes involved. Table 2.1 defines the types of spatial relations used.

2.5.4 Language Generation

Referential statements were synthetically generated based on the computed scene graph using a template-based generation method. From the table, synonyms for each relation are used to add variety into the statements. Every statement has at least one semantic relation and only uses object attributes if needed to distinguish the target object. The generated statements are also:

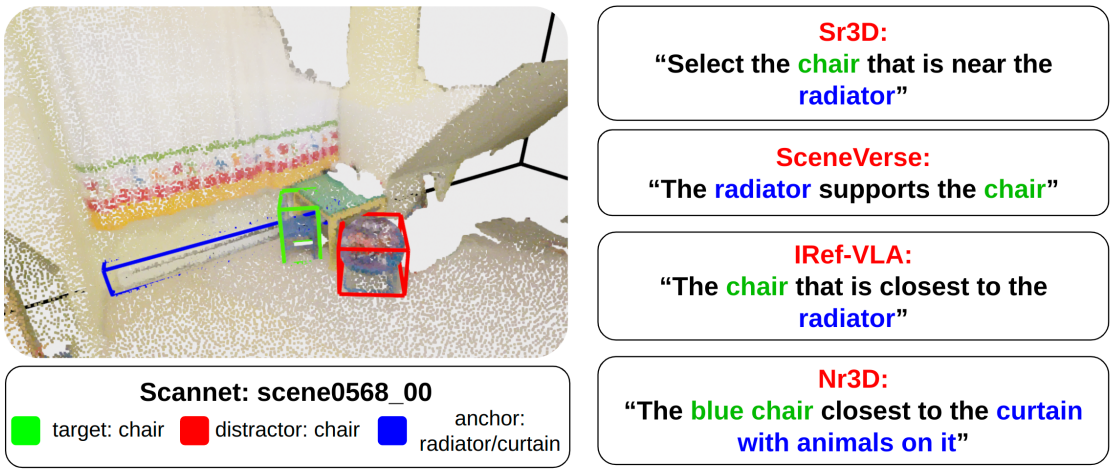


Figure 2.7: A comparison between heuristically generated statements describing a binary spatial relation from Sr3D, Nr3D [3], SceneVerse [33], and IRef-VLA. Both chairs are close to the radiator, so using the superlative relation "closest" is the clearest way to disambiguate.

1. **View-independent**: The relation predicate for the target object does not depend on the perspective from which the scene is viewed from.
2. **Unambiguous**: Only one possibility exists in the region for the referred target object.
3. **Minimal**: Following Grice's maxim of manner [22], statements use the least possible descriptors to most clearly disambiguate the target object (Figure 2.7).

Additionally, the dataset includes "imperfect" referential statements describing non-existent objects. These false statements serve to enhance robustness to noisy language and improve evaluation, as identifying non-existent objects is a key skill for language grounding. The statements are generated by altering one target or anchor object attribute in existing statements to similar values, ensuring they are contextually similar to true statements.

2.6 Baseline Evaluation

We evaluate our benchmark on both the original referential grounding task and our extended task. First, we compare our data with ReferIt3D [3] using two SOTA

Table 2.2: Dataset generalizability on various baseline models

Method	Train Checkpoint	Test Set			
		Sr3D	Nr3D	IRef-VLA	
		Overall	Overall	ScanNet	Full
MVT [28]	Baseline (Sr3D, reported)	64.5%	-	-	-
	Baseline (Sr3D, reproduced)	59%	31.8%	29.0%	17.2%
	IRef-VLA-ScanNet	50.0%	29.7%	56.0%	26.7%
	IRef-VLA-Full	41.0%	25.9%	44.0%	47.0%
3D-VisTA [84]	Baseline (Sr3D, reported)	76.4%	-	-	-
	Baseline (Sr3D, reproduced)	75.7%	46%	39.2%	24.8%
	SceneVerse *(0-shot text)* [33]	-	43.1%	-	-
	IRef-VLA-ScanNet	62.4%	41.8%	63.7%	32.3%
	IRef-VLA-Full	65.8%	44.9%	70.8%	60.6%

supervised methods for object referential grounding. Then, we implement a graph-search baseline for the task of grounding with imperfect references.

2.6.1 Referential Object Grounding

Experimental Setup

To evaluate the effects of scaling up the amount of referential language and number of real-world scenes on the referential grounding task, we train two open-source supervised referential grounding baseline models on our data: MVT [28] and 3D-VisTA [84]. For the training splits, we use the official ScanNet/ReferIt3D train and validation splits for our ScanNet data, and follow an 80% train, 20% validation split for the remaining scenes. To demonstrate generalizability, we test the *zero-shot transfer* capabilities of these models trained on IRef-VLA by training the models first on the ScanNet scenes alone, and then on the full dataset, and evaluating directly on the Sr3D and Nr3D [3] test sets, which consist of synthetically generated and human-uttered referential statements respectively. Our zero-shot transfer results

along with a comparison to the baseline model performance are shown in Table 2.2. Both models are trained until training loss convergence.

Generalizability Results

We observe the following:

(i) Even without seeing any Nr3D statements and without direct training on any Sr3D statements, we observe relatively high accuracies on both test sets when training MVT (50% on Sr3D, 29.7% on Nr3D) and 3D-VisTA (62.4% on Sr3D, 41.8% on Nr3D) with our ScanNet statements. On the Nr3D test set, we note that the baselines trained on Sr3D perform higher due to similar view-dependent statement distribution, achieving 31.8% and 46% accuracy for MVT and 3D-VisTA respectively. However, the small performance differences of 2.1% with MVT and 4.2% with 3D-VisTA using the baseline trained on our data shows that our pipeline for synthetically upscaling only the number of referential statements and using new relations without increasing the number of scenes still improves the zero-shot capabilities of object referential models.

(ii) We observe that increasing the number of training scenes from our dataset further improves the grounding performance of 3D-VisTA on the IRef-VLA ScanNet split from 63.7% to 70.8%, on Sr3D from 62.4% to 65.8%, and Nr3D from 41.8% to 44.9% while MVT instead underfits likely due to being a smaller model. Upscaling the number of our training scenes further improves performance on zero-shot transfer to Nr3D, narrowing the gap for 3D-VisTA between this checkpoint and the Sr3D checkpoint to 1.1%, despite IRef-VLA containing only view-independent relations. 3D-VisTA trained on our full data also performs 1.8% better than 3D-VisTA trained on the SceneVerse zero-shot text split consisting only of their synthetic statements [33], verifying that better heuristics for generating natural-sounding referential statements improve the effectiveness of upscaling the number of scenes.

(iii) Both pre-trained baselines perform poorly generalizing to our IRef-VLA validation sets at 29%/17.2% accuracy on IRef-VLA-ScanNet/Full with MVT and 39.2%/24.8% with 3D-VisTA, highlighting the difficulty of our benchmark. While there is a significant domain shift with the pre-trained baseline models

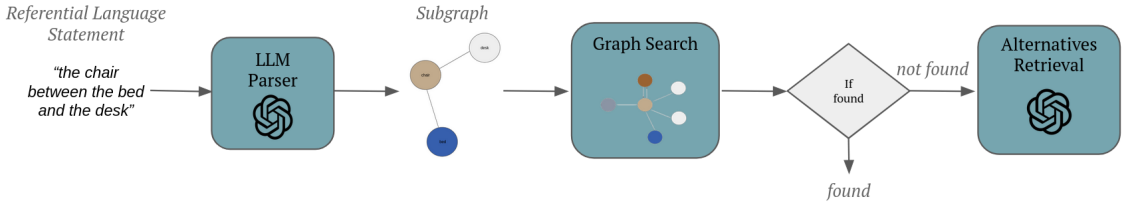


Figure 2.8: Pipeline for graph-search and alternative generation baseline

on our data, those trained on IRef-VLA show a smaller gap when evaluated on Sr3D. This suggests that our dataset's diverse language and scene distribution improves generalization, especially in structured language.

2.6.2 Referential Grounding with Imperfect References

Experimental Setup

To establish a quantitative baseline for grounding with imperfect references, we assess methods on two subtasks: 1) identifying existence of objects and 2) suggest alternatives when necessary. We augment SOTA methods for the former and evaluate our graph-search baseline for the latter. Methods are evaluated on a split of our dataset corresponding to the ReferIt3D [3] test split. The results can be found in table 2.3 and implementation details are further described below.

Table 2.3: Classification results for grounding object existence

Baseline Model	True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)	F1-Score
MVT + Binary Classifier	66.2%	97.1%	2.9%	33.8%	78.3%
Graph-Search	90.4%	98.9%	1.1%	9.6%	94.4%

Augmented SOTA Models: As existing SOTA referential grounding baselines cannot directly determine whether a referred object is in the scene, an additional binary classification head was added to the MVT model as a point of comparison. The concatenated object features are passed through a simple two-layer MLP and

trained with a cross-entropy loss. The additional referential losses are only added if the object truly exists in the scene, ensuring that the object grounding learning is not affected. We use pre-trained checkpoints and finetune with the binary classification loss.

Graph Search Baseline: To benchmark robustness in grounding with imperfect object references and demonstrate a simple method for alternative generation, we implement a graph-search method using heuristically-generated scene graphs. We first use an LLM, gpt-4o-mini[3], to parse each statement into a scene subgraph representation with few-shot prompting using five training samples. A given statement is parsed into: target object, anchor objects, attributes, and relation in JSON format, then converted into a subgraph representation where nodes consist of object properties and edges represent relations between objects. We then implement a search method that searches the scene graph for the referenced subgraph. As we are searching for subgraphs and not a single target node, we use breadth-first search to find candidate target nodes, then smaller depth-first searches to find the remaining subgraph. If the exact subgraph is not found, we extract existing referential statements corresponding to partial subgraph matches and prompt the LLM to choose the statement closest to the input statement in a multiple-choice question-answer (MCQA) style prompt. The full pipeline is shown in Figure 2.8.

