Towards Practical Vision-and-Language Navigation Systems Through 3D Referential Grounding

Nader Zantout

CMU-RI-TR-25-60

July 11, 2025

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

Thesis Committee:
Wenshan Wang, *chair*
Ji Zhang, *co-chair*
Jean Oh
Ayush Jain

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

To my parents, Halima and Wissam.

Abstract

As robots transition toward practical deployment as collaborative agents in human environments, it becomes essential to improve language-conditioned environmental understanding. A vision-and-language navigation (VLN) system must adapt to both the types of language used and the actions expected by a human collaborator. Often, a single sentence containing spatial relations and semantic attributes—e.g., "fetch the yellow bottle on the table"—is all that is provided to specify a target object in a complex scene. The task of identifying the correct object from such a statement is known as *3D referential grounding*.

This thesis develops and deploys a practical VLN system through the lens of 3D referential grounding, a particularly challenging task due to the large number of objects in typical scenes and the relative scarcity of 3D data compared to 2D. We pursue two complementary approaches: (1) scaling up the training of an **end-to-end 3D referential grounding model**, and (2) decomposing the task into a **modular pipeline**.

First, we introduce **IRef-VLA**, a large-scale benchmark for **I**nteractive **Ref**erential **V**ision-and-**L**anguage-guided **A**ction. IRef-VLA aims to improve generalization in 3D referential grounding using synthetic utterances generated from scene graphs with view-independent spatial relations. Baseline models trained on IRef-VLA show strong zero-shot transfer performance, and an LLM-based graph search baseline achieves high grounding accuracy, motivating a modular alternative to end-to-end approaches.

We then explore this modular approach in **SORT3D**, a **S**patial **O**bject-centric **R**easoning **T**oolbox for 3D grounding with foundation models. SORT3D combines real-time semantic mapping, vision-language captioning, query-based object filtering, and structured spatial reasoning via LLMs into a deployable system. It demonstrates strong zero-shot performance across two benchmark datasets and on real-world robotic platforms operating in unseen environments.

Together, these systems establish a template for building effective collaborative embodied agents, where the ideal model is a middleground between fully end-to-end learning and a fully heuristics-based approach, and act as a springboard towards the creation of general purpose VLN systems deployable in all environments.

Acknowledgments

First and foremost, I extend my heartfelt gratitude to my advisors, Ji Zhang and Wenshan Wang. You have given me the opportunity to perform research in what I believe is by far the most interesting area in modern robotics. I had the opportunity to grow alongside you two, and your support has pushed me towards making the most of my time at CMU, and becoming a roboticist I have great pride in.

On a similar level, I want to thank my teammates, Haochen Zhang and Pujith Kachana, for being incredible collaborators and dear friends. None of the work I present is thesis would have been remotely possible without you two. I have learned from you two a great deal throughout my master's here at CMU, and for that I am incredibly grateful. I would also like to thank Guofei Chen, for the late nights we spent debugging semantic mapping modules in time for demos, and for being a great friend, and a wonderful human being.

On a slightly different note, I am immensely grateful to Yonatan Bisk. Though our in-person meetings were brief, the insights I have gained from you have completely reshaped the way I think of language, and significantly driven my research forward.

Words would do little justice to describe how grateful I am to MSR Alumnus Yves-Georgy Daoud. Even from before my arrival to Pittsburgh, you have been an incredible guide, mentor, and friend. You helped put my beaten, battered self back onto its feet during some of my lowest times. I have little clue on how to repay you for all you've given me, so thank you, Yves, for all you've done.

I additionally thank Mohamad Qadri for being a fun mentor and a fun person, and Haokun Zhu and Avigyan Bhattacharya for being good friends.

I would like to thank my thesis committee members, Jean Oh and Ayush Jain. Your work has inspired me deeply throughout my master's.

Last, far from least, are my parents, Halima and Wissam. I could write a section ten times the length of this thesis and I would not even begin to scratch the surface of acknowledging what you've done for me. For supporting me every moment of my life, and supporting my ambition to

get here, unflinchingly and unconditionally. Thank you. This thesis is just as much your gargantuan efforts as it is mine.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, modern robotics has undergone a significant paradigm shift—from robots used for task-specific applications with highly specialized autonomy systems (both learned and hand-crafted), towards more general robotic systems with high level intelligence that can generalize to many different scenarios. Fueled by the emergence of foundation vision-and-language models trained on internet-scale data, the shift in research towards creating **Embodied Artificial General Intelligence (AGI)** has led to impressive demonstrations of embodied agents acting in unseen scenarios both in academia and in industry, along with the creation of **Vision-Language-Action (VLA)** policies achieving unprecedented performance on a vast array of different scenes and tasks [3, 4].

Within the broad landscape of VLA policies, we place a particular focus on the subdomain of **Vision-and-Language Navigation (VLN)** for mobile robots in this thesis. We concretely define VLN as *the ability of a mobile agent to navigate through its environment given its perceptual input and a set of language commands whose interpretation results in the agent taking navigational actions towards a particular goal.* These instructions range in the granularity of the actions they describe and the amount of reasoning required to follow them, from sequences describing every movement performed by a robot as it follows a trajectory to a goal (e.g. "head upstairs and walk past the piano, then turn right at the end of the hallway...") [5, 6], to queries only specifying a goal object to navigate to (e.g. “go to the painting next to the fridge”) [7], to commands that require implicit planning and interpretation (e.g. “fetch me my print”).

While end-to-end trajectory following remains an open challenge in VLN research, the acceleration of the field towards the practical deployment of assistive embodied agents in human-shared workspaces has necessitated new avenues of VLN research focusing on commands that are more commonly uttered by humans. For instance, while describing the full path from a downstairs living room sofa to a dresser in an upstairs bedroom is useful when the agent the path is being described to is unfamiliar with the layout of the house, a person later requesting an interaction with the dresser would more often just refer to it as ”the dresser in the bedroom”, requiring the agent to reason about it from just that utterance. The agent, in this formulation, is required to ground the correct object, location, or series of objects and locations being referred to within its internal representation of its environment, and plan a correct path towards these objects or locations—a task known as **three-dimensional (3D) referential grounding**.

1.2 Problem Statement

This thesis addresses the development of a practical VLN system by focusing on 3D referential grounding, which is foundational to enabling a robot’s language-guided understanding of its environment. Large vision-language models (VLMs) have shown impressive capabilities on 2D visual grounding powered by billion-scale image datasets [8], yet end-to-end 3D referential grounding models remain far less generalizable and capable due to the higher complexity of 3D scenes which cannot be captured by single 2D images and the relative lack of 3D data, and by extension full scene understanding remains difficult for VLN systems. This task is therefore approached from two major perspectives: scaling up the training of an **end-to-end 3D referential grounding model** for indoor scenes, followed by **decomposing the task of referential grounding into a modularized approach**, and leveraging foundation models to create a VLN system deployable in real life.

1.3 Contributions

In chapter 2 of this thesis, we tackle scaling up the training of end-to-end referential grounding models through creating **IRef-VLA**, a large scale dataset targeting **I**nteractive **Ref**erential **V**ision and **L**anguage-guided **A**ction in 3D Scenes as a novel benchmark for 3D referential grounding [9]. Using heuristics to automatically generate graphs of view-independent spatial relationships between objects and language templates to create referential statements, we improve the generalizability of referential grounding baselines to different scenes and referential statements from unseen distributions, which we refer to as *zero-shot transfer performance*. We additionally create an augmented version of the referential grounding task using imperfect references for future research into multi-turn dialogue for VLN systems, and implement an LLM-based graph search method that searches for object existence in the spatial scene graphs and suggests alternatives if needed. The strong zero-shot transfer performance of models trained on IRef-VLA, combined with the effectiveness of the LLM-based graph search baseline in grounding object existence, suggests a promising alternative to end-to-end 3D referential grounding. Rather than grounding the entire sentence in a single step, this approach decomposes the task into modular stages: instance segmentation and scene graph construction, followed by transforming the input utterance into a structured graph search query that is executed over the spatial scene graph.

We pursue this modular approach in chapter 3. We propose **SORT3D**, a **S**patial **O**bject-centric **R**easoning **T**oolbox for **3D** Grounding Using LLMs, as a modular approach to 3D grounding targeting real-world deployment as the centerpiece of a practical VLN system [10]. SORT3D employs a semantic mapping module for real-time object instance segmentation, a 2D VLM captioning module to extract rich object semantic attributes for use in natural language grounding, a filtering module that only keeps objects relevant to the query, and an LLM-based spatial reasoning module augmented with a spatial reasoning toolbox to resolve the target object. By leveraging strong linguistic priors in LLMs and strong visual priors in 2D VLMs, our approach achieves competitive zero-shot performance on view-dependent 3D referential grounding on two benchmarks—IRef-VLA and ReferIt3D [1]. More importantly, when deployed on two robotic ground vehicles in completely unseen real-world environments, SORT3D successfully follows object navigation instructions with complex spatial relations and semantic attribute references along with implicit reasoning, making SORT3D a successful template and schema for developing practical embodied AI systems.

We conclude our argument in chapter 4, having established a paradigm for building practical VLN systems with IRef-VLA and SORT3D, with a brief comparison between end-to-end and modular approaches to VLN, discussing the optimal strategy as we progress towards embodied AGI.

Chapter 2

Scaling Referential Grounding Datasets to Train End-to-End Object-centric Navigation Models

2.1 Abstract

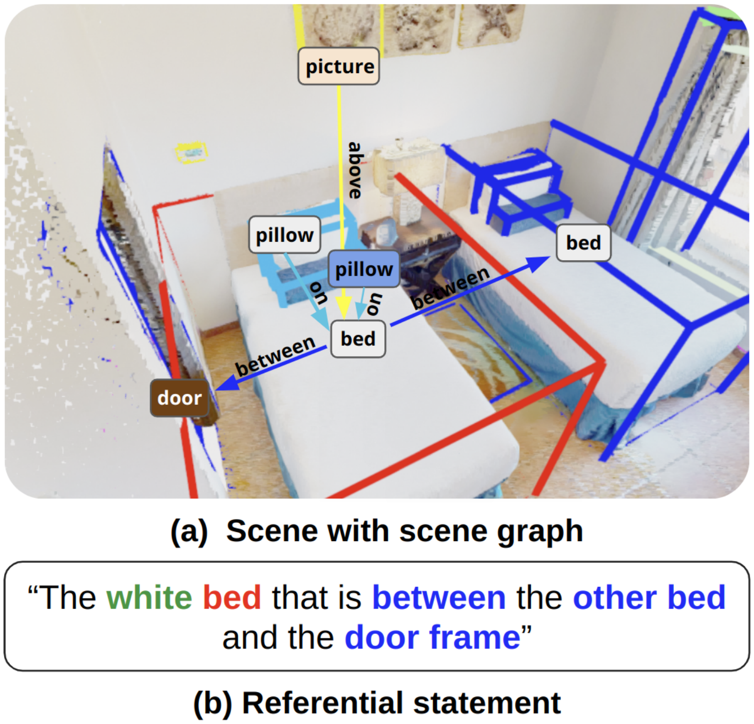
With the recent rise of large language models, vision-language models, and other general foundation models, there is growing potential for multimodal, multi-task robotics that can operate in diverse environments given natural language input. One such application is indoor navigation using natural language 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 available at `https://github.com/HaochenZ11/IRef-VLA`.

