DIV-Nav: Open-Vocabulary Spatial Relationships for Multi-Object Navigation

Jesús Ortega-Peimbert, Finn Lukas Busch, Timon Homberger, Quantao Yang, and Olov Andersson

***Abstract*— Advances in open-vocabulary semantic mapping and object navigation have enabled robots to perform an informed search of their environment for an arbitrary object. However, such zero-shot object navigation is typically designed for simple queries with an object label such as *'television'* or *'blue rug'*. While single- and multi-object search have progressed, real-time navigation with explicit spatial relationship reasoning during online map construction remains largely unexplored, as existing methods either rely on offline 3D reconstruction or handle only individual object targets without relational constraints. Here, we consider more complex free-text queries with spatial relationships, such as 'find the remote on the table'. We present DIV-Nav, a real-time semantic map-based navigation system for sequential multi-object search that addresses this problem through a series of relaxations: i) Decomposing natural language instructions with complex spatial constraints into simpler object-level queries, ii) computing the Intersection of individual semantic belief maps via continuous-valued scoring to identify regions where all objects co-exist, and iii) Validating discovered objects against the original spatial constraints via a vision-language model. We further investigate how to adapt frontier exploration for online semantic mapping to such spatial search queries to more effectively guide the search process. We validate our system through extensive experiments on the MultiON benchmark and real-world deployment on a Boston Dynamics Spot robot, achieving an 88% success rate on multi-object spatial-relationship navigation tasks. More details and videos are available at https://anonsub42.github.io/reponame/.**

I. Introduction

Effective robot navigation in human environments requires an understanding of natural language, where targets are often defined by their spatial relationships. While a command like *'find a chair'* only requires identifying a single object, real-world instructions often include spatial proximity constraints such as *'the chair next to the desk'* or *'the book on the nightstand'*. These queries are intuitive for humans, who naturally decompose them into constituent objects and reason about their arrangements. However, enabling robots to perform this reasoning in real-time during exploration remains a significant challenge.

This challenge is central to Object Goal Navigation (ObjectNav), the task where an agent must autonomously localize and navigate to a specific instance of an object category. Recent advances in Vision-Language Models (VLMs) have improved semantic understanding and 2D visual reasoning. Despite these gains, even large state-of-the-art models typically struggle with sufficient 3D understanding to robustly steer a robot over long-range navigation tasks by themselves [1].

To enable robust object navigation, current robotics research focuses mainly on hybrid approaches with conventional metric map exploration (e.g., frontier-based search) where the search objective is based on semantic similarity from a lightweight contrastive vision-language model such as CLIP [2]. Such contrastive vision-language models differ from modern large vision-language models (LVLMs) by having a simpler text encoder and only outputting a similarity score for each image and text query. Nevertheless, for simple objects these are capable of directing robotic search towards semantically promising directions (e.g., a kitchen if you are looking for a coffee machine) while retaining the robustness properties of conventional map-based navigation.

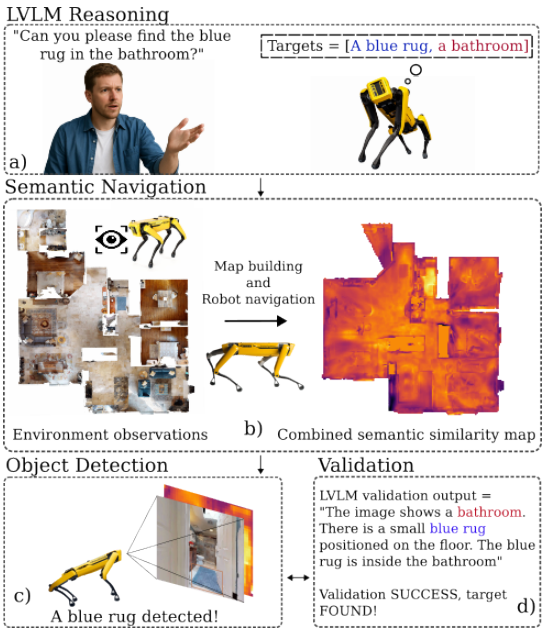


Fig. 1: Given a spatially-constrained natural language query, our system: (a) uses an LVLM to decompose the query into simpler proximity queries, (b) projects these into an online semantic belief map and intersects per-target similarity maps to localize regions where all objects likely co-exist, and (d) continuously validates candidate objects against the original constraints via an LVLM.