Results

Table 2.4: Accuracy of parsing and alternatives modules in graph-search baseline

Baseline Model	LLM Parsing Accuracy	Average Alternative Similarity
Graph-Search	94.0%	61%

We first quantify the LLM parsing accuracy as it directly upper bounds the downstream grounding and alternative scoring. For each statement, we compare the LLM-generated structural output to the ground-truth sub-scene graphs. We achieve a parsing accuracy of 94% as seen in Table 2.4. The results of classifying object

[3]https://platform.openai.com/docs/models/gpt-4o-mini

existence are in Table 2.3. We note that the graph search baseline is able to find the correct object that exists 90% of the time (TP rate) using the heuristically-generated scene graphs as the knowledge base, indicating an upper-bound for robust grounding when ground-truth calculated relations are used. In particular, the true negative rate is 97% for MVT and 98.9% for the graph-search method, indicating that referential grounding methods can be augmented to explicitly determine when a referred object does not exist as described. When deploying referential grounding methods in the real-world, this would enable robustness of results to changing scenes and mistakes by humans. From Table 2.4, scoring LLM-selected alternatives with a simple heuristic results in a score of 61%, indicating that matching object descriptions alone without direct visual information can set a baseline for alternative selection where over half the aspects match. This can be used as a lower performance bound for comparison to other alternative selection methods developed.

2.7 Limitations

As is, our dataset uses synthetically generated language, which, while scalable, lacks view-dependent and allocentric statements common in natural communication. Expanding our dataset to include such statements with possible LLM augmentation or human labeling will enhance the dataset diversity and provide more complex spatial relations. Additionally, the heuristics-based scoring metric for alternatives does not fully capture human preferences or the subtle nuances of alternative suggestions, potentially leading to mismatches with genuine human evaluation criteria. Incorporating human-generated alternatives or having human-scoring of the alternative retrieval method will better capture the subtleties of human intent. Another future direction of work is to explore a multi-turn dialogue setting for specifying navigation goals instead of the single step currently modeled.

2.8 Application to CMU VLA Challenge

The release of the IRef-VLA dataset has also been used to support the CMU Vision-Language-Autonomy (VLA) Challenge[4] as a training and question-generation resource. The challenge provides a real-robot system equipped with a 3D LiDAR and a 360-degree camera and base autonomy onboard that can estimate the sensor pose, analyze the terrain, avoid collisions, and navigate to waypoints. Teams are expected to come up with a vision-language model that can take a natural language navigation query or question about the scene and respond accordingly.

The natural language questions are separated into three categories: numerical, object reference, and instruction following. All queries contain an object-centric spatial reasoning component that requires semantic spatial understanding of the objects in the scene, thus, the diverse grounding statements in our dataset provides a starting point for model training of fine-tuning. The challenge questions incorporates the object-centric grounding task in more practical scenarios and in unseen environments, both in simulation and the real world, where vision and language may be more challenging to align. We aim that this will help verify the transferability of IRef-VLA data to be used in more practical scenarios and aid the generalizability of methods proposed.

2.9 Conclusion

Aiming to advance robust scene understanding for interactive robotic navigation, we introduce IRef-VLA, a novel benchmark dataset for referential grounding with imperfect references. Our benchmark provides a large-scale resource for grounding in 3D scenes while incorporating unique features such as structured scene graphs and imperfect statements to form the novel task of referential grounding with imperfect language. We validate the dataset's diversity and difficulty through baseline experiments with SOTA models, provide a baseline implementation using scene graphs for grounding and alternative generation, and propose metrics to evaluate performance. With this new benchmark and task, we hope to enable the development of generaliz-

[4]https://www.ai-meets-autonomy.com/cmu-vla-challenge

able robotics agents robust to imperfections and ambiguities in the real-world when interacting with humans using natural language.

2.10 Future Work: Benchmarking Dialogue in VLN

Since the release of IRef-VLA, a number of other 3D grounding datasets have been released, such as ViGiL3D [72], SURPRISE3D [27], and PlaceIt3D [1]. Such datasets continue to help expand and diversify the space of 3D vision-language grounding data, improving on both task, language, and scene diversity. SURPRISE3D focuses on language-grounded semantic segmentation, while PlaceIt3D uses spatial object references to guide object placement. The release of open-source 3D scan datasets such as Scannet++ [77] also allows for the creation of language grounding queries on a set of new high-fidelity scenes. However, there continues to be a gap for language-interactive or dialogue-based 3D grounding benchmarks and thus, uncertainty in the field on how to develop or evaluate such methods. We discuss this further in 4 and leave this for future work.

2.11 Acknowledgements

This chapter was jointly authored with Nader Zantout, and contributed to by Pujith Kachana, Zongyuan Wu, Wenshan Wang, and Ji Zhang.

Chapter 3

Online 3D Semantic Reasoning and Navigation

3.1 Abstract

Interpreting object-referential language and grounding objects in 3D with spatial relations and attributes is essential for robots operating alongside humans. However, this task is often challenging due to the diversity of scenes, large number of fine-grained objects, and complex free-form nature of language references. Furthermore, in the 3D domain, obtaining large amounts of natural language training data is difficult. Thus, it is important for methods to learn from little data and zero-shot generalize to new environments. To address these challenges, we propose SORT3D, an approach that utilizes rich object attributes from 2D data and merges a heuristics-based spatial reasoning toolbox with the ability of large language models (LLMs) to perform sequential reasoning. Importantly, our method does not require text-to-3D data for training and can be applied zero-shot to unseen environments. We show that SORT3D achieves state-of-the-art zero-shot performance on complex view-dependent grounding tasks on two benchmarks. We also implement the pipeline to run real-time on two autonomous vehicles and demonstrate that our approach can be used for object-goal navigation on previously unseen real-world environments. All source code

for the system pipeline is publicly released. [1]

3.2 Introduction

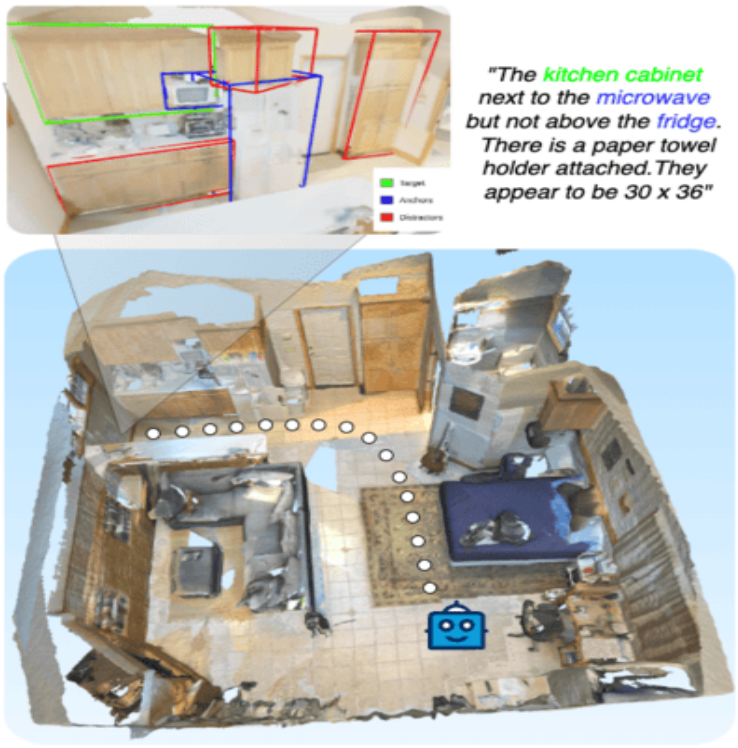


Figure 3.1: An example of our system's workflow for using referential object grounding for downstream object-goal navigation. The agent uses the 2D image for fine-grained grounding in the presence of distractor objects.

As we progress toward generalizable robots operating in human-centered environments like homes and offices, it is crucial for these agents to interact through natural language and align visual observations with natural language references. This capability is essential for applications such as robot caregivers and indoor assistants. Resolving natural language expressions referring to specific objects using semantic object attributes and inter-object spatial relations—the core challenge of **3D referential grounding**—remains difficult despite being an intuitive task for humans. For example, understanding statements such as "*the chair closest to the closet door*",

[1] https://github.com/nzantout/SORT3D

is a task trivial for humans [3] but still challenging for robots. While humans are usually able to identify objects from referring expressions by filtering out irrelevant objects, reasoning about spatial relationships, and utilizing semantic object attributes, such tasks remain challenging for state-of-the-art (SOTA) methods due to several reasons. First, indoor environments often contain numerous objects belonging to fine-grained classes [49], with distributions that vary widely across different homes and environments. Second, training end-to-end learning-based methods on 3D referential grounding requires a large amount of annotated data aligning language references to a 3D scene, which the 3D domain lacks in comparison to the 2D vision-language domain [3].

While a number of existing works have developed end-to-end methods to tackle this task through training multi-modal alignment with large transformer models [28, 31, 84], these methods require large-scale annotated data, often overfitting to specific syntactic structure in training datasets while struggling to generalize to more complex utterances, resulting in mediocre performance. More recently, numerous works have leveraged the reasoning capabilities and rich language semantics of large language models (LLMs) for 3D referential grounding [15, 20, 26, 38, 75, 76, 78]. While some achieve strong results on benchmark datasets [20], complex real-world natural language expressions that employ both semantic attributes and spatial relations such as "the tall recycling bin to the left if you are facing the door" remain challenging. Many LLM-based approaches either struggle with poor zero-shot performance or rely on careful fine-tuning or heavy prompt engineering tailored towards benchmark datasets. Consequently, such methods are not typically designed for system deployment, relying on high fidelity reconstructed meshes of 3D scenes [10, 18] and failing to account for constraints in model size, efficiency, and noise in the real-time semantic mapping process.

To this end, we propose **SORT3D**, a **S**patial **O**bject-centric **R**easoning **T**oolbox for **3D** Grounding Using LLMs, shown in Figure 3.2, as a novel pipeline for 3D spatial reasoning tailored towards the downstream application of object-goal navigation [9]. SORT3D is modeled intuitively after human reasoning in object disambiguation, enabling data-efficient 3D referential grounding without requiring annotated 3D training data. To achieve this, we decompose the task into a three-stage approach. After obtaining object names and bounding boxes using an instance-level semantic

mapping module, we leverage SOTA 2D vision-language models (VLMs) to extract semantic object attributes important for distinguishing objects. We then use an open-vocabulary object filtering module to efficiently filter relevant objects in large scenes with hundreds of objects, similar to how humans focus their attention to relevant objects. We finally leverage the strong semantic language and sequential reasoning priors of LLMs using chain-of-thought prompting [73] to parse a complex referential statement into a series of function calls to our **spatial reasoning toolbox**, containing search functions that return the IDs of objects that satisfy heuristically defined elementary spatial relations. This delegates complex relative spatial reasoning to structured rule-based logic, ensuring greater accuracy and interpretability. As a result, our method only requires a single in-context example of the toolbox usage and no other training data.