2.2 Introduction

The advent of Large Language Models (LLMs) [11, 12, 13] and Vision-Language Models (VLMs) [14, 15, 16] pre-trained on internet-scale data has led to a rapid evolution in multimodal intelligence, evident by a vast improvement on various language-image tasks such as Visual Question Answering (VQA) [17], image retrieval [18], and image captioning [14]. This, in turn, has led to renewed excitement in the robotics research community towards developing embodied general intelligence [3, 4], and with it, the need for methods that are capable of both reasoning about large 3D scenes and interacting with humans. Particularly, the ability to understand natural language and ground that language to the physical world are key skills for interactive robots. For example, in collaborative settings requiring object interactions, humans often refer to objects with minimal but sufficient linguistic descriptions to elicit the desired interaction from a collaborator [19]. Given a referential statement such as "fetch the red mug on the top shelf", a human collaborator is immediately able to intuit the correct object type and location and perform the requested action. Consequently, an effective indoor assistive robot must have the ability to 1) similarly solve such a problem, 2) handle imperfect or ambiguous language, and 3) interact with humans to achieve the intended goal.

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. Aside from the difficulties in low level perception and scene representation, associating a language expression with a particular object in the scene, or **3D referential grounding**, is itself a significantly complex and challenging task. On one hand, a scene can have hundreds of objects, contain objects belonging to fine-grained classes, and have many similar objects [20].



(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.

On another hand, human referential language often involves spatial reasoning, implicit and explicit affordances, and synonymous expressions, and may even be incorrect or refer to something that does not exist—for example, a human may want to fetch *"the remote on the table"* when the remote is actually on the sofa. Additionally, the scale of available vision-language data in the 3D space pales in comparison to the amount of 2D data, which has been crucial to the success of 2D vision-language learning methods [21, 22]. As a result, a significant gap remains between the performance of 2D VLMs on referring expression comprehension with 2D images [23] and that of 3D end-to-end models on scene-level referential grounding [24, 25]. Current end-to-end models therefore fail to offer the accuracy and robustness needed for creating vision-and-language navigation (VLN) systems for assistive embodied agents.

To improve language-aligned scene understanding and 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 Fig. 2.1.

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

2.3.1 Referential Object Grounding Datasets

The task of referential object grounding has been extensively studied in 3D datasets such as ReferIt3D [1], ScanRefer [26], and SceneVerse [2]. ReferIt3D and ScanRefer establish standard benchmarks on the ScanNet dataset [27], offering both synthetically generated utterances focused solely on spatial relations (Sr3D), as well as human-annotated expressions that incorporate a broader range of references, including spatial cues and object attributes (Nr3D and ScanRefer) [1, 26]. SceneVerse scales up data collection by generating a significantly larger set of referential expressions—initially via templates and then rephrased using an LLM. However, similar to Sr3D, these expressions are limited to spatial relationships and omit explicit references to object attributes such as color, size, or shape—features commonly used by humans in referential language. As a result, models trained on SceneVerse generalize poorly to the Nr3D benchmark [2], and consequently, we include color and size object attributes in IRef-VLA to more closely approach natural referential language.

2.3.2 Semantic Scene Graph Datasets

Generating scene graphs from 3D scenes has also been explored in 3DSSG [28], Hydra [29], HOV-SG [30], and ConceptGraphs [31]. 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.

2.3.3 Trajectory-Following Datasets

While this work focuses on building VLN systems through the lens of referential grounding, many prior efforts have centered on the classical subtask of *trajectory following*—where an agent must interpret a set of instructions describing a full trajectory and execute the corresponding series of actions to reach a goal from a known starting point. Prominent benchmark datasets for this task include Room-to-Room [5] and Room Across Room [6], both built on Matterport3D [32], which provide sequences of language instructions tied to specific navigation paths.

Although trajectory following remains an important and unsolved challenge, it does not reflect the typical distribution of language used in real-world human-agent interactions. In practice, a human might only specify a full trajectory when the agent is unfamiliar with the environment; once familiar, commands are more likely to be short referential statements like “go to the red sofa.” Supporting such interactions requires the agent to maintain an internal representation of the scene, localize itself, interpret under-specified language, and plan a viable path—even under changes in layout or ambiguity.

Thus, focusing on referential grounding over full-path trajectory execution aligns more closely with the demands of practical collaborative agents.

2.3.4 Referential Object Grounding Baselines

A number of works have addressed referential object grounding, primarily on the ReferIt3D and ScanRefer benchmarks. These include models such as BUTD-DETR [33], MVT [24], ViL3DRel [26], 3D-VisTA [25], and GPS trained on SceneVerse [2]. Despite massively upscaling training data in terms of both the number of scenes and language, GPS only achieves 64.9% accuracy on Nr3D [2] and exhibits limited zero-shot generalization to novel scenes and language. In addition to their poor zero-shot performance, these models are 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.

2.3.5 Language Interaction in Embodied Agents

Some works [34, 35, 36] 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 [37] offers a human-generated dataset of task-driven dialogues for language grounding, dialogue understanding and task reasoning. [38] 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, [39] 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 IRef-VLA: A Benchmark for Interactive Referential Grounding with Imperfect Language in 3D Scenes

To advance the creation of robust interactive navigation agents, we introduce IRef-VLA, a synthetically-generated, publicly available benchmark dataset. It combines 3D scans from five real-world datasets: ScanNet [27], Matterport3D [32], Habitat-Matterport 3D (HM3D) [20], 3RScan [40], and ARKitScenes [41], as well as high fidelity custom scenes built in Unity. Fig. 2.2 shows the distribution of regions from each source. Each scene includes:

1. A scene point cloud.
2. A list of objects with semantic class labels, bounding boxes, and prevalent colors.

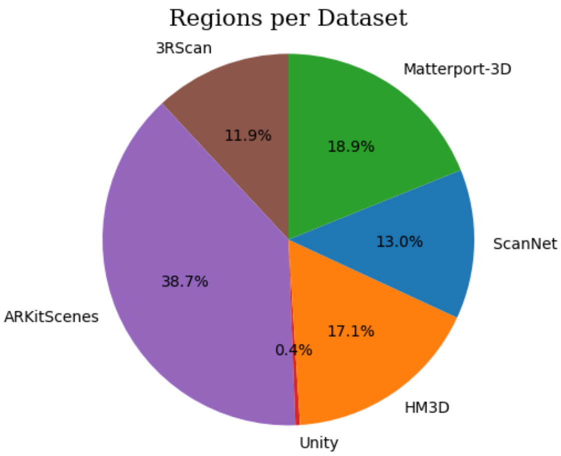


Figure 2.2: Breakdown of regions from each data source.

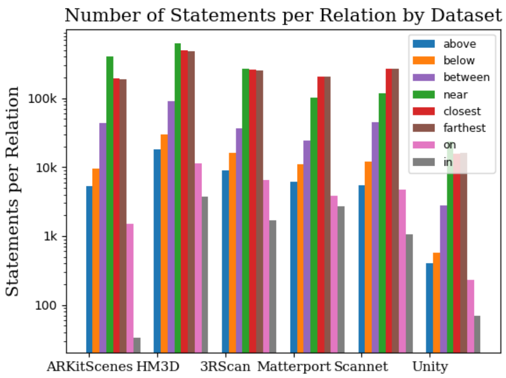


Figure 2.3: Number of statements per relation type from each dataset processed.

3. A list of traversable free spaces.
4. A list of regions with semantic labels and bounding boxes.
5. A scene graph of spatial relations split by room.
6. Referring expressions with ground-truth annotations for the target objects, along with augmented imperfect language statements.

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 across 477 classes, 7.6 million inter-object spatial relations, and 4.7 million referential statements, vastly upscaling the available data for 3D referential grounding both in terms of the number of scenes and in terms of the number of referential statements. Fig. 2.3 shows the distribution of spatial relations per dataset. The dataset processing pipeline, summarized in Fig. 2.4, is delineated in the following sections.

2.4.1 3D Scan Processing

Even though each 3D dataset provides reconstructed meshes, we only provide point clouds sampled from the meshes. This is in line with the SceneVerse [2] processing scheme, and most state of the art baselines only use point clouds sampled from reconstructed meshes. The PLY mesh vertices for ARKitScenes, Matterport3D, and

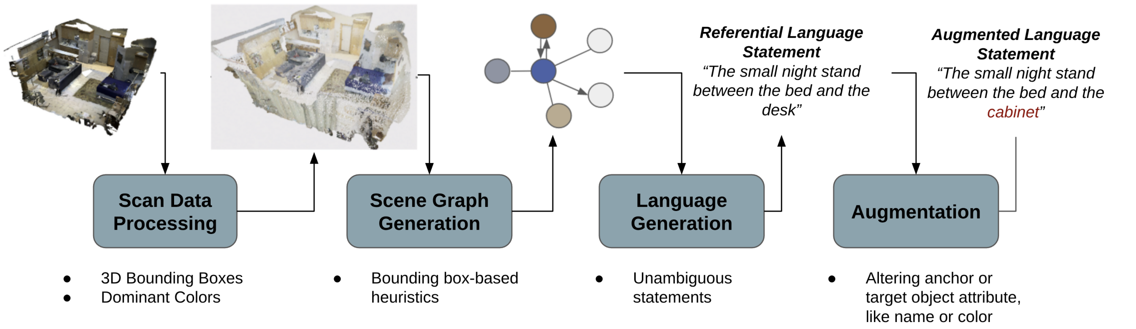

Referential Language Statement
"The small night stand between the bed and the desk"
Augmented Language Statement
"The small night stand between the bed and the cabinet"
Scan Data Processing
Scene Graph Generation
Language Generation
Augmentation
3D Bounding Boxes
Dominant Colors
Bounding box-based heuristics
Unambiguous statements
Altering anchor or target object attribute, like name or color

Figure 2.4: Data processing pipeline consisting of: 3D scan processing, scene graph Generation, language generation, and augmented statement generation.