Initially, these approaches focused on single object queries [3]. More recently, online semantic mapping has been proposed that maintains open-set semantic embedding vectors in the map, so that accumulated semantic information can be reused for multiple, consecutive search queries of single objects [4]. However, neither of these handles queries with spatial relationship constraints, the gap DIV-Nav addresses.

We present DIV-Nav, a real-time navigation system that bridges this gap by combining LVLM-based reasoning with online semantic mapping. Our framework utilizes a three-stage process: (1) Decomposing complex instructions into

simpler proximity queries via an LVLM, (2) identifying regions where objects likely co-exist through a continuous-valued Intersection operation on semantic belief maps, and (3) Validating candidate regions in image space to confirm both object presence and spatial arrangements. This loop enables real-time semantic object navigation with complex spatial relationship queries in free-text form without requiring offline 3D reconstruction while maintaining real-time performance.

The contributions of our work include:

- We propose a robust map-based method for online open-vocabulary object search with spatial constraints. By using an LVLM to decompose the constraints into individual object queries with relaxed proximity relationships, we show that these can be efficiently searched for online with a spatial-semantic belief map, and resulting candidates later verified against the original spatial constraint in image-space.
- We introduce a semantic intersection operation for combining individual similarity maps from the decomposed object queries into a joint belief map, identifying regions where all queried objects are likely to co-exist. We investigate how to incorporate this combined map into the frontier exploration objective of an online semantic mapping system, guiding active exploration toward spatially relevant regions without overly constraining the search before all objects have been observed.
- We demonstrate the system's effectiveness through comprehensive evaluation on a public benchmark and via quantitative real-world experiments with a Boston Dynamics Spot robot across four indoor environments. We make the code available to the community.[1]

II. Related Work

A. Zero-shot Object Navigation

Visual-language navigation approaches [5], [6], [7], [8] enable agents to follow step-by-step instructions to reach a target. Yokoyama et al. [3] proposed Vision-Language Frontier Maps (VLFM), a zero-shot framework that combines occupancy mapping with semantic reasoning from a pre-trained VLM such as BLIP-2 [9]. Their approach computes semantic similarity between real-time RGB observations and text prompts, projects the scores onto a top-down value map, and uses depth and odometry to build an occupancy map.

Huang et al. [10], [11] introduced Visual Language Maps (VLMap), which fuses pretrained visual–language features with 3D reconstructions to create persistent, language-indexable spatial-semantic representations. While the resulting VLMaps allow for descriptive, spatial queries, the approach assumes the map to be pre-generated at navigation time. These works collectively illustrate the potential of combining vision-language understanding with spatial mapping and planning. More recently, Busch et al. [4] presented OneMap, a real-time, open-vocabulary mapping framework that constructs a belief map over CLIP-aligned features [2].

[1]We make the code available on acceptance.

Shah et al. [12] demonstrated that composing multiple pre-trained models can enable navigation without language-annotated trajectory datasets. Similarly, Long et al. [13] proposed InstructNav, a unified framework capable of handling multiple instruction formats in a zero-shot setting. InstructNav's Dynamic Chain-of-Navigation (DCoN) module translates instructions into action–landmark pairs that are continuously updated based on new observations. In contrast, DIV-Nav builds a single reusable semantic map online that can be queried consecutively for complex spatial object queries, enabling spatial relationship reasoning even on partially explored maps.