We evaluate our method on standard 3D object referential grounding benchmarks, ReferIt3D [3] and IRef-VLA [81], and demonstrate performance competitive with SOTA on complex view-dependent statements while requiring less data to zero-shot generalize. We also deploy our full pipeline on two robotic ground vehicles for real-time indoor navigation, demonstrating our method's ability to further generalize to previously unseen environments and run online alongside real-time perception and autonomy stacks.

3.3 Related Work

Referential Object Grounding Datasets

Using referring expressions to identify a target object in a 3D scene, defined as the 3D referential grounding task, has been explored in a number of datasets such as ReferIt3D [3], ScanRefer [13], SceneVerse [33], and IRef-VLA [81]. Of these datasets, only the Nr3D subset of ReferIt3D and ScanRefer contain human-generated natural language utterances, while the remaining datasets generate synthetic statements using template-based generation and/or LLMs to rephrase. Synthetic datasets only employ basic spatial relations like "lamp by the desk" in referential statements, although IRef-VLA adds the usage basic semantic object attributes—color and size—to create unambiguous references. Compared to the template-generated datasets, the human-

generated statements in Nr3D contain far more complex spatial grounding language, such as "Facing the three boxes, it is the box on the right that is on top of the larger blue box.", which requires view-point grounding, "The lamp to the left of the desk. NOT the lamp between the beds", which contains negation, and "the chair closest to the metal appliance" which uses coarse object references and semantic attribute. This complexity, coupled with the small scale of human-generated 3D referential language data, motivates the need for 3D referential grounding methods to leverage strong semantic language priors from other sources (like LLMs), and use a more structured approach for grounding.

3D Referential Object Grounding Baselines

Grounding object references to 3D scenes has been explored more extensively by various methods since the introduction of benchmarks specific to the task. Approaches to this task include end-to-end models like BUTD-DETR [31], MVT [28], ViL3DRel [14], 3D-VisTA [84], and GPS trained on SceneVerse [33], which fuse multi-modal information in large transformer models and are trained and fine-tuned directly on referential grounding benchmarks. More recent methods decompose the task, leveraging neuro-symbolic frameworks like NS3D [26], and LLMs and VLMs in a zero-shot manner to effectively reason about spatial relations like ZSVG3D [79], VLM-Grounder [75], CSVG [78], and Transcrib3D [20]. Utilizing the reasoning capabilities of LLMs in the text domain, Transcrib3D achieves overall accuracies of 70.2% and 98.4% respectively on subsets of Nr3D and Sr3D, outperforming all previous methods. To achieve this, Transcrib3D represents a scene as a list of object names, colors, and positions, and relies on iterative code generation and benchmark-specific prompt engineering giving the LLM guiding principles for grounding. Transcrib3D additionally fine-tunes smaller LLMs on incorrect answers with corrections self-reasoned by larger LLMs. While significant progress in grounding accuracy has been made, the complex spatial reasoning required for the Nr3D dataset, especially with statements that involve egocentric viewpoints and utilize object semantics, continues to pose a challenge, especially for zero-shot methods [20, 75, 78, 79]. We aim to address this gap with our method by combining structured heuristics and vision foundation models with the sequential reasoning capabilities of LLMs.

Grounding Large Language Models in 3D

Recent efforts have also aimed to develop 3D foundation models capable of handling general 3D tasks. These models leverage the strong reasoning capabilities of pretrained LLMs, achieving strong performance on 3D scene understanding tasks by fine-tuning the LLMs on tokenized 3D data. One of the pioneering works in this space, 3D-LLM [25], distills rich 2D foundation model features and a 2D VLM backbone to improve performance, mitigating the lack of 3D data. [15] improves upon the work to handle grounding to different viewpoints and generalizes to various 3D tasks with large-scale language-scene pretraining. The efficacy of 2D foundation model features in these approaches demonstrates that grounding 3D understanding in well-established 2D and language-based representations is a powerful approach for advancing 3D reasoning. This factor subsequently inspires the design of our pipeline with the incorporation of 2D captions.

Vision-and-Language Navigation with LLMs

A number of recent works have also leveraged the sequential reasoning capabilities of LLMs for navigation tasks. NavGPT [83] and NavCoT [39] use VLMs to generate text descriptions of viewpoints in the scene in 2D, then task the LLM with selecting the next action in an instruction-following task. Such methods demonstrate the effectiveness of leveraging 2D visual information to guide grounding and action selection in the 3D space. However, within a collaborative setting between a human and a robotic agent, the human would more commonly use single referential statements like "fetch the tv remote on the cabinet" and assume the robot has full knowledge the scene rather than describing a full trajectory towards an object. For building a 3D referential grounding-based VLN system with object-goal navigation as the basic downstream task, text descriptions of landmarks must also be combined with a structured map representing the scene.

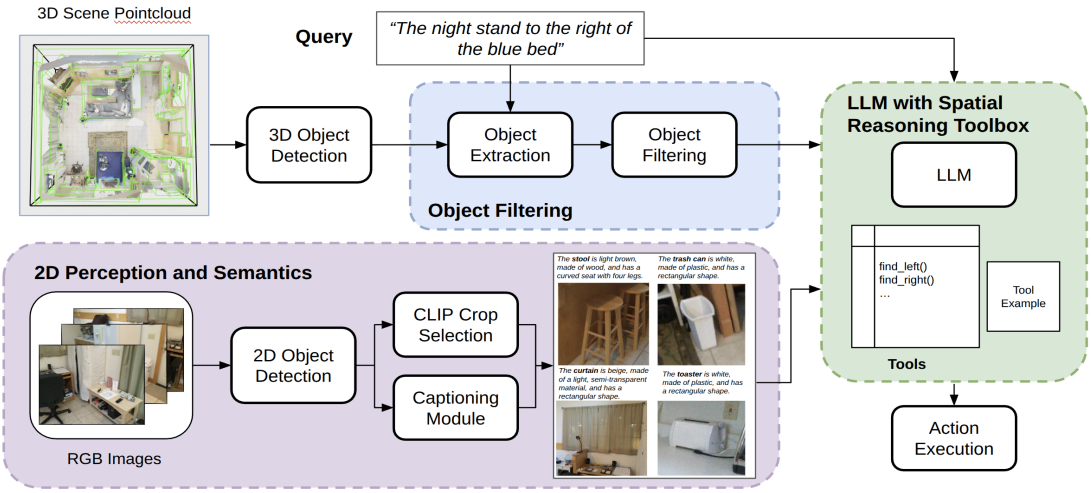


Figure 3.2: The full system diagram for the SORT3D framework

3.4 Methodology

In a human-agent collaborative setting, humans commonly refer to objects in the scene by relating them to other objects using commands like "grab the red mug in the top left cupboard over the sink". Finding the correct object being referenced from the utterance is the task of **3D referential grounding**, which is fundamental for the deployment of a practical VLN system. 3D referential grounding additionally acts as a precursor to downstream tasks such as object-goal navigation, multi-action instruction-following, and scene visual question answering (VQA). We therefore present **SORT3D**, a zero-shot pipeline for 3D referential grounding, which decomposes the task into multiple subtasks, leverages foundation models to obtain robust zero-shot performance, and targets downstream mobile robot navigation in a real-world environment.

The input to the grounding pipeline consists of perception information from the scene and a free-form referring expression in natural language. The output is the ID of the target object referenced. Figure 3.2 shows our proposed framework, which can be broken down into four components: (i) an instance-level semantic mapping system to obtain 3D bounding boxes for real-world deployment, (ii) a captioning pipeline to incorporate rich 2D semantic information for each object, (iii) filtering for relevant objects based on the input utterance, and (iv) LLM-based reasoning augmented with

a spatial reasoning toolbox to resolve the target object, followed by code generation for executing a downstream action. Each of these components is described in further detail below.

3.4.1 Instance-level Semantic Mapping

For our real-world experiments, we use an object instance-level semantic mapping module running in real-time to obtain the 3D bounding boxes to be input into the LLM and the spatial reasoning toolbox. This component is the only pipeine change required for our method to be deployed in the real-world. Our mobile robot perception setup for real-world experiments consists of a 360 camera and a 3D LiDAR (section 3.6.3 contains further hardware details). We initially perform object detection in 2D using open-vocabulary object detection [41] and instance segmentation [52] models. We then project the registered LiDAR point clouds onto the semantic images and associate each point with its corresponding pixel semantic ID. As the robot moves and produces new observations, we associate per-frame object instance pointclouds using a 2D tracking module and 3D proximity priors, followed by filtering steps to obtain 3D instance pointclouds[2]. The usage of open-vocabulary 2D foundation models allows our semantic mapping module to generalize to new environments as we show in our real-world experiments (section 3.6.3). For our results on the ReferIt3D benchmarks, we simply use ground truth bounding boxes and instance segmentations.

3.4.2 Enhancing Object Perception with 2D Captions

Accurately understanding the attributes and affordances of 3D objects is an essential first step for referential grounding. Existing works [20], [28] use high-level information such as object bounding boxes and labels for this task, often acquired through 3D segmentation. While 3D segmentation provides useful object-centric information, it often fails to capture fine-grained attributes like color and shape, which is a key bottleneck of 3D object grounding models [84]. Nonetheless, large-scale training for accurate 3D perception is still a challenging problem due to the lack of data and task complexity. On the other hand, VLMs trained on vast amounts of 2D data

[2]For further details on the semantic mapping module used, see the code released at: https://github.com/gfchen01/semantic_mapping_with_360_camera_and_3d_lidar

have strong priors for object understanding in 2D. Notably, VQA models excel at captioning objects in a scene. Therefore, we leverage 2D VQA models to generate descriptions for 3D objects in the scene, providing richer visual details for grounding that are otherwise missed with pure 3D perception. This approach also mirrors how humans perceive and identify objects: narrowing down candidate objects through attributes and relations to resolve ambiguities. In our approach, this finer-grained inspection is achieved by generating captions for each object from cropped object images, providing a more intuitive and precise form of object grounding.