ScanNet, along with their associated vertex colors, are directly used as the point clouds for scenes from these datasets. For HM3D, Unity, and 3RScan, point clouds are uniformly sampled from scene meshes at a density proportional to the total area of each mesh, and colors are sampled from UV textures.

Object bounding boxes are derived from the 3D instance segmentations provided in each dataset. The bounding boxes are oriented about the z-axis following ScanNet [27], where the rotation is obtained by fitting a minimal-area bounding rectangle to the object's 2D convex hull in the xy-plane. Region bounding boxes are also obtained from object instance segmentations and object-region mappings. ARKitScenes, 3RScan, and ScanNet have single-room scenes and hence one region per scene, while Matterport3D and HM3D provide object-region mappings, and Unity scene regions are custom-segmented. Each object is labeled with an open-vocabulary class name from its original dataset, mapped to NYU40 [42] and NYUv2 [43] schemas with the provided mappings[1]. The dominant three colors are obtained for each object by clustering the point cloud colors about the nearest CSS3 colors in CIELAB space, then manually mapping the CSS3 colors to CSS2.1 colors.

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

[1] 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.

the table").

2.4.2 Scene Graph Generation

Eight different types of semantic spatial relations are heuristically calculated based on the yawed object bounding boxes to generate a scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of relations. Each vertex in the graph represents an object along with its attributes (name, colors, size...). A binary spatial relationship (e.g. above, below, closest...) is represented as a directed edge between two object vertices, while a ternary relationship (e.g. between) is an edge between a target object vertex and a pair of anchor object vertices. Relations are generated exhaustively for every pair or triplet of objects within a region, then filtered based on the semantic classes involved.

Table 2.1 defines the types of spatial relations used. The spatial relationships in this dataset do not depend on an observer's viewpoint—they are, by design, view-independent, or *allocentric*. We relegate generating egocentric relationships to a future work.

2.4.3 Language Generation

Referential statements are 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:

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. In other words, the vertex-edge-vertex triplet representing the referential statement in the spatial scene graph must be unique in terms of vertex and edge attributes.
3. **Minimal**: Following Grice's maxim of manner [44], statements use the least possible descriptors to most clearly disambiguate the target object (Fig. 2.5).

Additionally, the dataset includes "imperfect" referential statements describing

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

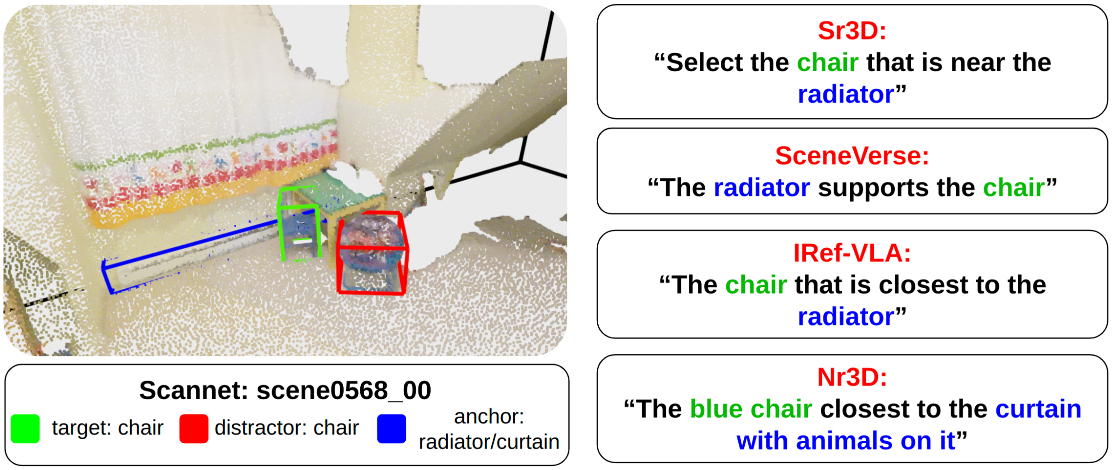

Sr3D:
“Select the chair that is near the radiator”
SceneVerse:
“The radiator supports the chair”
IRef-VLA:
“The chair that is closest to the radiator”
Nr3D:
“The blue chair closest to the curtain with animals on it”
Scannet: scene0568_00
🟩 target: chair
🟥 distractor: chair
🟦 anchor: radiator/curtain

Figure 2.5: A comparison between heuristically generated statements describing a binary spatial relation from Sr3D, Nr3D [1], SceneVerse [2], and IRef-VLA. Both chairs are close to the radiator, so using the superlative relation ”closest” is the clearest way to disambiguate.

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.

2.5 Task Formulation: Referential Grounding with Imperfect References

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.

We differentiate this task from the original referential object-grounding task in

ReferIt3D [1] and embodied tasks such as ObjectNav [45], ObjectGoal [46], and AreaGoal [46]. Compared to the standalone object referential grounding task in benchmarks such as ReferIt3D [1] and ScanRefer [26], 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) [45], 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.

2.5.1 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.

To quantitatively assess the quality of retrieved object alternatives, we use a heuristic scoring system. 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. 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

Table 2.2: Dataset generalizability on various baseline models

Method	Train Checkpoint	Test Set			
		Sr3D	Nr3D	IRef-VLA	
		Overall	Overall	ScanNet	Full
MVT [24]	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 [25]	Baseline (Sr3D, reported)	76.4%	-	-	-
	Baseline (Sr3D, reproduced)	75.7%	46%	39.2%	24.8%
	SceneVerse *(0-shot text)* [2]	-	43.1%	-	-
	IRef-VLA-ScanNet	62.4%	41.8%	63.7%	32.3%
	IRef-VLA-Full	65.8%	44.9%	70.8%	60.6%

weight on the i-th aspect of S:

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

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.

2.6 Experiments

We evaluate our benchmark on both the original referential grounding task and our extended task. First, we compare our data with ReferIt3D [1] using two SOTA 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 Grounding Baselines

Experimental Setup

To evaluate the effects of scaling up the amount of referential language and the number of real-world scenes on the referential grounding task, we train two open-source supervised referential grounding baseline models on our data: MVT [24] and 3D-VisTA [25]. 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 [1] 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 [2], 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 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 [1] test split. The results can be found in table 2.3 and implementation details are further described below.

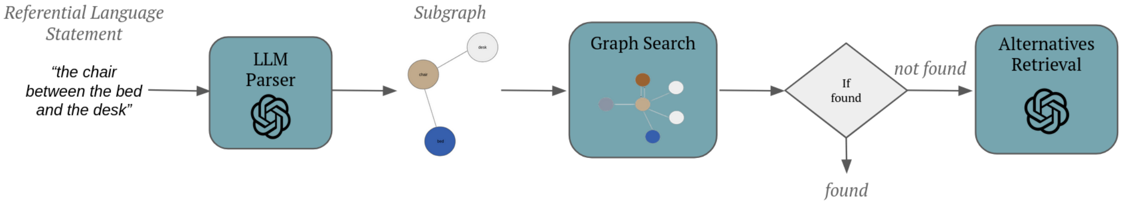


Figure 2.6: Pipeline for graph-search and alternative generation baseline.

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[2], 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 Fig. 2.6.

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

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 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 Further Discussion of Results

2.7.1 Referential Grounding as a Graph Search Problem

When trained on IRef-VLA, the relatively good zero-shot transfer performance of baseline referential grounding models on Nr3D as shown in table 2.2 implies that many referential utterances that use spatial relations can be grounded through heuristic search methods given a list of objects in the scene. Since a scene graph containing spatial relationships can be generated with heuristics that only require object names and bounding boxes, then referential statements denoting simple binary or ternary spatial relations can be generated by picking a unique vertex-edge-vertex triplet in the graph, an end-to-end referential grounding model effectively learns the "inverse" of this process. Evidently, on many types of object-relational referential statements, 3D referential grounding can be approached alternatively by decomposing it into the following subtasks:

1. Obtaining object bounding box proposals using a 3D detection/instance segmentation model.
2. Creating a spatial scene graph, either heuristically like in this work, or using a learned approach [28, 29, 30].
3. Parsing the referential statement into a graph search query, and searching for the object in the scene graph.

The performance of our LLM-based graph search baseline on grounding object existence for the interactive referential grounding task on IRef-VLA demonstrates the viability of LLMs in referential grounding given the preceding decomposition of the task and an appropriate language-based scene representation. Additionally, more complicated object-relational referential statements like ”the computer under the desk with the widescreen monitor on top of it" can be represented as larger subgraphs of the spatial scene graph, and can easily be grounded using this approach, indicating its versatility.

2.7.2 Leveraging Foundation Models

Even with the introduction of IRef-VLA, a considerable gap still exists between the amount of 3D scene language-aligned data and 2D image language-aligned data, and hence a gap between the performance of 3D VLMs and 2D VLMs. Leveraging foundation models either in dataset generation or in model development is therefore imperative to improve the performance of referential grounding-based VLN systems.

LLMs can be used either for augmenting referential statements with synonymous statements, as a submodule in a decoupled grounding system as described in section 2.7.1, or as a backbone in a 3D VLM [47]. Additionally, objects are sometimes described using their textures rather than their colors, which is a purely 2D visual feature that cannot be obtained from color clustering. Using a captioning model on 2D object crops, like in [2], is therefore conducive towards providing richer attributes in the dataset. We relegate that to future work.

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

2.9 Limitations and Future Work

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, along with leveraging captioning models to obtain object attributes, using LLMs to augment statements, generating more complex statements like ”the computer under the desk with the widescreen monitor on it” using spatial relation subgraphs of higher depth, and 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.

2.10 Acknowledgments

This chapter was authored jointly with Haochen Zhang, and contributed to by Pujith Kachana, Ji Zhang, and Wenshan Wang.