Recent efforts have addressed spatial constraints using relational graphs in MLFM [14] or multi-channel correlation maps in Finder [15]. Unlike these, DIV-Nav avoids complex graphs by employing a continuous-valued intersection of semantic belief maps to identify proximity-based co-existence. We further distinguish our work from methods in related domains: O3D-SIM [16] and ConceptFusion [17] similarly resolve spatial language queries against a semantic map, but require a complete pre-built reconstruction before any queries can be answered and perform no online search planning. CORE-3D [18] follows the same offline retrieval paradigm. Additionally, LeGo-Drive [19] addresses language-enhanced navigation for autonomous driving maneuvers, such as lane changes and parking, which differ fundamentally from the indoor object search primitives addressed in our work.

B. Learned Object Navigation

Learning-based ObjectNav has progressed via two main paths. End-to-end RL, initiated by DD-PPO [20] and refined by modular semantic-map policies like SemExp [21], has scaled to large procedural environments using powerful visual encoders [22]. Simultaneously, imitation learning reduces interaction via human demonstrations: PIRLNav [23] combines behavior cloning with RL finetuning, while newer methods utilize trajectory diffusion [24] or unlabeled egocentric video [25], though both necessitate substantial offline training.

A parallel effort integrates foundation models into learned pipelines. CL-CoTNav [26] fine-tunes a VLM with chain-of-thought reasoning on expert trajectories, while Scene-LLM [27] aligns 3D voxel representations with an LLM through multi-stage training. Others encode semantic information via implicit neural representations [28]. These methods improve scene understanding but remain dependent on task-specific training data and fixed object vocabularies, limiting their ability to handle novel spatial relationship queries at test time.

In contrast, recent zero-shot approaches have matched or exceeded learned methods without navigation-specific training [3], [4]. Our work extends this line to compositional spatial queries, a capability absent from both learned and existing zero-shot methods, while automatically benefiting from improvements in the underlying foundation models without retraining.

III. METHOD

Our proposed approach addresses the problem of guiding a robot towards a navigation goal that is characterized by targets with spatial relationships, inferred from a natural language input. While the traditional ObjectNav problem typically targets simple object categories, such as *'a chair'*, our method enables inferring objects from abstract descriptions and searching for distinct object instances by accounting for spatial relationship constraints.

The proposed pipeline involves leveraging a compact LVLM like multimodal Phi4 5.6B [29] (capable of running onboard robots) to decompose the natural language inputs into multiple navigation targets that can be queried against online-constructed semantic maps, combining the resulting similarity maps to identify locations where spatial relationships are likely satisfied, and employing visual validation to confirm goal achievement. In the following, we will refer to such models that can both process and produce natural language, as well as process visual inputs as LVLMs. Note that this differs from language-aligned vision models such as CLIP [2] which cannot produce and process language in the same way.

A. Problem Formulation

The system is provided with a continuous stream of posed RGB-D observations $\{I_t, D_t\}$ where $I_t \in \mathbb{R}^{H\times W\times 3}$ are RGB images and $D_t \in \mathbb{R}^{H\times W}$ contains corresponding depth information. Furthermore, it is provided with an input in the form of a natural language target description L. L either explicitly or implicitly indicates targets as well as inter-target spatial relationships. The desired output consists of navigation actions that guide the agent to the primary target that satisfies both, the target type and the spatial relationships specified in L.

B. System Overview

An overview of the proposed system is given in Fig. 2. Our approach involves exploring a previously unseen environment and incrementally building a semantic representation (Fig. 2 (b)), to guide a robot towards a specified navigation target. To this end, we adopt the uncertainty-aware semantic belief mapping and navigation framework of [4]. This system leverages patch-level image embeddings produced by encoding the RGB frames I_t using the backbone of a CLIP-aligned segmentation model [30][2] and grounds them on a 2D grid map using depth frames D_t and corresponding poses. A Gaussian blurring kernel is applied during the integration of features to account for depth sensing noise. Querying the resulting semantic map with a text embedding yields a 2D map S of similarity values.