A key decision to be made for captioning is what image to give the VQA model when multiple views of an object are present. We make this choice by using the viewpoint that has the *highest CLIP similarity* with the target label, which is obtained from ground truth for the ReferIt3D benchmarks and from the semantic mapping module for the real-world experiments. We use Qwen2-VL-7B [67] as our VLM as we found it to perform best in generating accurate and concise descriptions following our template, and the quantized version of Qwen2.5-VL-Instruct-3B for system deployment due to memory constraints. We query the VLM with the following prompt format:

‘‘You are an AI model that describes the characteristics of an object in an image. Describe the <object> in this image, using properties like color, material, shape, affordances, and other meaningful attributes. Provide the response in this format: ‘The <object> is <color>, <material>, <shape>’".

We release all object crops and captions as a supplement to the ScanNet [18] dataset along with our code. A sample of object crops and their corresponding captions are shown in Figure 3.3.

3.4.3 Filtering for Relevant Objects

Indoor environments such as homes can consist of hundreds of objects, of which only a few are relevant to a given language query or task. Thus, inspired by the ability for humans to filter out irrelevant objects and the success of past works [20, 26], we implement an LLM-based filtering module consisting of two queries. Given an input command like “The nightstand to the right of the bed”, the first query extracts object nouns and modifiers (e.g. nightstand and bed), and the second query returns the

Figure 3.3: Generated image crops and corresponding caption

relevant object instances from the list of IDs and names provided by the perception module (in this case returning all nightstands and beds in the scene). We use Mistral Large 2 [66] for these steps, filtering objects based on their text descriptions, to best leverage the ability of LLMs to process textual information.

3.4.4 Spatial Reasoning Toolbox

With only the relevant objects extracted, the subset of objects and their captions are then fed to an LLM reasoning agent as a list of n_o objects, where each object o_i is represented by the list of attributes: $\{\texttt{id}, \texttt{name}, \texttt{caption}, c_x, c_y, c_z, \texttt{size}\}$, where `id` is a unique integer identifier for the object, (c_x, c_y, c_z) are discretized coordinates of the object center, and `size` is the area of the largest face.

However, directly prompting off-the-shelf LLMs for 3D referential grounding with this scene representation results in poor performance due to their limitations in spatial and mathematical reasoning: for example, when querying "the nightstand to the left of the bed", the LLM picks the one with the smallest x value. As a result,

we abstract spatial reasoning away from the LLM by creating a **spatial reasoning toolbox** consisting of heuristic search functions that find objects referred to by a set of elementary spatial relations. The key idea is that when referring to an object using relative spatial relations, only a fixed set of relations are needed. Thus, defining this set of heuristic functions is sufficient for handling diverse inter-object referential scenarios. The search functions and their arguments are listed in Table 3.1.

Spatial Search Functions
`find_near(target, anchor)`*
`find_between(target, anchor1, anchor2)`*
`find_above(target, anchor)`*
`find_below(target, anchor)`*
`order_bottom_to_top(targets)`*
`order_smallest_to_largest(targets)`*
`find_objects_near_room_corner(targets)`*
`find_left(target, anchor)`†
`find_right(target, anchor)`†
`order_left_to_right(target, anchor)`†

Table 3.1: Heuristic search functions in the spatial toolbox. View-independent functions are marked with an asterisk (*), and view-dependent functions are marked with a dagger (†).

The LLM is prompted with an in-context example to decompose a referential statement into a series of search calls and choose a single object ID from the returned lists of IDs. For example, given the query "Find the computer near the desk with a printer on it", the LLM first calls `find_below(desk, printer)` returning `[2]`, then `find_near(computer, 2)` returning `[3, 4]`, finally picking the computer with ID 3. We note a few implementation details regarding the search functions:

(i) For statements employing view-dependent relationships, like left and right, we tackle the case where no specific observer viewpoint is given (common in Nr3D), and assume the relationship is unambiguous from any feasible viewing direction. We therefore determinine left/right direction from the signed angle formed by the nearest point in free space and the target and anchor's centroids.

(ii) We use the area of the largest face in `order_smallest_to_largest`, which is more intuitive than volume for flat objects.

(iii) All search functions have access to the full output of the perception module including object names and 3D bounding boxes, which is partially hidden from the LLM. This allows us to develop more complex logic for search functions without affecting LLM reasoning, which is an advantage of our approach.

Our accompanying repository contains further implementation details on the spatial search functions.

3.4.5 Parsing for Action Execution

For our benchmark results, the LLM is prompted to only output a single object ID denoting the chosen object. For our real-world deployed pipeline, the LLM may output a series of either `go_near` or `go_between` function calls which take object IDs as arguments and generate waypoints in the scene which are sent to an obstacle-avoidant downstream planner for sequential navigation. `go_near(target)` places a waypoint at the closest point in traversable space to the target's center, while `go_between(target1, target2)` places one near the midpoint between the two targets.

3.5 Experimental Setup

We quantitatively evaluate our method on two 3D object-referential datasets, ReferIt3D [3] and IRef-VLA [81]. Both datasets consist of utterances describing a target object in a ScanNet [18] scene using spatial relations. In particular, ReferIt3D is split into Sr3D, which consists of synthetically generated utterances from five relation categories while Nr3D consists of natural language statements collected from humans, with unconstrained methods of describing target objects. Statements are categorized as "Easy"/"Hard" based on the number of "distractor" objects of the same class as the target object in the scene and also "View-Dependent"/"View-Independent". IRef-VLA consists entirely of template-generated statements that are view-independent but contains utterances for a diverse set of 3D scans and enforces every statement to contain a spatial relation. The set of spatial relations is expanded to include eight total relations, including ternary relations (e.g. "between") and numerical relations (e.g. "second closest"). Utterances may also contain attributes such as color and

size if needed to disambiguate the target object from distractors. Additionally, we deploy SORT3D on two ground vehicles, and validate the system's generalizability by testing navigation commands containing references to spatial relations, references to object attributes, and implicit or indirect requests in three previously unseen indoor environments.

3.5.1 Referential Grounding on Benchmark Datasets

We test our model on both ReferIt3D subsets and the subset of IRef-VLA using ScanNet scenes and compare to SOTA baselines. On each data subset, we evaluate our model on 200 sampled statements[3], sampled to match the distribution of Easy, Hard, View-Dependent, and View-Independent statements in the original ReferIt3D test dataset. We focus our comparison against Transcrib3D [20] which, to the best of our knowledge, is the best performing model on ReferIt3D to date. For a fair comparison, we run Transcrib3D with the same two LLMs we use on the same test splits[4]. For our methods, we conduct multiple trials on each data split to measure variance in LLMs, reported with standard deviation values on the grounding accuracy, which we note that other LLM-based methods do not report.

3.6 Results and Discussion

3.6.1 Referential Grounding on Benchmark Datasets

The grounding accuracy on ReferIt3D is shown in Tables 3.2 and 3.3, and accuracy on IRef-VLA is shown in Table 3.4. We see that our method achieves higher accuracy with GPT-4o as the LLM backend and is on par with SOTA methods on View-Dependent statements in Nr3D and Hard statements in IRef-VLA while requiring no data to train. We also note that the use of LLMs introduces variance between trials, affecting grounding accuracy up to 6%.

[3]We use a subset of the test set due to the cost of LLM evaluation on the full test set and the need for multiple runs to obtain variance statistics

[4]The GPT4 model used in their work is now a legacy model. We run their method with GPT4o instead.

While supervised baselines such as ViL3DRel [14], 3D-VisTA [84], and SceneVerse [33] report slightly higher overall grounding accuracies, these methods are explicitly trained on ReferIt3D data. Similarly, while Transcrib3D reports higher accuracies, it relies on guiding principles [20] that are tailored to the language used in Nr3D and Sr3D, which improve performance on those benchmarks.

In contrast, our approach is purely zero-shot, requiring only one single example of how to use the spatial reasoning toolbox, which does not have to be from a particular dataset, and we employed no dataset-specific training or fine-tuning. Despite this, by leveraging foundation models to obtain object semantic attributes and mapping spatial reasoning into sequential reasoning, our spatial reasoning toolbox approach achieves overall performance comparable to SOTA supervised and fine-tuned methods on Nr3D, and performance on view-dependent statements on par with Transcrib3D. On Sr3D, SORT3D surpasses SOTA supervised training methods and achieves close overall performance to Transcrib3D while surpassing it in view-dependent accuracy. This demonstrates the effectiveness of our approach at handling spatial reasoning where viewpoint anchoring is required. We see that SORT3D is able to explainably resolve complex view-dependent relations with multiple anchors and complex semantic descriptions (Figure 3.4-a), while also providing explainable model failure points by analyzing its chain of thought reasoning (Figure 3.4-b).

On IRef-VLA, our method surpasses other methods on Hard statements by a large margin. IRef-VLA contains a large set of statements using size and color descriptions when referring to objects, which our method effectively grounds by utilizing object filtering, semantic attributes captured by captions, and the spatial reasoning toolbox to identify target objects with multiple distractors in the scene. Transcrib3D, when used in its zero-shot formulation with principles and not fine-tuned on the dataset's hard statements [20], fails to generalize well despite using the same LLM backbone. The results on IRef-VLA strongly support the benefits of incorporating 2D captions, as they provide critical, fine-grained object-level attributes that refine the referential grounding.