Chapter 3

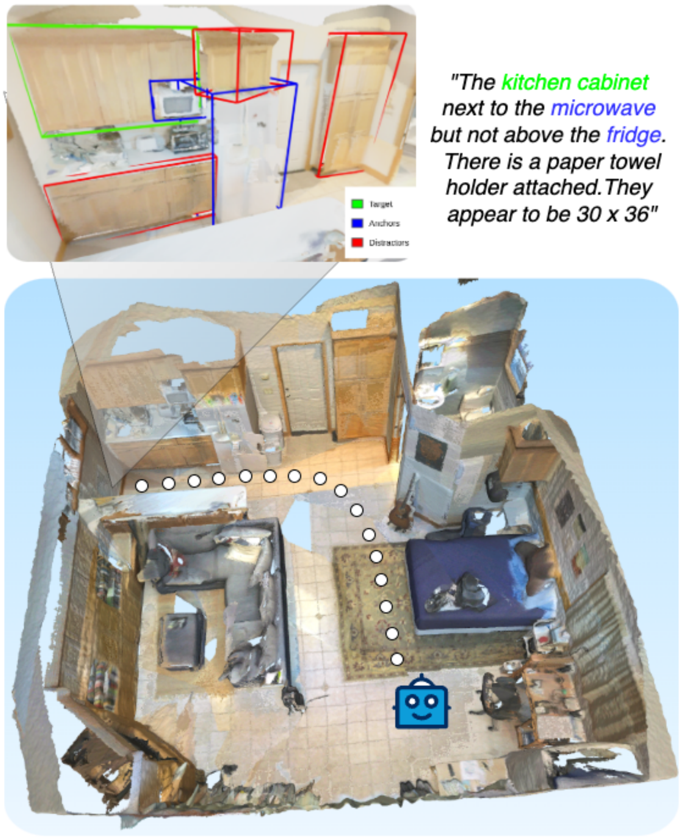
A Modular Approach to Object-centric Navigation Leveraging Foundation Models

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 available at `https://github.com/nzantout/SORT3D`.

3.2 Introduction



"The kitchen cabinet next to the microwave but not above the fridge. There is a paper towel holder attached.They appear to be 30 x 36"

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 spatial relations—the core challenge of **3D referen-**

tial grounding—remains difficult despite being an intuitive task for humans. For example, understanding statements such as "*the chair closest to the closet door*", is a task trivial for humans [1] 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 spaces often contain a large number of objects from fine-grained categories [20], with distributions that vary widely across 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 [1].

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 [24, 25, 33], 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 both in a zero-shot fashion [48, 49, 50, 51, 52, 53] and with pretraining on 3D referential datasets [54]. While some achieve strong results on benchmark datasets [48], 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, semantically labeled reconstructed meshes of 3D scenes [27, 32] 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 [45]. 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 large language models (LLMs) using chain-of-thought prompting [55] 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 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 [1] and IRef-VLA [9], 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, dynamic environments and run online alongside real-time perception and autonomy stacks.

3.3 Related Work

3.3.1 Referential Object Grounding Datasets

Using referring expressions to identify a target object, also defined as the 3D referential grounding task, has been explored in a number of datasets such as ReferIt3D [1], ScanRefer [26], SceneVerse [2], and IRef-VLA [9] in the 3D domain. Of these datasets, only the Nr3D subset of ReferIt3D and ScanRefer contain human-generated natural language utterances, while the remaining datasets generate synthetic referential statements using template-based generation and/or LLMs to rephrase. Synthetic datasets only employ basic spatial relations like ”lamp on the desk” and ”computer next to the office chair" 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 attributes. 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.

3.3.2 3D Referential Object Grounding Baselines

Grounding object references to 3D scenes has been explored 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 [33], MVT [24], ViL3DRel [56], 3D-VisTA [25], and GPS trained on SceneVerse [2], 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 [49], and LLMs and VLMs in a zero-shot manner to effectively reason about spatial relations like ZSVG3D [57], VLM-Grounder [52], CSVG [50], and Transcrib3D [48]. 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 after skipping statements that do not contain the target object or were answered incorrectly by human annotators. To achieve this, Transcrib3D relies on a simplified representation of the scene containing a list of objects along with their names, positions, and dominant colors, along with 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 [48, 50, 52, 57]. We aim to address this gap with our method by combining structured heuristics and vision foundation models with the sequential reasoning capabilities of LLMs.

3.3.3 Grounding Large Language Models in 3D

Recent efforts have also aimed to develop 3D foundation models capable of handling general 3D tasks. These models build upon pretrained LLMs, fine-tuning them on tokenized 3D data. By leveraging the strong reasoning capabilities of LLMs, these approaches bridge the gap between natural language understanding and 3D perception. One of the pioneering works in this space, 3D-LLM [47], adopts a 2D VLM as its backbone. To extract 3D features, it projects a given 3D input into multiple 2D images, obtains dense 2D features using ConceptFusion [58], and reconstructs 3D representations from these 2D embeddings. [54] improves upon the work to handle grounding to different viewpoints and generalizes to various 3D tasks with large-scale language-scene pretraining. Another significant contribution in this domain is LEO [59], which aims to create a unified 3D foundation model. LEO utilizes a pretrained LLM and applies LoRA [60] fine-tuning to adapt it for 3D-centric tasks. A key enabling factor for these methods is their reliance on pretrained LLMs and dense 2D data, demonstrating 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.

3.3.4 Vision-and-Language Navigation with LLMs

A number of recent works have also leveraged the sequential reasoning capabilities of LLMs for the trajectory-following subtask within vision-and-language navigation (VLN). NavGPT [61] and NavCoT [62] 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 a full internal representation of 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.

3.4 SORT3D: Spatial Object-centric Reasoning Toolbox for Zero-Shot 3D Grounding Using Large Language Models

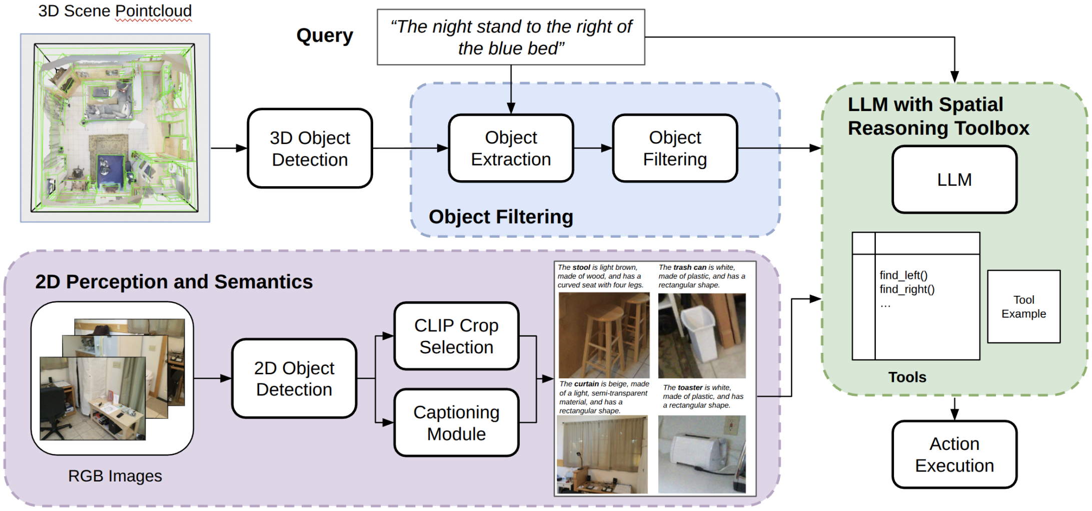

3D Scene Pointcloud
Query
“The night stand to the right of the blue bed”
3D Object Detection
Object Extraction
Object Filtering
Object Filtering
LLM with Spatial Reasoning Toolbox
LLM
find_left()
find_right()
...
Tool Example
Tools
2D Perception and Semantics
RGB Images
2D Object Detection
CLIP Crop Selection
Captioning Module
The stool is light brown, made of wood, and has a curved seat with four legs.
The trash can is white, made of plastic, and has a rectangular shape.
The curtain is beige, made of a light, semi-transparent material, and has a rectangular shape.
The toaster is white, made of plastic, and has a rectangular shape.
Action Execution

Figure 3.2: The full system diagram for the SORT3D framework

In a human-agent collaborative setting, humans commonly utter single references to objects in the scene, 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 a fundamental task for the deployment of a practical VLN system. 3D referential grounding additionally acts as a precursor to important tasks for a collaborative agent, including object-goal navigation, multi-action instruction following like ”grab the red mug from the living room table, then put it in the dishwasher", 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:

1. An instance-level semantic mapping system to obtain 3D bounding boxes for real-world deployment.
2. A captioning pipeline to incorporate rich 2D semantic information for each object.
3. Filtering for relevant objects based on the input utterance.
4. LLM-based spatial reasoning augmented with a spatial reasoning toolbox to resolve the target object, followed by code generation for executing a downstream action during real-world navigation.

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. Our mobile robot perception setup for real-world experiments consists of a 360 camera and a 3D LiDAR (section 3.5.3 contains further hardware details). We initially perform object detection in 2D using the open-vocabulary object detector Grounding DINO [63], and obtain object instance masks using SAM 2 [64]. We then project the LiDAR point clouds—which are registered using a modified version of LOAM [65]—onto the now semantically annotated 360 image and associate each point with its corresponding pixel semantic ID. As the robot moves and produces new observations, we track object instances across frames using ByteTrack [66], utilizing the object's 3D position to improve tracking robustness.

Afterwards, we associate new LiDAR pointcloud observations with tracked object IDs, and postprocess the instance point clouds through clustering, removing outliers, and merging improperly tracked point cloud instances corresponding to the same objects using 3D proximity priors. We finally obtain axis-aligned 3D object bounding boxes to be sent to the rest of the pipeline. For our results on the ReferIt3D benchmarks, we simply use ground truth bounding boxes and instance segmentations.

We note that, due to the usage of 2D vision foundation models trained on billion-scale image data in detection and segmentation, our semantic mapping module generalizes to any environment, and allows zero-shot customization of detected objects by editing the set of labels used to prompt Grounding DINO. This leads to robust performance in real-world experiments. For this work, the semantic mapping module assumes that the objects are static. We relegate dynamic object mapping to an soon-upcoming version of the module.