While this similarity map provides rich spatial-semantic information, it can only be queried using relatively simple object classes that align with CLIP's feature space, limiting its applicability to complex language instructions that contain spatial relationships. We bridge this gap by employing an LVLM as an intermediary reasoning layer that translates natural language queries into sets of basic object queries Q compatible with the semantic map's CLIP-based representation (Fig. 2 (a)). The basic object queries are determined under consideration of their spatial relationships. Hereby, we focus on *proximity* relationships, which allows us to model natural language descriptions such as *'inside'*, *'on top of'*, *'near'*, etc. Moreover, this enables us to use the LVLM to infer higher-level concepts (e.g. *'towel'* $\rightarrow$ *'bathroom'*), readily capturing their relationship as proximity, which can help provide guidance early on when the environment has only been sparsely explored.

Similarity maps S_i that result from querying the semantic map with $q_i \in Q$ are combined to produce a joint similarity map S_{comb} (Fig. 2 (c)). High-similarity regions in S_{comb} indicate locations where the targets corresponding to q_i are likely located close to each other and are used to guide the navigation module (Fig. 2 (d)) inherited from [4]. Crucially, rather than treating this as a static search, our framework incorporates S_{comb} into the online frontier exploration objective. By weighting unexplored boundaries (frontiers) with these similarity scores, the system actively biases the search process toward spatially relevant regions even before all objects have been explicitly observed in the environment. Upon reaching such a location, we utilize the LVLM to validate discovered objects against the original spatial constraint specifications in L (Fig. 2 (e)). If the validation does not confirm object presence or spatial relation, the search continues.

C. Decomposing Natural Language Queries

To decompose a given natural language query into object and location targets for which the semantic map can be queried, we prompt the LVLM to reason over the given command and extract the following information:

1) Identify all objects and locations explicitly mentioned in the instructions, T_{expl}.
2) Infer reasonable higher-level concepts or locations if not explicitly stated, T_{inferred}. For *'a towel'*, the LVLM might infer *'kitchen'* or *'bathroom'*.
3) Infer possible objects if the prompt is a demand rather than an explicit target T_{impl}. For *'the room is on fire!'*, the LVLM extracts *'fire extinguisher'*.
4) Identify the primary target $\hat{T}$, and the spatial relationship of all other extracted targets with the primary target, $R(\cdot, \hat{T})$. For *'the blue rug in the bathroom'*, *'rug'* becomes the primary target, and the relationship is *'in'*.

As a result, we obtain a primary target $\hat{T}$, and a list of other relevant targets $T_{\text{all}} = \hat{T} \cup T_{\text{expl}} \cup T_{\text{inferred}} \cup T_{\text{impl}}$. We further filter this list to retain only targets $t \in T_{\text{all}}$ where $R(t, \hat{T})$ can be understood as a proximity relationship. For instance, we retain proximity relationships such as *'in'*, *'on top'*, *'near'*, but discard non-proximity relationships such as *'not in'*, *'far from'*. As a result, for *'the rug **not** in the bathroom'*, we would discard *'bathroom'* from the list. We denote the resulting filtered target set Q.