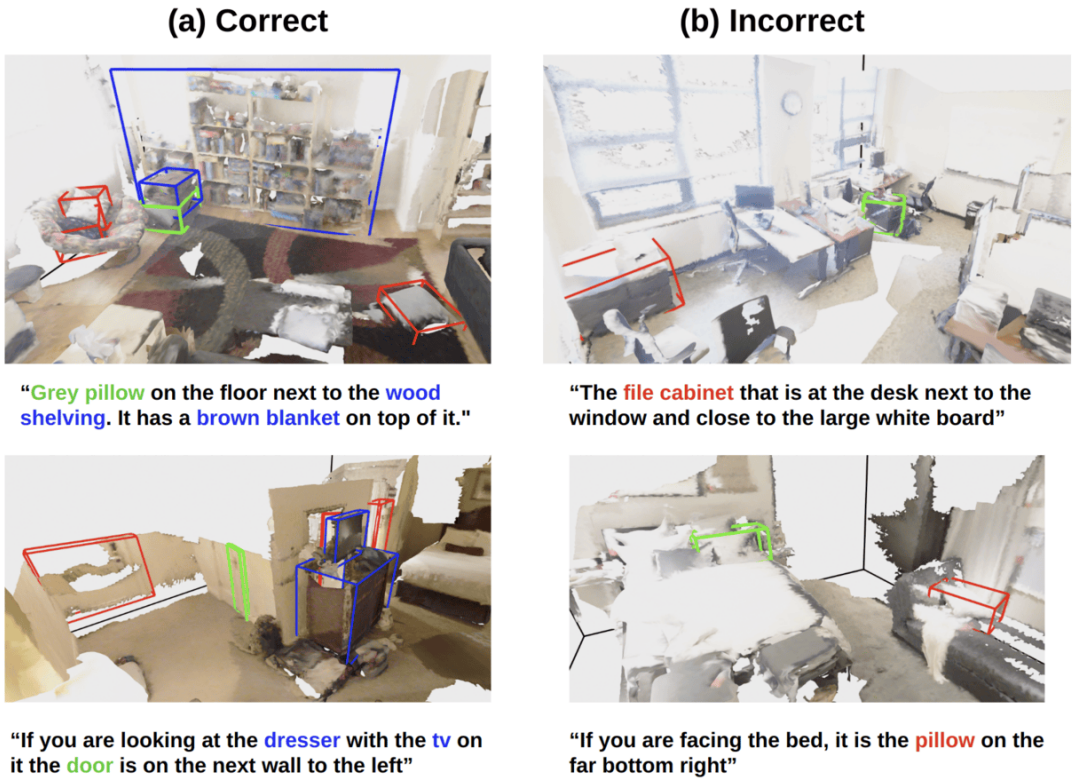


Figure 3.4: Correct (a) and incorrect (b) grounding examples. Top left and bottom left respectively show correctly grounded view-independent and view-dependent statements. Top right and bottom right are two examples of model logic failing: in the top right image, the model picks out the desk closest to a window, but not near the whiteboard. In the bottom right, the model fails at pragmatics, picking out the rightmost pillow, instead of recognizing that the sentence implies choosing a pillow on the bed.

Table 3.2: Performance on the Nr3D dataset from the ReferIt3D benchmark. Asterisks (*) indicate results reported from the paper directly. "View Dep." and "View Ind." stand for view-dependent and view-independent respectively.

	Nr3D		
Method	Overall	View Dep.	View Ind.
Supervised Methods			
NS3D* [26]	62.7	62.0	-
ViL3DRel* [14]	64.4	62.0	64.5
3D-VisTA* [84]	64.2	61.5	65.1
SceneVerse-GPS* [33]	64.9	56.9	67.9
Zero-Shot Methods			
ZSVG3D* [79]	39.0	36.8	40.0
VLM-Grounder* [75]	48.0	45.8	49.4
CSVG* [78]	59.2	53.0	62.5
Transcrib3D* [20] (GPT-4)	**70.2**	**60.1**	**75.4**
Transcrib3D (GPT-4o)	**65.6**	**63.3**	**66.7**
Transcrib3D (Mistral)	63.8	57.1	**66.7**
Ours (GPT-4o)	62.0±1.2	56.6±0.0	64.3±1.7
Ours (Mistral)	61.6±0.3	59.4±0.9	62.6±0.9

Table 3.3: Performance on the Sr3D dataset from the ReferIt3D benchmark. Asterisks (*) indicate results reported from the paper directly.

	Sr3D		
Method	Overall	View Dep.	View Ind.
Supervised Methods			
ViL3DRel* [14]	72.8	63.8	73.2
3D-VisTA* [84]	76.4	58.9	77.3
SceneVerse-GPS* [33]	77.5	62.8	78.2
Zero-Shot Methods			
Transcrib3D* [20] (GPT-4)	98.4	98.2	98.4
Transcrib3D (GPT-4o)	**96.5**	88.9	**96.9**
Transcrib3D (Mistral)	96.0	77.8	**96.9**
Ours (Mistral)	92.0±0.7	90.9±0.0	92.2±0.8
Ours (GPT-4o)	92.0±0.0	**95.5±0.0**	91.6±0.0

Table 3.4: Grounding accuracy on IRef-VLA test subset

	IRef-VLA		
Method	Overall	Easy	Hard
Transcrib3D (Mistral)	70.5	76.25	47.5
Transcrib3D (GPT-4o)	**77.5**	**82.5**	57.5
Ours (Mistral)	69.0±0.7	70.0±0.8	65.0±0.0
Ours (GPT-4o)	71.8±1.8	71.0±1.3	**75.0±3.5**

3.6.2 Ablation of Captioning Module

We evaluate the effect on grounding accuracy of adding open-vocabulary captions generated from 2D images of objects in the scene. We augment the Transcrib3D [20] baseline model with our captions for each object as additional information passed into the LLM reasoner. We hypothesize that these finer descriptions of object attributes will help the model in disambiguating objects when given free-form referential statements. We test this hypothesis through an ablation study using GPT-4o with both our method and Transcrib3D on Nr3D, shown in Table 3.5. In both Transcrib3D and our method, we observe consistent and significant improvements across all statements types. For our approach specifically, the addition of captions improves performance the most (11%) on view-dependent statements. These results demonstrate that understanding detailed object attributes is important for effective 3D grounding and leveraging 2D VLMs is an effective method for this. Many referential statements rely on subtle distinctions—such as color, shape, texture, or affordance—that traditional 3D models often miss.

3.6.3 Real-World Validation

To validate SORT3D's pipeline in the real-world, we implement the full system on two autonomous mobile robots: a mecanum-wheeled robot and a differential drive robot with a wheelchair base, each equipped with 360-degree cameras, respective Livox and Velodyne LiDARS, onboard Intel NUCs for low level autonomy, and RTX 4090s with 16GB and 24GB of VRAM respectively. We validate our system in two previously unseen indoor environments: a student lounge (Figure 3.6, and a university

Table 3.5: Grounding accuracy with and without captions on Nr3D

	Nr3D				
Method	Overall	Easy	Hard	View Dep.	View Ind.
Transcrib3D (GPT-4o)	58.5	67.5	45.0	54.0	60.6
Transcrib3D (GPT-4o + captions)	61.0 (↑4.3%)	70.8 (↑4.9%)	46.3 (↑2.9%)	55.4 (↑2.6%)	63.5 (↑4.8%)
Ours (GPT-4o)	54.5	59.2	47.5	50.7	56.2
Ours (GPT-4o + captions)	60.5 (↑11.0%)	64.2 (↑8.4%)	55.0 (↑5.3%)	56.6 (↑**11.6%**)	62.3 (↑10.9%)

corridor (Figure 3.5). We attempt two different types of navigational queries, which target the system's ability to ground statements that employ both spatial references and object semantic attributes. Before issuing a query, we navigate each robot around the scene to build a semantic map after prompting the open vocabulary detector with the names and synonyms of objects in the scene (shown in RViz in the bottom of Figures 3.6 and 3.5), and collect image crops for each detected object. As an implementation detail, captions are batch-generated only after a user query is typed into the system to speed up semantic mapping and decrease overall power consumption. RViz visualizations of the instance level semantic maps, the objects and corresponding waypoints chosen by the grounding model for each statement, and pictures of each platform navigating through its environment are shown in Figures 3.6 and 3.5. In each statement we test, SORT3D successfully grounds one or more referenced objects, demonstrating the versatility of our approach for grounding complex expressions involving spatial references and semantic attributes in previously unseen scenes. Further experiments on more environments and types of statements are found in our accompanying repository and videos.

3.7 Limitations

We acknowledge that limitations exist within our approach. First, our system relies on internet access to call online APIs, which is a reasonable assumption for indoor

Figure 3.5: Navigation on the mecanum robot in a university corridor given the statement "Go to the table next to the bookshelf, then to the chair next to the plant." The system successfully grounds then navigates to both spatially referenced objects.

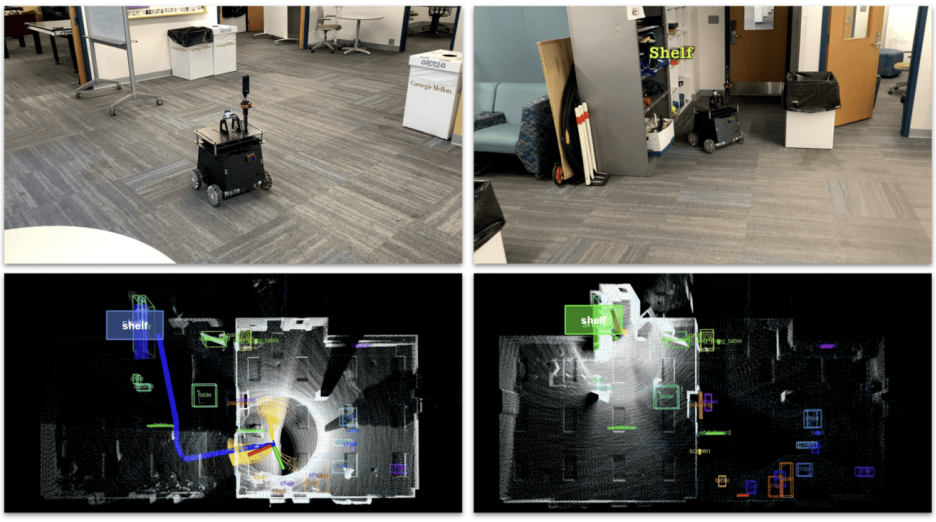


Figure 3.6: Navigation on the mecanum robot in the student lounge given the statement "I want to play a board game, fetch me one from the shelf." The system successfully navigates to the shelf with the board games guided by the semantic caption descriptions when there are multiple shelves in the room.

environments like homes and offices, but may not hold for other environments where generalizable robot systems may need to be deployed. To address this limitation, the modular design of our pipeline allows for the LLMs to be easily replaced by smaller local models.

We also recognize that our evaluation set is limited. There are few existing benchmarks designed to rigorously test online 3D referential grounding with diverse, attribute-rich natural language. This constraint makes it challenging to fully assess the generalizability of our approach across different environments or while deployed on a system. Evaluating our method in a simulated environment with real-time interactions would provide deeper insights into its effectiveness and adaptability as a system. It would also allow us to investigate more practical settings such as multi-turn grounding scenarios and failure correction.

3.8 Conclusion

We introduce SORT3D, a robust, data-efficient, and deployable method for 3D referential grounding with complex view-dependent reasoning. Our framework combines 2D visual features, LLM-based filtering, and a heuristics-driven spatial reasoning toolbox to enhance LLMs' ability to handle spatial and mathematical reasoning in 3D scenes. SORT3D achieves results that are competitive to SOTA methods on view-dependent and attribute-based statements while outperforming other fully zero-shot methods. We further demonstrate its performance and generalization by deploying the pipeline on a robotic system for real-time indoor navigation.