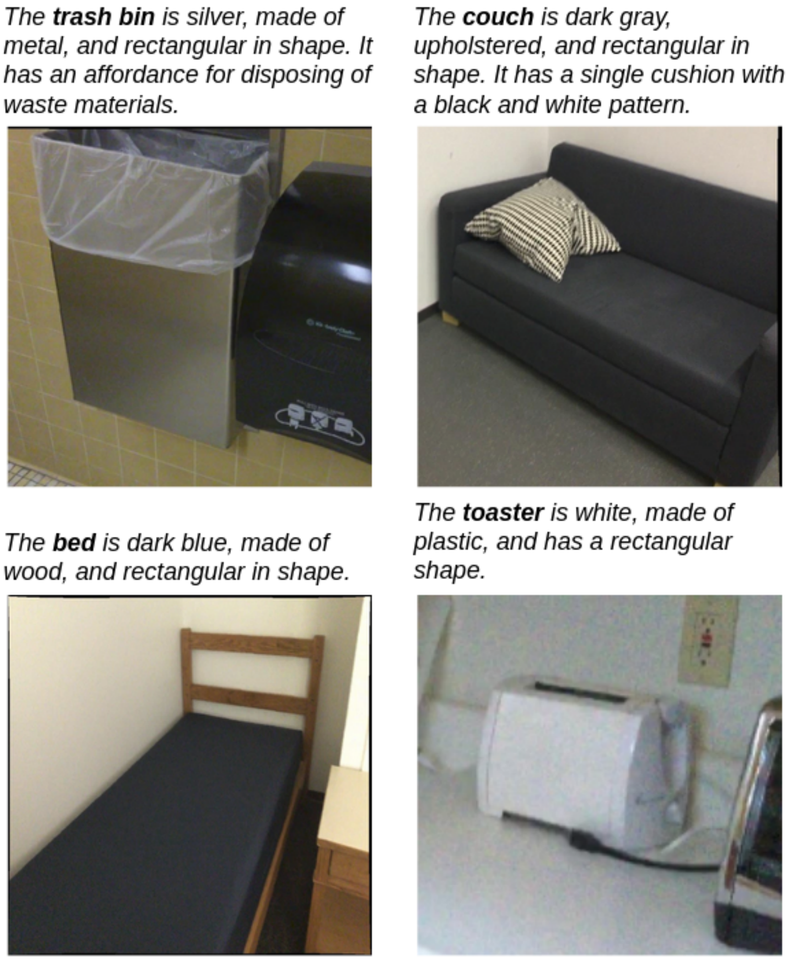
3.4.2 Enhancing Object Perception with 2D Captions

Accurately understanding the attributes and affordances of 3D objects is an essential first step for referential grounding. Existing works [48], [24] 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 [25]. 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 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 as the agent navigates around the scene and captures multiple views of each object. 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 for the benchmark 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 [68] 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 a query 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 name> is <color>, <material>, <shape>'". We release all object crops and captions as a supplement to the ScanNet [27] 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 [48, 49], we implement an LLM-based filtering module consisting of two LLM queries. In the first query, the LLM is given the input query—for example, "The night stand to the right of the bed"—and prompted to extract the objects in the sentence, returning a list of nouns and their modifiers—in this case, the list `[nightstand, bed]`. In the second query, the LLM is given the list of object IDs and names provided by the perception module and the extracted list of nouns, and prompted to return the list of relevant IDs within the object list—for example, `[[2, nightstand, ...], [3, bedside table, ...], [4, bed, ...]]`. We use Mistral Large 2 [69] for these steps, filtering objects based on their text descriptions to best leverage the ability of LLMs to process textual information.



*The **trash bin** is silver, made of metal, and rectangular in shape. It has an affordance for disposing of waste materials.*

*The **couch** is dark gray, upholstered, and rectangular in shape. It has a single cushion with a black and white pattern.*

*The **bed** is dark blue, made of wood, and rectangular in shape.*

*The **toaster** is white, made of plastic, and has a rectangular shape.*

Figure 3.3: Generated image crops and corresponding caption

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. 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. In our initial experiments, we only provided this list of objects to an LLM, and given a language query referring to an object in the scene, prompted the LLM utilizing chain-of-thought reasoning and a set of in-context examples to ground the object within the scene. This resulted in poor performance, due to the inherent limitations in spatial and mathematical reasoning within an LLM—for example, when providing a query with a view-dependent spatial relationship like ”the nightstand to the left of the bed”, the LLM picks the one with the smallest x value. Additionally, queries referring to edges of objects like ”the monitor at the end of the desk” almost always fail, as we only provide object centers to the LLM. Adding bounding boxes to the object representations leads to worse performance and further confusion by the LLM, requiring a different approach.

Given the limitations in the ability of LLMs to perform spatial reasoning under our scene representation, 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 elementary spatial relations. The arguments and return values of the search functions are described in tables 3.1 and 3.2. All of the search functions have access to the full output of the perception module, containing object names, captions, and full 3D bounding boxes. For referential statements employing view-dependent/egocentric relationships (table 3.2, we currently tackle the case where no specific observer viewpoint is given, which is a common human referential utterance as demonstrated in Nr3D. We assume that the relationship is unambiguous from any viewing direction where an observer may feasibly stand and see the objects used in the reference (e.g. ”the cabinet to the left of the water cooler”, where both objects are against a wall), and hence use the point in traversable space nearest to the object pair as a viewpoint. The target is then to the left of the anchor if the cross product—equivalent to the sine of the signed angle—between the xy-planar vector joining the observer and the anchor and the vector joining the observer and the target is positive, and to the right of the anchor if negative. For `order_smallest_to_largest`, we use the area of the largest face to represent size, which is more intuitive for flat objects. We refer readers to our accompanying repository for further implementation details regarding the spatial search functions.

The perceptual input, along with the function implementations, are all abstracted away from the LLM, which is only aware of the function arguments and return values described in tables 3.1 and 3.2, along with usage examples for each function. The LLM is prompted with a single in-context example to use these functions as much as possible in reasoning, and tasked with decomposing a referential statement into one or more search calls, then choosing the correct ID from the returned IDs by performing a set intersection. For example, given the query "Find the computer near the desk with a printer on it", the LLM would first call `find_below(desk, printer)` returning `[2]`, then `find_near(computer, 2)` returning `[3, 4]`, finally picking the computer with ID 3. In this formulation, instead of performing spatial reasoning given an arbitrary representation, the LLM is only tasked to translate language into repeated function calls employing sequential logic—a domain LLMs excel that, due to their training on large code corpora. Additionally, by abstracting away the implementation of spatial search tools from the LLM, search functions could be made arbitrarily complex, potentially using full object point clouds in computing spatial relations, or even using learned modules on a grounding task simplified by the LLM into a classification task (e.g. a module that learns how to classify a "hanging on" relation purely given vision input). We open the field for future contributions into improving the spatial reasoning toolbox.

3.4.5 Parsing into Actions

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 for sequential navigation. `go_near`, supplied with a single object ID, places a waypoint at the closest point in the scene's free space to the object's center traversable by the robot. `go_between`, supplied with two object IDs, places a waypoint at the point in free space closest to the midpoint

Function	Arguments	Usage

`find_near`	`target_name: str,`
	`anchor_id: int`
		Returns IDs of all objects with name `target_name` ordered from nearest to object with ID `anchor_id` to furthest.

`find_between`	`target_name: str,`
	`first_anchor_name: str,`
	`second_anchor_name: str`
		Returns IDs of all objects with name `target_name` that are either horizontally or vertically inbetween objects with name `first_anchor_name: str` and objects with name `second_anchor_name: str`.

`find_above`	`target_name: str,`
	`anchor_name: str`
		Returns IDs of all objects with name `target_name` that are above any objects with name `anchor_name`.

`find_below`	`target_name: str,`
	`anchor_name: str`
		Returns IDs of all objects with name `target_name` that are below/underneath any objects with name `anchor_name`.

`order_bottom_to_top`	`target_name: str`	Returns IDs of all objects with name `target_name` ordered from bottom to top.
`order_smallest_to_largest`	`target_name: str`	Returns IDs of all objects with name `target_name` ordered from smallest to largest.
`find_objects_near_room_corner`	`target_name: str`	Returns IDs of all objects with name `target_name` within $1m$ of the corner of the room.

Table 3.1: Heuristic search functions comprising the spatial toolbox for view-independent/allocentric spatial relationships.

Function	Arguments	Usage
`find_left`	`target_name: str`, `anchor_id: int`	Returns IDs of all objects with name `target_name` that are to the left of object with ID `anchor_id`.
`find_right`	`target_name: str`, `anchor_id: int`	Returns IDs of all objects with name `target_name` that are to the left of object with ID `anchor_id`.
`order_left_to_right`	`target_name: str`	Returns IDs of all objects with name `target_name` ordered from left to right.

Table 3.2: Heuristic search functions comprising the spatial toolbox for view-dependent/egocentric spatial relationships with unambiguous viewing anchors.

of the objects' centers. The waypoints are sent to FAR Planner [70], which plans and executes an obstacle-avoidant path to each point.

3.5 Experiments

We evaluate our method on two 3D object-referential datasets, ReferIt3D [1] and IRef-VLA [9]. Both datasets consist of utterances describing a target object in a ScanNet [27] 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 as "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 different 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 [1], 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 [48] which, to the best of our knowledge, is the best performing model on ReferIt3D to date. We note, however, that their evaluation skips over statements in the test set that either do not explicitly mention the target object or human annotators were unable to guess correctly during the dataset generation. We provide results for our method on both cases, with these statements included and skipped, where "+skip" indicates the exclusion of such statements and subsequent simplification of the problem. We run the experiments using both Mistral Large 2 and GPT-4o as the LLM in the spatial reasoning module. For a fair comparison, we run Transcrib3D with the same two LLMs we use on the same test splits [2]. 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.

The grounding accuracy on ReferIt3D is shown in Tables 3.3 and 3.4, and accuracy on IRef-VLA is shown in Table 3.5. We see that our method achieves higher overall 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 on some subsets up to 7.2%.