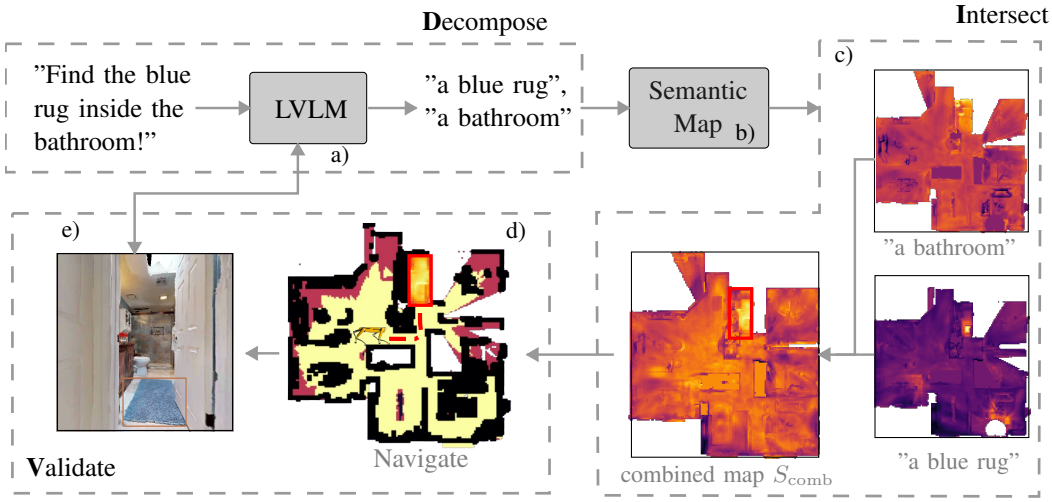


Fig. 2: Overview of DIV-Nav. (1) We **D**ecompose spatially-constrained instructions into object-level queries on a semantic map (a-b); (2) Compute Intersections to create a joint belief map that actively guides online frontier exploration toward regions where objects likely co-exist (c-d); (3) **V**alidate high-similarity regions by approaching the region, and via LVLM reasoning to confirm individual objects and their spatial relationships (e).

D. Querying the Map

Given the decomposed target set Q, the goal of this module is to identify map regions where all constituent objects are likely to co-exist in physical space. Each query $q_i \in Q$ is encoded with the CLIP [2] text encoder to obtain an embedding $q_{\text{CLIP},i} \in \mathbb{R}^f$, and its cosine similarity against every map cell yields a per-query similarity map $S_i(x, y)$.

To aggregate these individual maps into a single *spatial co-existence score*, we require a score that is high only where *all* objects are simultaneously likely to be present, which is the defining condition for proximity relationships such as *'on'*, *'inside'*, and *'next to'*. We formulate this as a continuous-valued intersection, i.e. $S_{\text{int}} = \min_i S_i(x, y)$. The minimum has a clear geometric interpretation: a cell scores highly only if every constituent object has high semantic similarity there, and is suppressed as soon as any one object is unlikely to be present. This is strictly tighter than alternatives such as a sum or average, which can yield high scores even when individual object matches are weak. Crucially, the operation is applied over *continuous* similarity values rather than binary thresholds, which avoids per-object threshold tuning in 2D map space and preserves graded navigation guidance throughout exploration.

A useful property of this formulation is its interaction with the spatial blurring inherent to the semantic map, which applies a Gaussian kernel during feature integration to account for depth noise and FOV boundary uncertainty. This blurring causes similarity responses for spatially adjacent objects to overlap, so S_{int} correctly highlights their shared vicinity even for relationships like *'next to'* that do not correspond to a strict geometric intersection.

At early stages of exploration, unobserved targets produce near-uniform low similarity across the map, which would suppress S_{int} and leave the robot without directional guidance. We address this by blending the intersection with the strongest individual match:

$$S_{\text{comb}}(x, y) = \alpha \cdot S_{\text{int}}(x, y) + (1 - \alpha) \cdot \max_i S_i(x, y), \quad (1)$$

where α = 0.9 is fixed throughout our navigation, strongly prioritizing S_{int} once sufficient coverage exists, while the max term helps the robot toward the most semantically relevant observed region during sparse early exploration. For queries where no additional context targets are extracted, $Q = \{\hat{T}\}$ and S_{comb} reduces to $S_{\hat{T}}$, recovering standard single-object behavior. An example of the decomposition and resulting similarity maps is shown in Fig. 2.