3.9 Future Work: VLN in Outdoor Environments

An immediate follow-up direction is to extend the SORT3D framework to outdoor environments. In outdoor environments, autonomous driving is one area in which an online vision-language navigation interface would be practical as human users could dictate certain actions for their vehicle to take based on preferences. One scenario is parking in urban settings, e.g. "park under the tree, in front of the red car". Another, far more dynamic scenario, is lane changing, where an instruction such as "change to

the left lane to follow behind the silver van" could be given.

Due to the modular structure of the pipeline, extending to outdoor environments only requires altering the semantic mapping with 3D object detection module and the spatial toolbox functions to adapt to the task. Depending on the task, the pipeline may need to ground regions or sections of open space, such as parking spots, in addition to objects, before passing the output to a lower level action module. As many of the components in SORT3D such as the 2D captioning module and LLM reasoning module leverage open-vocabulary and open-world foundation models, little change is required for such components though they could potentially benefit from fine-tuning for specific outdoor use cases.

To the best of our knowledge, there is a lack of diverse, real-world benchmarks for object-referential vision-language navigation in the autonomous driving domain. Talk2Car [19], CARLA-NAV [32], and SUP-AD [69] are existing datasets that can potentially be leveraged for training and evaluation while we leave the curation of a comprehensive benchmark for future work.

3.10 Acknowledgements

This chapter was jointly authored with Nader Zantout, and contributed to by Pujith Kachana, Jinkai Qiu, Guofei Chen, Wenshan Wang, and Ji Zhang.

Chapter 4

Towards Interactive Vision-Language Navigation with Dialogue Ambiguity Resolution

4.1 Introduction

While progress has been made in single-step vision-language tasks for downstream navigation, there still exists a gap between advances on structured benchmarks and real-world deployment. Although the SORT3D pipeline in Chapter 3 works online in a real-world navigation system, the setup itself and the SOTA baselines we compare to still makes simplifying assumptions on the task compared to a practical use case. One major assumption made is that instructions given by the user are complete, unambiguous, and correct, and that any given task can be completed in one execution attempt. In real-world settings however, human instructions are often vague, incomplete, or misaligned with the environment. With object-referential grounding as a basis, we define **ambiguous vision-language navigation (VLN) instructions** as instructions where there are multiple possible choices for referenced target object and choosing the incorrect one would be undesirable fo the user. Existing methods for vision-language navigation often cannot handle this uncertainty, nor handle further clarifications. For robot applications like home assistants, it is necessary for their

interactions with humans to be iterative and adaptive, to fit with evolving instructions and goals.

Dialogue-enabled VLN is a future direction that would allow agents to engage in such interactions with users by communicating uncertainty or asking clarifying questions. However, it comes with a number of challenges. First, such a task requires the storing of context of past interactions or conversation history, which can be computationally expensive, as well as difficult to maintain for longer context scenarios. Second, there is a lack of standardized benchmarks to evaluate the performance of this task. To begin with, human preferences differ and the correctness of dialogue can be subjective. For example, defining what it means to "Ask the best clarifying question" itself is difficult and how this is measured varies among existing work. Evaluating the quality of an alternative suggestion or clarification question can be difficult without a universal benchmark or even a lack of dialogue data. Human data curation or annotation is additionally expensive to obtain.

Each of the challenges is an opportunity for future work before truly interactive systems can be deployed in human-centric environments. This chapter aims to discuss existing related works, initial steps, and the future directions for interactive vision-language navigation focused on the subtask of ambiguity resolution.

4.2 Related Work

4.2.1 Uncertainty Detection in Instruction Following Tasks

Several works have explored the first subtask of ambiguity resolution, which is to identify when an instruction is uncertain or ambiguous given what is perceived from the environment. KnowNo [53] developed a conformal-prediction-based framework to detect incomplete or ambiguous task instructions, allowing the system to proactively seek clarification before execution during task planning, however, it requires a calibration dataset. [47] comes up with an ambiguity detection module for dialogue based on thresholding and a fixed template of clarification questions to choose from. More recently, other works [42] study and benchmark the ability of large reasoning models to recognize and communicate their own uncertainty.

4.2.2 Benchmarking Ambiguity Resolution

A number of robot instruction-following datasets have implicitly included ambiguous natural language samples, while not specifically focusing on these. For example, large-scale benchmarks for vision-language navigation such as Room-to-Room (R2R) [6] and Cooperative Vision-and-Dialog Navigation (CVDN) [68] sometimes contains under-specified descriptions that require pragmatic inference or dialogue to resolve. However, these datasets do not have any ground-truth annotation as to how these should be resolved. Similarly, robot task-oriented benchmarks such as ALFRED [61] and TEACH [45] include ambiguity through partial observability, and object co-referencing across multi-step goals. [21] specifically includes dialogue with ideal questions to ask to complete the task. Although these benchmarks do not explicitly annotate ambiguity, they help motivate the development of interactive and uncertainty-aware agents.

More recent work has proposed benchmarks that directly focus on ambiguity. Situated Instruction Following [43] introduces controlled ambiguous instructions in embodied 3D environments to study situated language grounding in different environments. KnowNo [53] defines a small diagnostic benchmark for detecting incomplete or underspecified task instructions, using conformal prediction as a formal tool for identifying uncertainty. Other datasets [46, 64] define ambiguity only as underspecified instructions. AmbiK [30] goes further by curating a dataset that categorizes multiple forms of ambiguity during robot task planning, including ambiguity about preferences, commonsense knowledge, and safety, enabling finer-grained evaluation of both detection and resolution strategies. However, their dataset is not focused on vision-language navigation and does not provide a comprehensive taxonomy of ambiguity types.

Together, these benchmarks contribute to the shift from ambiguous instructions as noise to explicitly formulating this as a vision-language problem and benchmarking it. This shift also prompts the need for metrics to systematically measure ambiguity, as well as taxonomies and ambiguity resolution methods. In Section 4.4, we aim to provide a taxonomy of ambiguities for object-centric navigation.

4.2.3 Task Clarification in VLN

Several works in embodied instruction following have explored enabling agents to ask clarifying questions when they encounter ambiguous or underspecified instructions. Early approaches such as Just Ask [17] trained agents using RL to recognize confusion and query for help directly, without choosing what to ask. Building on this, Good Time to Ask [82] introduced a framework for strategically deciding when to ask for help based on uncertainty estimates during object-goal navigation. However, they only focus on whether a target object is in a specified location. More recent efforts have focused on generating clarification questions. ELBA [59] integrates a confusion detection module with a question-answer generator, enabling agents to decide at each step whether to ask a clarifying question grounded in their navigation goals and past states. Other works [63] leverage LLMs to guide question generation when ambiguity is detected after computing uncertainty-aware similarity scores between detected objects and instructions. Most recently, Ask-to-Act [51] frames the task of asking clarifying questions as a reinforcement learning problem, where a multimodal LLM is tuned to provide reward signals that help the agent learn not only when to ask but also how to phrase effective questions. While a taxonomy of different ambiguous attributes is provided for their object-fetching task, the questions are limited to three types to simplify the problem.

Together, these works further emphasize that ambiguity resolution requires both reliable uncertainty detection and informative clarification questions, while there is remains a lack for a standardized taxonomy of ambiguity types, benchmark or evaluation metrics for the task. This motivates further research into developing both evaluation resources and integrated frameworks.

4.3 Problem Formulation

Beginning with IRef-VLA from Chapter 2, we have created a benchmark that loosens the assumption of perfect instructions by providing examples of misaligned referential grounding statements and the closest intended target object. While this evaluates scenarios where the instruction is incorrect and the agent should notify the user when the target object isn't found or suggest an alternative, ambiguity resolution differs

slightly in what's required for the agent. Depending on the type of ambiguity present, the agent must use it's perception outputs to determine a clarifying difference and ask a question that would resolve the *most ambiguity* assuming some valid target object exists and that the user is a privileged agent without intent to withhold information. More concretely, in the context of object-centric grounding for downstream VLN, we formulate this problem as follows:

Task and Environment. Let O denote the scene observation, K the object-referential user instruction and T the target object. The set of candidate targets is a nonempty set $\mathcal{N} = \{t_1, \ldots, t_m\}$. The output of a grounding model produces a posterior distribution over targets:

$$p_t \equiv p(T = t \mid K, O), \quad \sum_{t \in \mathcal{N}} p_t = 1$$

Ambiguity From an information-theoretic perspective, the expected amount of information to resolve the target can be characterized by the entropy, defined as:

$$H(T \mid K, O) = -\sum_{t \in \mathcal{N}} p_t \log p_t$$

Questions. Let Q_a denote a question about attribute a of an object (e.g., color, relative position, size, class, etc.). For each target t and attribute a, let $p(v \mid t, a)$ be the probability that attribute a of target t takes value $v \in \mathcal{V}_a$. With an ideal perception and grounding module, this probability will be either 0 or 1. The likelihood of answer $y = v$ to question Q_a is

$$p(y = v \mid t, Q_a) = p(v \mid t, a),$$

and the predictive distribution over candidate answers is:

$$p(y = v \mid Q_a) = \sum_{t \in \mathcal{N}} p_t \, p(v \mid t, a)$$

Belief Update. Upon receiving a clarifying answer y from the user, the posterior is updated via Bayes' rule:

$$p(t \mid K, O, y, Q_a) \propto p_t \, p(y \mid t, Q_a)$$

Information Gain. With a question and corresponding answer, the expected reduction in entropy can be calculated using the likelihood over all possible answers. This can also be interpreted as the expected information gain, IG to asking a question about attribute a:

$$IG(a) = H(p) - \sum_{v \in \mathcal{V}_a} p(v) \; H\left(\frac{p_t \, p(v \mid t, a)}{\sum_{t'} p_{t'} \, p(v \mid t', a)} \right),$$

where $p(v) = \sum_t p_t \, p(v \mid t, a)$ is the probability prior of each answer given the perception and $H(p) = -\sum_t p_t \log p_t$ is the entropy of the current distribution.