While supervised baselines such as ViL3DRel [56], 3D-VisTA [25], and SceneVerse [2] report slightly higher overall grounding accuracies, these methods are explicitly

[1] 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

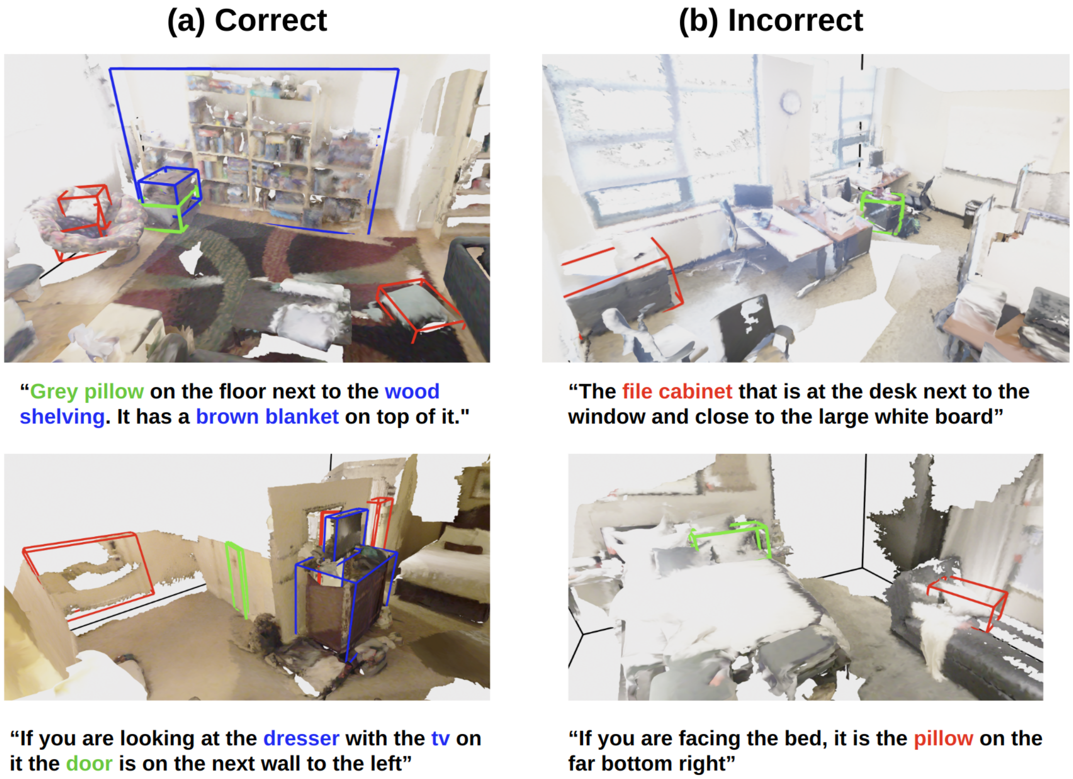
[2] the GPT-4 model used in their work is now a legacy model. We run their method with GPT-4o instead.

trained on ReferIt3D data. Similarly, while Transcrib3D [48] reports higher accuracies, it relies on a) removing statements that annotators incorrectly answered and b) guiding principles 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 a single example of how to use the spatial reasoning toolbox, which does not have to be from a particular dataset, and we employ no dataset-specific training or fine-tuning. By leveraging foundation models to obtain object semantic attributes and mapping spatial reasoning into sequential reasoning, our spatial reasoning toolbox approach achieves performance comparable to SOTA supervised methods on view-dependent statements in Nr3D, surpasses zero-shot methods on overall grounding accuracy and view-dependent statement accuracy without skipping incorrectly annotated statements, and achieves performance on par with Transcrib3D when skipping incorrect annotations.

On Sr3D, SORT3D surpasses SOTA supervised training methods on overall grounding accuracy, achieves close overall performance to Transcrib3D, and surpasses Transcrib3D 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), and are additionally able to explain model failure points by analyzing its chain of thought (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, object 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 via self-correction on the dataset's hard statements [48], 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.



"Grey pillow on the floor next to the wood shelving. It has a brown blanket on top of it."

"The file cabinet that is at the desk next to the window and close to the large white board"

"If you are looking at the dresser with the tv on it the door is on the next wall to the left"

"If you are facing the bed, it is the pillow on the far bottom right"

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 3.3: 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* [49]	62.7	62.0	-
ViL3DRel* [56]	64.4	62.0	64.5
3D-VisTA* [25]	64.2	61.5	65.1
SceneVerse-GPS* [2]	64.9	56.9	67.9
Zero-Shot Methods			
ZSVG3D* [57]	39.0	36.8	40.0
VLM-Grounder* [52]	48.0	45.8	49.4
CSVG* [50]	59.2	53.0	62.5
Transcrib3D* [48] (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 3.4: 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* [56]	72.8	63.8	73.2
3D-VisTA* [25]	76.4	58.9	77.3
SceneVerse-GPS* [2]	77.5	62.8	78.2
Zero-Shot Methods			
Transcrib3D* [48] (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 3.5: 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**

3.5.2 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 [48] 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 3.6. 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.

3.5.3 Real-world Validation on a Mobile Robot

To validate SORT3D's pipeline in the real-world, we implement the full system on two autonomous mobile robots:

1. A differential-drive ground vehicle with a wheelchair base equipped with a 360-degree camera, a Velodyne LiDAR, an onboard Intel NUC for low level autonomy, and a desktop RTX 4090 with 24GB of VRAM.
2. A mecanum-wheeled mobile platform with a 360 camera, a Livox Mid-360

Table 3.6: 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%)

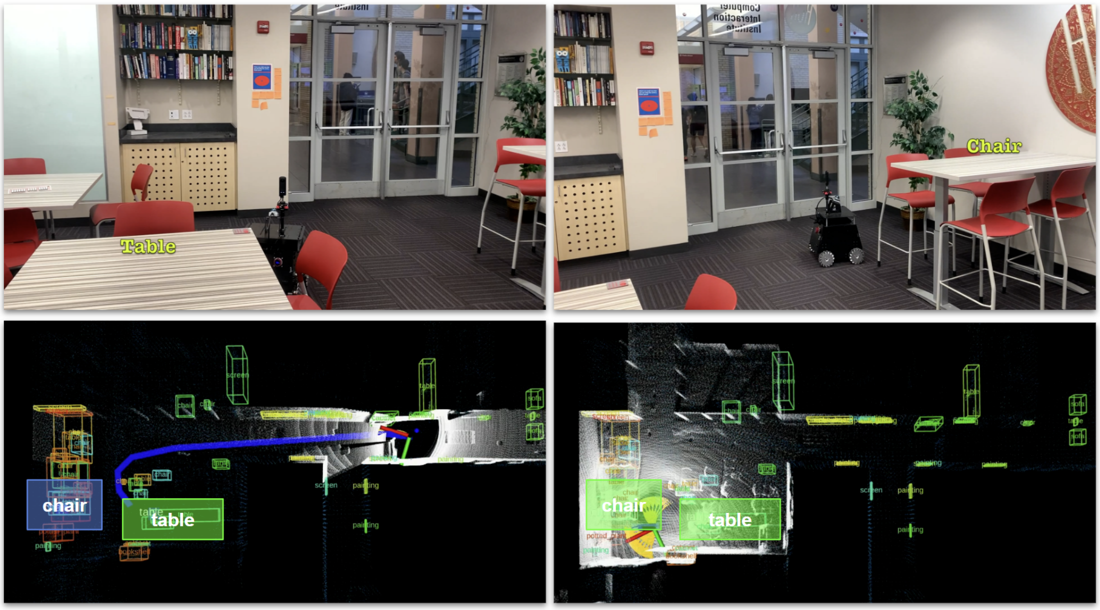
LiDAR, an onboard NUC, and a laptop RTX 4090 with 16GB of VRAM.

We validate our system in three previously unseen indoor environments—a kitchen area with a portion of the surrounding office (figure 3.7, a student lounge (figure 3.6, and a university corridor (figures 3.5). We attempt three different types of navigational queries listed in table 3.7, which target the system's ability to ground statements that employ spatial references and object semantic attributes, along with its ability to infer which objects to navigate to from implicit statements like "fetch me my print". 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 figures 3.7, 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 3.7, 3.6, and 3.5 for some of the expressions listed in table 3.7.

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.

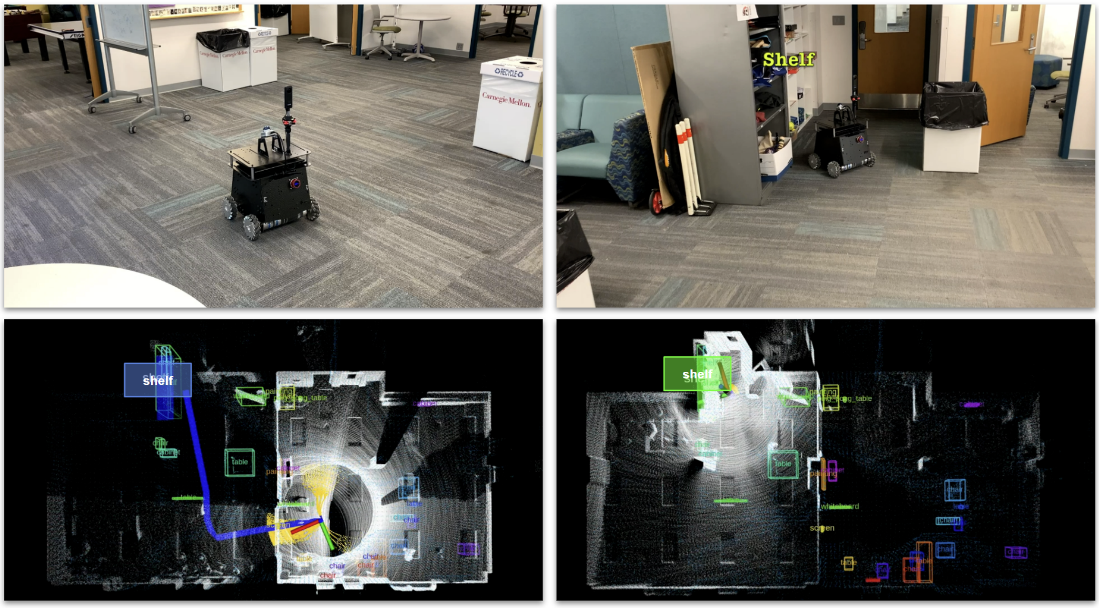
Platform	Environment	Reference Type	Statement
Mecanum	Student lounge	Spatial relation	Go to the painting next to the fridge.
		Spatial relation and semantic attribute	Go to the whiteboard next to the painting.
		Semantic attribute	I want to play a board game, fetch me one from the shelf.
	University corridor	Spatial relation	Go to the table next to the bookshelf, then to the chair next to the plant.
Wheelchair	Kitchen area	Spatial relation and semantic attribute	Go to the blue garbage bin next to the wet floor sign.
		Implicit statement	The plant is on fire, what do I do?
			I am sitting on the sofa, and my print is done. Fetch me my print.

Table 3.7: The list of statements used for real-world validation, categorized by the robotic platform they are tested on, the testing environment, and the type of referential intelligence being tested in the statement.