E. Navigation and Validation

The navigation strategy, including the map representations and frontier-scoring mechanism, is adopted directly from [4] and extended here to operate on the multi-object combined map S_{comb}, which directs the robot toward regions most likely to satisfy the full spatial arrangement specified in the query. The system maintains four binary maps derived from the uncertainty estimates of the semantic map: the **Observed Map** $\mathcal{O}$ (regions reached by at least one sensor update), the **Semantically Explored Map** $\mathcal{E}$ (regions where uncertainty falls below 0.2, indicating reliable semantic features), the **Searched Map** $\mathcal{C}$ (same threshold, but for the current query; reset at each new query), and the **Navigable Map** $\mathcal{N}$ (obstacle-free regions available for path planning).

Navigation candidates are drawn from two sources. **Frontier candidates** lie on the boundary between $\mathcal{E}$ and $\mathcal{O}$, representing areas observed but not yet semantically reliable; each frontier is scored by $\max(S_{\text{comb}}(x, y))$ over reachable cells in $\mathcal{O} - \mathcal{E}$. **Cluster candidates** are high S_{comb} regions within $(\mathcal{E} - \mathcal{C})$, representing previously explored areas that warrant revisiting under the current query. Each cluster is scored by its peak S_{comb} value.

A key property of this frontier scoring is that S_{comb}, through its intersection term, biases exploration toward re-

gions where *all* decomposed sub-targets are likely to co-exist. This steers the robot away from isolated instances of any single object and toward the specific location where the full spatial relationship can be verified, a behavior that single-object navigation scores cannot produce.

The robot greedily selects the highest-scoring candidate from either source, and A* on $\mathcal{N}$ computes a collision-free path. During navigation, object detections are gated by a consensus filter: a candidate is accepted only when the detector fires *and* the current cell falls in the top 5-percentile of S_{comb}, suppressing detections of objects that match the target type but are in the wrong spatial context.

Upon a passing detection, the robot approaches within $0.5\,\text{m}$ and triggers LVLM validation, prompting it to confirm (1) that the primary target is present and (2) that the original spatial constraint is satisfied. If the LVLM confirms that the conditions are satisfied, we terminate the search. We reset the searched map, and receive a new natural language query if available.

The Decompose-Intersect-Validate framework addresses three key challenges in spatial relationship navigation. First, decomposition via LVLM enables translation from complex natural language (which CLIP cannot directly encode) into simpler object-level queries compatible with semantic maps. Second, continuous-valued intersection provides spatial reasoning without requiring explicit 3D instance segmentation or object pose estimation—high-scoring regions in the intersection map indicate where multiple object beliefs overlap, suggesting spatial proximity. Third, LVLM validation is necessary because semantic maps may have high similarity scores for objects that match the type but not the specific instance or spatial arrangement (e.g., multiple tables in a room). Together, these components enable efficient spatial search (via map-based reasoning) while maintaining accuracy (via visual validation).

IV. Experimental Setup

We evaluate our method for multi-object navigation tasks given in natural language in the MultiON Challenge 2024 benchmark [31], as well as through real-robot navigation experiments. We use Phi-4-multimodal-instruct [29] as our LVLM model. For object detection, see Sec. III-E, we follow [4] and use YOLOv7 [32] for MS-COCO [33] classes, and an open-set detector Yolo-World [34] otherwise.

Multi-Object Navigation in HSSD: The MultiON 2024 challenge extends the traditional object navigation framework presented in [35] by including navigation targets described by language instructions, such as *'find the red short pillar candle on the grey nightstand'*. Each of these language instructions may contain fine-grained descriptions of the target (e.g. *'find the mini spa candle'*) or may also contain spatial relations between objects, such as *'the mantel clock on the chest of drawers'*.

The task requires agents to navigate to a sequence of three objects located within realistic 3D environments from the Habitat Synthetic Scenes Dataset (HSSD) [36]. Each episode begins with the agent at a random starting position and orientation within an unseen environment. The agent receives a sequence of three natural language descriptions corresponding to target objects that must be found in sequence, but is only informed about the subsequent target once the previous has been found. Navigation is considered successful when the agent calls the FOUND action within $0.5\,\text{m}$ of each target object. The episode terminates either when all objects are successfully found or when an incorrect FOUND action is called. We evaluate on the minival split of the MultiON challenge, which consists of 100 episodes with a sequence of three target objects in 20 scenes. The agent has access to RGB-D camera observations, providing 256×256 resolution images with a 79° horizontal field of view, and a noiseless GPS+Compass sensor that provides location and orientation of the agent relative to the agent's initial position at episode start. The action space consists of discrete navigation primitives: move forward $0.25\,\text{m}$, turn left/right 15° and the FOUND action to indicate object discovery.