Objective. The objective of the task is thus to ask the question, Q_{a^*}, about the attribute a^* that maximizes the expected information gain.

$$a^\star = \arg\max_a IG(a)$$

4.4 Taxonomy of Ambiguity Types

Given the problem formulation, a taxonomy for types of ambiguity that might occur in real-world settings between an agent and user is critical to developing methods for resolving ambiguity, as well as analyzing results. Breaking down the types of uncertainty can also help elucidate the bottlenecks in a current VLN framework and whether improving perception, grounding, or semantic reasoning would aid better task success. We identify a taxonomy of types of ambiguity in instruction-following navigation tasks that require object-centric grounding in Table 4.1. Specifically, we identify three overarching categories of ambiguities, defined as:

- **Vision-Language Misalignment**: misalignment between the language instruction given and what is perceived in the scene

- **Agent-User Misalignment**: misalignment between the agent's beliefs and user's beliefs or perspective
- **Language Ambiguity**: inherent ambiguity in the language instruction, regardless of the observed scene

Ambiguity Type	**Description**	**Example**
Vision-Language Misalignment		
Referential Underspecification	Instruction does not provide enough detail to identify the intended object among multiple candidates	"Go to the chair" when there are several chairs in the room
Missing Object/Attribute	The described object or its attributes do not exist in the environment	"Find the red mug" when only blue mugs are present
Agent-User Misalignment		
Spatial/Relational	Vague or inconsistent inter-object spatial descriptions	"Stop not too far from to the lamp"
Orientation	Uncertainty in perspective-dependent terms like "left", "right", or "in front of"	"Turn left at the coffee table" (what's left is dependent on orientation)
Subjective	Instructions rely on personal judgment or subjective opinion	"Go to the nicer chair"
Language Ambiguity		
Syntactic	Multiple possible parses of an instruction that change meaning	"Go to the small table by the garbage can near the door" (what does "near the door" modify?)
Object Co-reference	Difficulty resolving pronouns or repeated references to the same object in a multi-step instruction	"Go to the table, then go to the TV, and stop by the flower pot near it."

Table 4.1: Taxonomy of ambiguity types in object-referential vision-language navigation

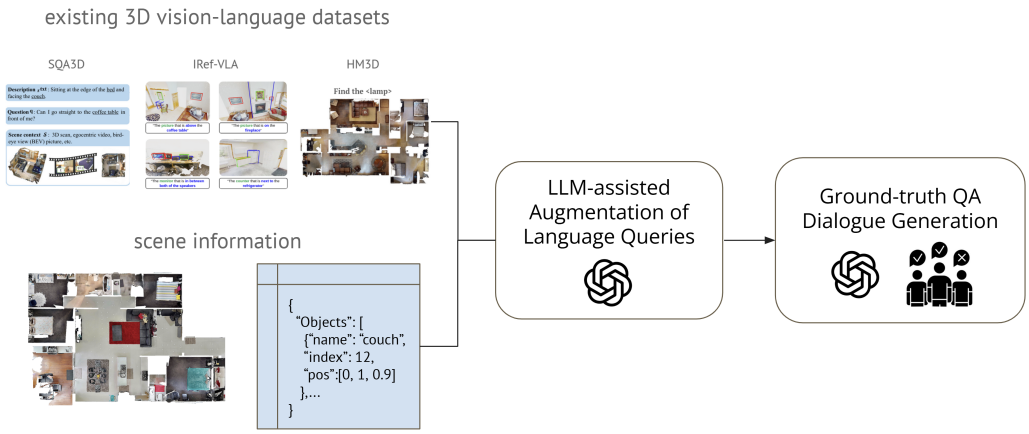


Figure 4.1: Example dataset creation pipeline for ambiguous language statements

4.5 Ambiguity Resolution Dataset and Pipeline

Looking ahead, we outline two tangible directions for future work: the creation of an ambiguity resolution dataset for the VLN setting, and a system pipeline for integrating an ambiguity resolution module. First, due to the lack of existing ambiguity resolution datasets, specifically for benchmarking settings such as VLN, we propose creating a new dataset according to Figure 4.1. Such a method could leverage the existing object-referential and language grounding datasets for various tasks such as situated reasoning, object-referential grounding, and object-goal navigation. Combining these with the extracted 3D scene information, LLMs can be used to augment the language statements or instructions into ambiguous statements categorized by the taxonomy previously proposed. Finally, to generate ground-truth multi-turn dialogue iterations where the agent asks questions to resolve the ambiguity, we could again leverage LLMs to automatically generate initial dialogue iterations to easily scale the data. At the same time, it is important for dialogue generation and evaluation to also involve human annotators to ensure alignment with human preferences and notions of ambiguity. The human annotators could verify the dialogue or make adjustments on what question, Q_{a*}, should be asked in each turn.

Second, we design a system-level pipeline design where a dialogue module for resolving ambiguity can be integrated into an existing perception and VLN framework, such as SORT3D. Figure 4.2 illustrates how a robot could monitor for uncertainty

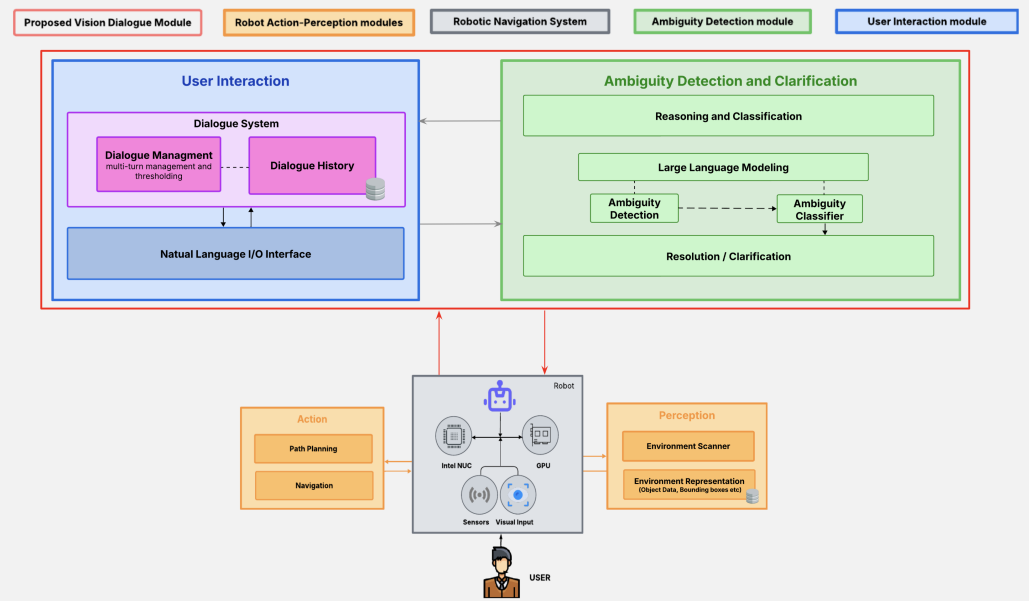


Figure 4.2: Example pipeline integrating ambiguity resolution and dialogue modules into a robot system

based on perceptual inputs in an ambiguity detection module, trigger a clarification step when ambiguity is detected, and then use dialogue to clarify the user's intent before acting and store these interactions. This could be separated into an ambiguity detection module internally and a user-facing dialogue module that engages in question-answering. Ideally, the dialogue system would store dialogue history or past queries and clarifications in a memory cache to retrieve in future iterations. The proposed design serves as a conceptual roadmap for building interactive, dialogue-enabled navigation systems in future work.

4.6 Conclusions and Future Work

In summary, this chapter highlights how moving from single-step vision-language navigation toward dialogue-based interaction is a necessary shift for building practical robot systems. Addressing ambiguity in human instructions requires not only robust perception and grounding to detect the ambiguity, but also the ability to engage in iterative communication to clarify user intent in as few steps as possible. Such a method should also be integrated into VLN pipelines in a modular and deployable

manner.

While current methods offer promising starting points, significant progress is still needed in data collection, developing evaluation standards, and storing contextual memory, to support such a system. By framing ambiguity resolution as both a central challenge, and natural next step from current VLN systems, we provide an analysis of existing work, formulate the problem mathematically, underscore the importance of standardized evaluation, and propose a taxonomy for ambiguity types as our contributions. We also propose an example of how an ambiguity and dialogue module may be integrated into existing VLN systems. Ultimately, advancing in this direction will be key for creating interactive vision-language navigation agents that can operate reliably in dynamic, everyday environments alongside humans.

4.7 Acknowledgements

The taxonomy of ambiguity types and system plan were jointly created with Calvin Qin, Anubhav Sharma, Yaqi Wang, and Wei Bin Au Yeong.

Chapter 5

Conclusions

This thesis explored how to develop deployable and interactive vision-language navigation agents through the fundamental task of object-centric language grounding, by addressing the challenges of data scale, real-time grounding, and exploring the potential for incorporating ambiguity resolution and interactive dialogue to complete navigation tasks.

Our first contribution, the IRef-VLA benchmark, provides the field with a large-scale benchmark that captures difficult spatial reasoning statements and the imperfections of referential language in 3D scenes. By curating diverse referential statements, semantic relations, and annotations at scale, it establishes a foundation for evaluating 3D spatial reasoning progress. Building on this resource, we contribute the SORT3D pipeline, which demonstrates that structured reasoning approaches combining heuristic tools with large language models can achieve strong zero-shot generalization, enabling real-time deployment of object-referential grounding systems in previously unseen environments. The modular design allows the pipeline to be easily adapted to other settings, such as outdoor autonomous driving. Finally, the exploration of dialogue-enabled VLN highlights an emerging frontier where ambiguity detection, clarification, and iterative interaction become essential for practical systems that operate with humans.

These contributions advance the path to robust language-driven navigation systems that can operate seamlessly with human users, which is a stepping stone towards the development of general-purpose robots. Moving forward, unifying 3D grounding,

reasoning, and dialogue in a single framework will be key to deploying systems that are not only capable, but robust to uncertainties inherent in real-world environments. Future directions crucial for deploying human-centric VLN systems include incorporating long-term memory, learning and adapting to new environments online, and storing spatio-semantic context.

Additionally, while the use of large foundation models has streamlined the process of incorporating web-scale knowledge and reasoning capabilities into semantic navigation systems, how to reliably deploy them onboard systems and ensure real-time action and reaction is still an ongoing challenge. To ensure alignment with users, validating such frameworks on physical robot systems alongside human users and incorporating human feedback will be essential before integrating robots into our homes and daily lives.