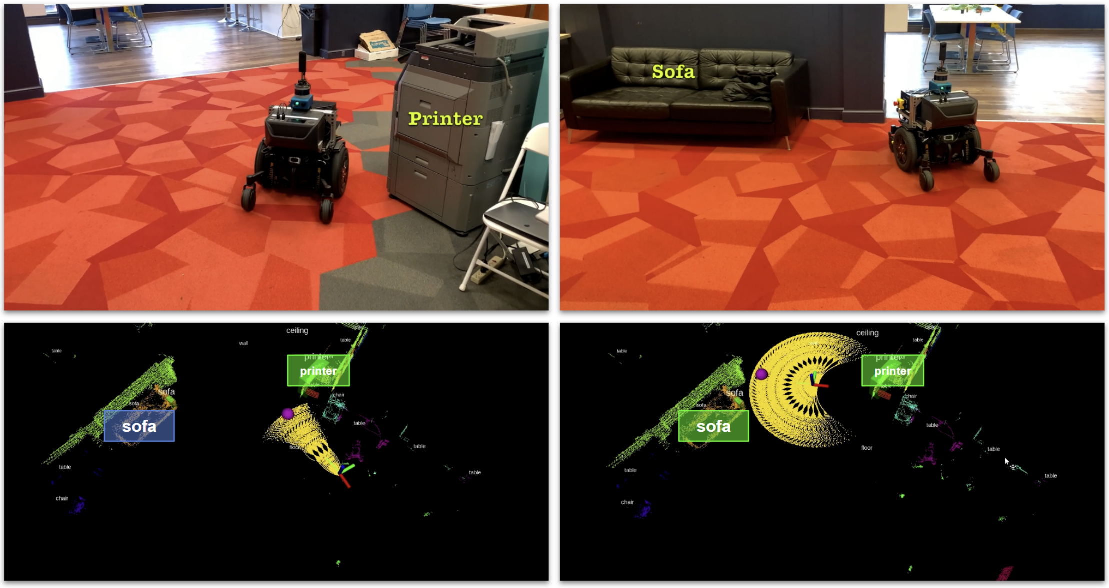
Go to the table next to the bookshelf, then to the chair next to the plant.

Figure 3.5: SORT3D real-world navigation on the Mecanum robot in the university corridor scene given the statement “Go to the table next to the bookshelf, then to the chair next to the plant.” The system successfully grounds both referential statements with the correct spatial relations and navigates to each object in sequence.



I want to play a board game, fetch me one from the shelf.

Figure 3.6: SORT3D real-world navigation on the Mecanum robot in the student lounge scene given the statement "I want to play a board game, fetch me one from the shelf." The system must successfully pick the shelf with the board game on it, which it can only do using the caption description of what is on each shelf, hence testing its grounding capabilities using semantic attributes. The system grounds the correct shelf and navigates towards it.



I am sitting on the sofa, and my print is done. Fetch me my print.

Figure 3.7: SORT3D real-world navigation on the Wheelchair robot in the kitchen scene given the statement “I am sitting on the sofa, and my print is done. Fetch me my print.” This statement tests the deductive reasoning of the model, implicitly requiring the robot to navigate to the printer to ”fetch” the print, then go to the sofa. The robot successfull performs this maneuver, demonstrating the LLM-powered deductive capabilities of SORT3D.

3.6 Conclusion

We introduce SORT3D, a robust, data-efficient, and deployable method for 3D referential grounding with complex spatial and semantic reasoning. Our framework effectively tackles 3D referential grounding by decomposing the task into four main stages: obtaining object instances through a semantic mapping module, generating object semantic descriptions through a VLM-based captioning module, filtering objects relevant to the query, and parsing the query into a series of searches using a heuristics-driven spatial reasoning toolbox to ground the object and output navigation commands. By leveraging strong visual semantic priors in 2D VLMs and language semantic priors in LLMs, SORT3D achieves benchmark results competitive with SOTA methods on view-dependent and attribute-based statements—while outperforming other fully zero-shot methods. Moreover, SORT3D demonstrates powerful generalization to completely unseen real-world environments, acting as a robust real-time indoor navigation system capable of understanding complex and implicit referential statements and navigating to referenced objects. Additionally, SORT3D is designed to be highly modular and easily customizable, serving as a springboard and catalyst for advancing the development of practical embodied AI systems designed for various environments.

3.7 Limitations and Future Work

We acknowledge that limitations exist within our approach. First, our current iteration of the spatial reasoning toolbox does not cover all possible elementary spatial relations. We currently do not include functions that handle the relations "in front of" and "behind" which sometimes refer to canonical object directions and other times depend on viewpoint, view-dependent relations that specify a viewpoint as in the statement "standing at the edge of the bed, the pillow on the top right in the stack" and differ with different viewpoints, and relations that require structuring and clustering objects, like "the third chair in the front row from the right" and "from the set of three desks, the desk on the right". We leave the implementation of these relations within the spatial reasoning toolbox to future work. We additionally open the field for learned search functions for spatial relations instead of hard-coded functions, especially for relations like "in front" and "between" that depend on fuzzy human semantics. Learning a search function can be cast as a binary classification on perceptual input, outputting a set of object IDs that satisfy a condition (without necessarily using language).

Additionally, our system does not explicitly handle references to rooms, such as ”the windows in the kitchen”. One approach to this is to create a hierarchically clustered object scene graph using heuristics or a learned approach, then search through the nodes of the scene graph in a top down manner before searching for spatial relations.

Regarding practical deployment, our system relies on internet access to call online APIs, which is a reasonable assumption for indoor environments like homes but may not hold for outdoor navigation scenarios. To address this limitation, the modular design of our pipeline allows for the LLMs to be easily replaced by smaller local models.

Finally, we 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.

3.8 Acknowledgments

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

Chapter 4

Conclusions and Future Directions

4.1 Summary of Contributions

This thesis explored the challenge of creating a practical Vision-and-Language Navigation (VLN) system through the lens of **3D referential grounding**—a fundamental task for robots to interpret human language commands in a collaborative setting, lying at the heart of embodied AI. 3D referential grounding requires a robust association between scene semantics and language semantics in a shared visuo-linguistic latent space, which has been achieved in 2D image space due to the abundance of text-image training data, but remains difficult in 3D space due to the lack of data and the increased complexity compared to 2D visual grounding. We approached this challenge from both end-to-end and modular perspectives to circumvent the limited amount of 3D data, aiming to strike a balance between generalizability, robustness, and deployability in real-world settings.

We first introduced **IRef-VLA**, a large-scale interactive benchmark for 3D referential grounding. By constructing view-independent spatial scene graphs and systematically generating referential utterances, IRef-VLA enables broad coverage across object relations, spatial configurations, and linguistic constructs. End-to-end referential grounding models trained on IRef-VLA demonstrated strong zero-shot transfer performance between IRef-VLA's synthetically generated references and human-generated references, and a graph search baseline on spatial scene graphs achieved a high accuracy on grounding object existence, revealing a simplification of the 3D referential grounding task enabling the decomposition of complex referential statements into search queries over a graphs of simple relations.

Based on this insight, we then proposed a complementary modular approach through **SORT3D**, a real-time, object-centric grounding system that decomposes the referential grounding task into four distinct stages: semantic mapping, attribute extraction via VLMs, query-based object filtering, and structured LLM-based reasoning using a spatial toolbox which outputs object IDs or navigation commands. SORT3D bridges powerful pre-trained models with symbolic reasoning, achieving strong generalization on IRef-VLA and ReferIt3D and demonstrating real-world success when deployed on physical robots in previously unseen environments.

Together, these systems establish a template for building embodied agents that can understand and follow grounded language instructions without needing exhaustive end-to-end training in every deployment context.

4.2 End-to-End vs. Modular Grounding: A Matter of Data

Throughout this work, we have observed that end-to-end referential grounding baselines suffer from poor zero-shot transfer performance, and while upscaling data offers improvements on human language benchmarks compared to previous approaches, performance is still poor relative to models trained directly on the benchmarks, and even poorer in unseen real-world environments. On the other hand, using a modular approach and allowing each submodule to leverage inductive biases suited to its function greatly improves zero-shot generalizability to new scenes. However, using a structured approach and only leveraging 2D and language priors leads to fundamental limitations in the model's capability. For instance, references to described regions of a wall or object become reliant on the captioning module, which generally does not capture all semantic attributes. This, in addition to the complexity of some of the spatial relations mentioned in the conclusion of chapter 3, necessitates either a redesign of the scene representation to include granularities smaller than objects, or the manual addition of a prohibitive amount of spatial search tools with finely tweaked parameters.

We can see that SORT3D and end-to-end trained to models lie on two ends of a spectrum, or a number line. Going towards the side of SORT3D means making the system more rigidly structured by manually separating modules and inducing biases, while going towards the side of end-to-end models shifts towards models that have more representational capacity. Ultimately, the location on this spectrum for any task depends solely on the amount of data available for this task. The more data available for a task, the more representational capacity a model needs for optimal performance, and vice versa with models needing more inductive biases for less data—a tradeoff that pervades deep learning.

Therefore, a key conclusion to be drawn from this thesis is the optimal placement on the spectrum for VLN, and what parts in SORT3D are to be changed to accommodate this placement. As LLMs are still excellent at decomposing a query into a set of simplified constituent queries and function calls, one way to shift towards the end-to-end model side of the spectrum is to use learned search functions instead of heuristic functions for certain relationships. These models may simply be object classification models, where object representations are placed into a perception backbone and output binary classifications for each object, without the use of language. Adjusting the scopes of the 3D learned submodules in this hybrid architecture therefore depends on data availability, and may change in the very near future as robot language data becomes more available. We leave the full study of the location of this model on the data spectrum to a future work.

4.3 Future Work

Other than leveraging foundation models and improving benchmark and real-world results, SORT3D's modularity allows the system to form the basis of further research into VLN outside the indoor environment, single-query setting. Some of these directions to be explored include, but are not limited to:

4.3.1 VLN in Outdoor Environments

Due to the use of open-vocabulary foundation models that are not trained on particular indoor datasets, SORT3D's architecture is easily adaptable towards outdoor environments and multiple agent form factors—including autonomous cars. Car autonomy can be augmented with language in situations when human interaction is needed, such as parking and picking people up.

4.3.2 Interactive Navigation

While we have explored imperfect statements requiring a second line of clarifying dialogue in IRef-VLA, a more thorough study of interactive navigation must include studying typical collaborative dialogue. Beyond asking for clarification, a truly adaptive dialogue system in a collaborative robot must continuously learn a user's preferences. In a sense, following Grice's maxims [44], an agent engaging in human-like dialogue in a servant-master dynamic with a human collaborator must effectively learn enough to eventually minimize dialogue to deem it successful.

4.4 Towards Embodied AGI

As the field of robotics converges with foundation models, building intelligent agents that understand and act in the world through human language remains a central milestone on the path toward Embodied Artificial General Intelligence (AGI). This thesis takes a leap in that direction by grounding language in structured 3D perception and demonstrating practical systems that reason, act, and adapt in human spaces. In the long term, the fusion of spatial reasoning, dialogue, and multimodal learning will be critical in shaping embodied agents that are not just reactive tools, but effective collaborative partners in human-centered environments.

Bibliography

[1] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. 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*, pp. 422–440, Springer, 2020. (document), 1.3, 2.3.1, 2.5, 2.5, 2.6, 2.6.1, 2.6.2, 3.2, 3.3.1, 3.5

[2] B. Jia, Y. Chen, H. Yu, Y. Wang, X. Niu, T. Liu, Q. Li, and S. Huang, “Sceneverse: Scaling 3d vision-language learning for grounded scene understanding,” *arXiv preprint arXiv:2401.09340*, 2024. (document), 2.3.1, 2.3.4, 2.4.1, 2.5, 2.2, ii, 2.7.2, 3.3.1, 3.3.2, 3.5.1, 3.3, 3.4

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, “Openvla: An open-source vision-language-action model,” 2024. 1.1, 2.2

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” in *arXiv preprint arXiv:2307.15818*, 2023. 1.1, 2.2

[5] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. 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. 1.1, 2.3.3

[6] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, “Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding,”

arXiv preprint arXiv:2010.07954, 2020. 1.1, 2.3.3

[7] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra, R. Mottaghi, J. Malik, and D. S. Chaplot, "Goat: Go to any thing," 2023. 1.1

[8] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "Laion-5b: An open large-scale dataset for training next generation image-text models," 2022. 1.2

[9] H. Zhang, N. Zantout, P. Kachana, J. Zhang, and W. Wang, "Iref-vla: A benchmark for interactive referential grounding with imperfect language in 3d scenes," 2025. 1.3, 3.2, 3.3.1, 3.5

[10] N. Zantout, H. Zhang, P. Kachana, J. Qiu, J. Zhang, and W. Wang, "Sort3d: Spatial object-centric reasoning toolbox for zero-shot 3d grounding using large language models," 2025. 1.3

[11] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023. 2.2

[12] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023. 2.2

[13] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023. 2.2

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021. 2.2

[15] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*, pp. 8821–8831, Pmlr, 2021. 2.2

[16] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024. 2.2

[17] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015. 2.2

[18] A. Karpathy, A. Joulin, and L. F. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," *Advances in neural information processing*

systems, vol. 27, 2014. 2.2

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

[20] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. 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, 2.4, 3.2

[21] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, pp. 32–73, 2017. 2.2

[22] S. Changpinyo, P. Sharma, N. Ding, and R. 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*, pp. 3558–3568, 2021. 2.2

[23] Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang, Z. Xie, Y. Wu, K. Hu, J. Wang, Y. Sun, Y. Li, Y. Piao, K. Guan, A. Liu, X. Xie, Y. You, K. Dong, X. Yu, H. Zhang, L. Zhao, Y. Wang, and C. Ruan, "Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding," 2024. 2.2

[24] S. Huang, Y. Chen, J. Jia, and L. Wang, "Multi-view transformer for 3d visual grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15524–15533, 2022. 2.2, 2.3.4, 2.2, 2.6.1, 3.2, 3.3.2, 3.4.2

[25] Z. Zhu, X. Ma, Y. Chen, Z. Deng, S. Huang, and Q. Li, "3d-vista: Pre-trained transformer for 3d vision and text alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2911–2921, 2023. 2.2, 2.3.4, 2.2, 2.6.1, 3.2, 3.3.2, 3.4.2, 3.5.1, 3.3, 3.4

[26] D. Z. Chen, A. X. Chang, and M. Nießner, "Scanrefer: 3d object localization in rgb-d scans using natural language," in *European conference on computer vision*, pp. 202–221, Springer, 2020. 2.3.1, 2.3.4, 2.5, 3.3.1

[27] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017. 2.3.1, 2.4, 2.4.1, 3.2, 3.4.2, 3.5

[28] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3d semantic scene graphs from 3d indoor reconstructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3961–3970, 2020. 2.3.2,

2

[29] N. Hughes, Y. Chang, and L. Carlone, “Hydra: A real-time spatial perception system for 3d scene graph construction and optimization,” *arXiv preprint arXiv:2201.13360*, 2022. 2.3.2, 2

[30] A. Werby, C. Huang, M. Büchner, A. Valada, and W. 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. 2.3.2, 2

[31] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, “Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning,” *arXiv preprint arXiv:2309.16650*, 2023. 2.3.2

[32] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3d: Learning from RGB-D data in indoor environments,” *International Conference on 3D Vision (3DV)*, 2017. 2.3.3, 2.4, 3.2

[33] A. Jain, N. Gkanatsios, I. Mediratta, and K. Fragkiadaki, “Bottom up top down detection transformers for language grounding in images and point clouds,” in *European Conference on Computer Vision*, pp. 417–433, Springer, 2022. 2.3.4, 3.2, 3.3.2

[34] P. Pramanick, C. Sarkar, S. Banerjee, and B. Bhowmick, “Talk-to-resolve: Combining scene understanding and spatial dialogue to resolve granular task ambiguity for a collocated robot,” *Robotics and Autonomous Systems*, vol. 155, p. 104183, 2022. 2.3.5

[35] M. Shridhar and D. Hsu, “Interactive visual grounding of referring expressions for human-robot interaction,” *arXiv preprint arXiv:1806.03831*, 2018. 2.3.5

[36] H. Zhang, Y. Lu, C. Yu, D. Hsu, X. Lan, and N. Zheng, “Invigorate: Interactive visual grounding and grasping in clutter,” *arXiv preprint arXiv:2108.11092*, 2021. 2.3.5

[37] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur, “Teach: Task-driven embodied agents that chat,” 2021. 2.3.5

[38] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, “Correcting robot plans with natural language feedback,” 2022. 2.3.5

[39] D. Shah, B. Osinski, B. Ichter, and S. Levine, “Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action,” 2022. 2.3.5

[40] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Niessner, "Rio: 3d object instance re-localization in changing indoor environments," in *Proceedings IEEE International Conference on Computer Vision (ICCV)*, 2019. 2.4

[41] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. 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. 2.4

[42] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from rgb-d images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 564–571, 2013. 2.4.1

[43] N. Silberman, D. Hoiem, P. Kohli, and R. 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*, pp. 746–760, Springer, 2012. 2.4.1

[44] R. E. Grandy and R. Warner, "Paul Grice," in *The Stanford Encyclopedia of Philosophy* (E. N. Zalta and U. Nodelman, eds.), Metaphysics Research Lab, Stanford University, Fall 2023 ed., 2023. 3, 4.3.2

[45] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020. 2.5, 3.2

[46] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018. 2.5

[47] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, "3d-llm: Injecting the 3d world into large language models," 2023. 2.7.2, 3.3.3

[48] J. Fang, X. Tan, S. Lin, I. Vasiljevic, V. Guizilini, H. Mei, R. Ambrus, G. Shakhnarovich, and M. R. Walter, "Transcrib3d: 3d referring expression resolution through large language models," *arXiv preprint arXiv:2404.19221*, 2024. 3.2, 3.3.2, 3.4.2, 3.4.3, 3.5.1, 3.3, 3.4, 3.5.2

[49] J. Hsu, J. Mao, and J. Wu, "Ns3d: Neuro-symbolic grounding of 3d objects and relations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2614–2623, 2023. 3.2, 3.3.2, 3.4.3, 3.3

[50] Q. Yuan, J. Zhang, K. Li, and R. Stiefelhagen, "Solving zero-shot 3d visual grounding as constraint satisfaction problems," *arXiv preprint arXiv:2411.14594*, 2024. 3.2, 3.3.2, 3.3

[51] J. Yang, X. Chen, S. Qian, N. Madaan, M. Iyengar, D. F. Fouhey, and J. 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)*, pp. 7694–7701, IEEE, 2024. 3.2

[52] R. Xu, Z. Huang, T. Wang, Y. Chen, J. Pang, and D. Lin, "Vlm-grounder: A vlm agent for zero-shot 3d visual grounding," *arXiv preprint arXiv:2410.13860*, 2024. 3.2, 3.3.2, 3.3

[53] R. Li, S. Li, L. Kong, X. Yang, and J. Liang, "Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding," *arXiv preprint arXiv:2412.04383*, 2024. 3.2

[54] Y. Chen, S. Yang, H. Huang, T. Wang, R. Xu, R. Lyu, D. Lin, and J. Pang, "Grounded 3d-llm with referent tokens," *arXiv preprint arXiv:2405.10370*, 2024. 3.2, 3.3.3

[55] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24824–24837, 2022. 3.2

[56] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Language conditioned spatial relation reasoning for 3d object grounding," *Advances in neural information processing systems*, vol. 35, pp. 20522–20535, 2022. 3.3.2, 3.5.1, 3.3, 3.4

[57] Z. Yuan, J. Ren, C.-M. Feng, H. Zhao, S. Cui, and Z. Li, "Visual programming for zero-shot open-vocabulary 3d visual grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20623–20633, 2024. 3.3.2, 3.3

[58] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, J. B. Tenenbaum, C. M. de Melo, M. Krishna, L. Paull, F. Shkurti, and A. Torralba, "Conceptfusion: Open-set multimodal 3d mapping," 2023. 3.3.3

[59] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, "An embodied generalist agent in 3d world," 2024. 3.3.3

[60] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. 3.3.3

[61] G. Zhou, Y. Hong, and Q. Wu, "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 7641–7649, 2024. 3.3.4

[62] B. Lin, Y. Nie, Z. Wei, J. Chen, S. Ma, J. Han, H. Xu, X. Chang, and X. Liang, "Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning," *arXiv preprint arXiv:2403.07376*, 2024. 3.3.4

[63] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," 2024. 3.4.1

[64] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," 2024. 3.4.1

[65] J. Zhang, S. Singh, *et al.*, "Loam: Lidar odometry and mapping in real-time.," in *Robotics: Science and systems*, vol. 2, pp. 1–9, Berkeley, CA, 2014. 3.4.1

[66] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," 2022. 3.4.1

[67] Q. Team, "Qwen2 technical report," 2024. 3.4.2

[68] Q. Team, "Qwen2.5 technical report," 2025. 3.4.2

[69] M. A. 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

[70] F. Yang, C. Cao, H. Zhu, J. Oh, and J. Zhang, "Far planner: Fast, attemptable route planner using dynamic visibility update," 2022. 3.4.5