Bibliography

[1] Ahmed Abdelreheem, Filippo Aleotti, Jamie Watson, Zawar Qureshi, Abdelrahman Eldesokey, Peter Wonka, Gabriel Brostow, Sara Vicente, and Guillermo Garcia-Hernando. Placeit3d: Language-guided object placement in real 3d scenes. *arXiv preprint arXiv:2505.05288*, 2025. 2.10

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2.2

[3] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 422–440. Springer, 2020. (document), 2.3, 2.4.1, 2.7, 2.6, 2.6.1, 2.6.2, 3.2, 3.3, 3.5

[4] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 1

[5] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 2.4.1

[6] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2.3, 4.2.2

[7] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages

2425–2433, 2015. 2.2

[8] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 2.5.1

[9] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 2.4.1, 3.2

[10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 2.5.1, 3.2

[11] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021. 2.2

[12] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024. 1

[13] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020. 2.3, 2.4.1, 3.3

[14] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in neural information processing systems*, 35:20522–20535, 2022. 3.3, 3.6.1, 3.2, 3.3

[15] Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024. 3.2, 3.3

[16] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language model, 2024. URL https://arxiv.org/abs/2406.01584. 2.3

[17] Ta-Chung Chi, Minmin Shen, Mihail Eric, Seokhwan Kim, and Dilek Hakkani-Tur. Just ask: An interactive learning framework for vision and language navigation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2459–2466, 2020. 4.2.3

[18] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2.5.1, 3.2, 3.4.2, 3.5

[19] Thierry Deruyttere, Dusan Grujicic, Matthew B Blaschko, and Marie-Francine Moens. Talk2car: Predicting physical trajectories for natural language commands. *Ieee Access*, 10:123809–123834, 2022. 3.9

[20] Jiading Fang, Xiangshan Tan, Shengjie Lin, Igor Vasiljevic, Vitor Guizilini, Hongyuan Mei, Rares Ambrus, Gregory Shakhnarovich, and Matthew R Walter. Transcrib3d: 3d referring expression resolution through large language models. *arXiv preprint arXiv:2404.19221*, 2024. 3.2, 3.3, 3.4.2, 3.4.3, 3.5.1, 3.6.1, 3.2, 3.3, 3.6.2

[21] 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. 4.2.2

[22] Richard E. Grandy and Richard Warner. Paul Grice. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2023 edition, 2023. 3

[23] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv preprint arXiv:2309.16650*, 2023. 2.3

[24] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 564–571, 2013. 2.5.2

[25] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models, 2023. URL https://arxiv.org/abs/2307.12981. 3.3

[26] Joy Hsu, Jiayuan Mao, and Jiajun Wu. Ns3d: Neuro-symbolic grounding of 3d objects and relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2023. 3.2, 3.3, 3.4.3, 3.2

[27] Jiaxin Huang, Ziwen Li, Hanlve Zhang, Runnan Chen, Xiao He, Yandong Guo, Wenping Wang, Tongliang Liu, and Mingming Gong. Surprise3d: A dataset for spatial understanding and reasoning in complex 3d scenes. *arXiv preprint arXiv:2507.07781*, 2025. 2.10

[28] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15524–15533, 2022. 2.2, 2.3, 2.2, 2.6.1,

3.2, 3.3, 3.4.2

[29] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *arXiv preprint arXiv:2201.13360*, 2022. 2.3

[30] Anastasiia Ivanova, Eva Bakaeva, Zoya Volovikova, Alexey K Kovalev, and Aleksandr I Panov. Ambik: Dataset of ambiguous tasks in kitchen environment. *arXiv preprint arXiv:2506.04089*, 2025. 4.2.2

[31] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *European Conference on Computer Vision*, pages 417–433. Springer, 2022. 2.3, 3.2, 3.3

[32] Kanishk Jain, Varun Chhangani, Amogh Tiwari, K Madhava Krishna, and Vineet Gandhi. Ground then navigate: Language-guided navigation in dynamic scenes. *arXiv preprint arXiv:2209.11972*, 2022. 3.9

[33] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. *arXiv preprint arXiv:2401.09340*, 2024. (document), 2.3, 2.7, 2.2, (ii), 3.3, 3.6.1, 3.2, 3.3

[34] Nassir Navab Federico Tombari Matthias Niessner Johanna Wald, Armen Avetisyan. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, 2019. 2.5.1

[35] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. *Advances in neural information processing systems*, 27, 2014. 2.2

[36] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 2.2

[37] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020. 2.3

[38] Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. *arXiv preprint arXiv:2412.04383*, 2024. 3.2

[39] Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. Navcot: Boosting llm-

based vision-and-language navigation via learning disentangled reasoning. *arXiv preprint arXiv:2403.07376*, 2024. 3.3

[40] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2.2

[41] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024. URL https://arxiv.org/abs/2303.05499. 3.4.1

[42] Zhiting Mei, Christina Zhang, Tenny Yin, Justin Lidard, Ola Shorinwa, and Anirudha Majumdar. Reasoning about uncertainty: Do reasoning models know when they don't know? *arXiv preprint arXiv:2506.18183*, 2025. 4.2.1

[43] So Yeon Min, Xavi Puig, Devendra Singh Chaplot, Tsung-Yen Yang, Akshara Rai, Priyam Parashar, Ruslan Salakhutdinov, Yonatan Bisk, and Roozbeh Mottaghi. Situated instruction following. In *European Conference on Computer Vision*, pages 202–228. Springer, 2024. 4.2.2

[44] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat, 2021. URL https://arxiv.org/abs/2110.00534. 2.3

[45] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025, 2022. 4.2.2

[46] Jeongeun Park, Seungwon Lim, Joonhyung Lee, Sangbeom Park, Minsuk Chang, Youngjae Yu, and Sungjoon Choi. Clara: classifying and disambiguating user commands for reliable interactive robotic agents. *IEEE Robotics and Automation Letters*, 9(2):1059–1066, 2023. 4.2.2

[47] Pradip Pramanick, Chayan Sarkar, Snehasis Banerjee, and Brojeshwar Bhowmick. Talk-to-resolve: Combining scene understanding and spatial dialogue to resolve granular task ambiguity for a collocated robot. *Robotics and Autonomous Systems*, 155:104183, 2022. 2.3, 4.2.1

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2.2

[49] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr

Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 2.2, 2.5.1, 3.2

[50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 2.2

[51] 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. *arXiv preprint arXiv:2504.00907*, 2025. 4.2.3

[52] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714. 3.4.1

[53] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023. 4.2.1, 4.2.2

[54] Allen Z Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until confident: Efficient exploration for embodied question answering. *arXiv preprint arXiv:2403.15941*, 2024. 1

[55] Gabriel Sarch, Lawrence Jang, Michael Tarr, William W Cohen, Kenneth Marino, and Katerina Fragkiadaki. Vlm agents generate their own memories: Distilling experience into embodied programs of thought. *Advances in Neural Information Processing Systems*, 37:75942–75985, 2024. 1

[56] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 1

[57] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action, 2022. URL https://arxiv.org/abs/2207.04429. 2.3

[58] Pratyusha Sharma, Balakumar Sundaralingam, Valts Blukis, Chris Paxton, Tucker Hermans, Antonio Torralba, Jacob Andreas, and Dieter Fox. Correcting robot plans with natural language feedback, 2022. URL https://arxiv.org/

abs/2204.05186. 2.3

[59] Ying Shen, Daniel Biś, Cynthia Lu, and Ismini Lourentzou. Elba: Learning by asking for embodied visual navigation and task completion. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5177–5186. IEEE, 2025. 4.2.3

[60] Mohit Shridhar and David Hsu. Interactive visual grounding of referring expressions for human-robot interaction. *arXiv preprint arXiv:1806.03831*, 2018. 2.3

[61] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 4.2.2

[62] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 2.5.2

[63] 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. *arXiv preprint arXiv:2412.01250*, 2024. 4.2.3

[64] Shohei Tanaka, Konosuke Yamasaki, Akishige Yuguchi, Seiya Kawano, Satoshi Nakamura, and Koichiro Yoshino. Do as i demand, not as i say: A dataset for developing a reflective life-support robot. *IEEE Access*, 12:11774–11784, 2024. 4.2.2

[65] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2.2

[66] Mistral AI Team. Mistral large 2: The new generation of flag- ship model. https://mistral.ai/news/mistral-large-2407/, 2024. [Accessed 01-03-2025]. 3.4.3

[67] Qwen Team. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671. 3.4.2

[68] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR, 2020. 4.2.2

[69] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao,

Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024. 3.9

[70] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2.2

[71] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3961–3970, 2020. 2.3

[72] Austin T Wang, ZeMing Gong, and Angel X Chang. Vigil3d: A linguistically diverse dataset for 3d visual grounding. *arXiv preprint arXiv:2501.01366*, 2025. 2.10

[73] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837, 2022. 3.2

[74] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024. 2.3

[75] Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. *arXiv preprint arXiv:2410.13860*, 2024. 3.2, 3.3, 3.2

[76] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7694–7701. IEEE, 2024. 3.2

[77] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 2.10

[78] Qihao Yuan, Jiaming Zhang, Kailai Li, and Rainer Stiefelhagen. Solving zero-shot 3d visual grounding as constraint satisfaction problems. *arXiv preprint arXiv:2411.14594*, 2024. 3.2, 3.3, 3.2

[79] Zhihao Yuan, Jinke Ren, Chun-Mei Feng, Hengshuang Zhao, Shuguang Cui, and

Zhen Li. Visual programming for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20623–20633, 2024. 3.3, 3.2

[80] Hanbo Zhang, Yunfan Lu, Cunjun Yu, David Hsu, Xuguang Lan, and Nanning Zheng. Invigorate: Interactive visual grounding and grasping in clutter. *arXiv preprint arXiv:2108.11092*, 2021. 2.3

[81] Haochen Zhang, Nader Zantout, Pujith Kachana, Ji Zhang, and Wenshan Wang. Iref-vla: A benchmark for interactive referential grounding with imperfect language in 3d scenes, 2025. URL https://arxiv.org/abs/2503.17406. 3.2, 3.3, 3.5

[82] Jenny Zhang, Samson Yu, Jiafei Duan, and Cheston Tan. Good time to ask: A learning framework for asking for help in embodied visual navigation. In *2023 20th International Conference on Ubiquitous Robots (UR)*, pages 503–509. IEEE, 2023. 4.2.3

[83] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7641–7649, 2024. 3.3

[84] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023. 2.3, 2.2, 2.6.1, 3.2, 3.3, 3.4.2, 3.6.1, 3.2, 3.3

[85] George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016. 2.2