Multi-Object Navigation in the Real World: We deploy our DIV-Nav system on a Boston Dynamics Spot quadruped robot. For observations, the system uses a single front-facing RealSense D455 stereo depth camera and a Livox Mid 360 lidar running Fast-LIO2 [37] for odometry. While LiDAR is utilized here to ensure robust odometry via Fast-LIO2, we emphasize that the DIV-Nav framework is sensor-agnostic regarding pose estimation. The choice of odometry source affects the global consistency of the metric map but does not alter the fundamental algorithmic approach to semantic mapping or spatial relationship reasoning. For real-world experiments, we adapt the feature localization parameters of the mapping to account for depth noise. Except for GPT-4o-mini, we run the entire stack on-board a Jetson Orin AGX, with the robot following a path-tracking controller publishing standard velocity commands. The full stack runs at 1.6 Hz map update rate; LVLM decomposition averages 1.046 s and LVLM validation averages 1.757 s per candidate, both infrequent relative to the map update cycle.

Inspired by the Multi-On challenge episode setup, we conduct 15 Multi-Object navigation experiments across four different scenes: an office waiting room, a robotics lab, a kitchen area, and an office lounge. The selected real-world scenes encompass significant variety in layout and illumination, ranging from naturally lit office lounges to artificially lit laboratory environments. This variety allows us to evaluate the robustness of the underlying CLIP-based semantic features to lighting variations and structural differences, such as open floor plans and narrow corridors. The targets cover both simple objects and more complex target descriptions with spatial relationships. In contrast to the MultiON benchmark, we do not terminate an episode if the robot incorrectly identifies a target and allow it to always attempt all three targets per episode. Note that we report the success rate (SR) *per object*.

Metrics: For multi-object navigation, we report *Progress* (Pr), the average fraction of found objects from the total number of targets per episode, as well as *Success* (SR), the

percentage of episodes where the agent successfully finds all target objects in the sequence. We further define *Success Rate per Attempted Target* (SRAT) as the success rate only for attempted object goals.

Baselines: We compare against the two MultiON challenge submissions, i.e. MOPA [38] and IntelliGO Labs [31] which were trained for the task using RL.

We include [4] as an additional baseline to isolate our spatial reasoning contribution; since it handles only simple object queries, we provide it with the primary target via our decomposition module (Sec. III-C).

To characterize query complexity, we define three dimensions: (1) the number of target objects extracted by the decomposition step, (2) the number of spatial relationships present, and (3) the relationship type: proximity/containment (e.g., *'on'*, *'in'*, *'near'*) versus non-proximity (e.g., *'far from'*, *'not in'*). Notably, fine-grained single-object descriptions without relational structure challenge CLIP's feature space rather than the spatial reasoning pipeline, and represent a distinct failure mode from relational queries.

The map covers a 60×60 m area discretized into a uniform grid of 600×600 cells (0.1 m/cell) for simulation, and 300×300 cells (0.2 m/cell) for real-world deployment.

V. Results

Our experiments were designed to answer two key questions: (1) Can our semantic intersection approach handle spatially-constrained navigation queries better than traditional zero-shot object-navigation approaches? (2) Can our method be successfully deployed on a real robot for natural language command search in real environments?

A. Multi-Object Navigation in HSSD

We evaluate DIV-Nav against several baselines on the MultiON challenge minival split. Table I presents our results compared to MOPA [38], IntelliGO Labs (the winning submission from the 2024 challenge) [31], and OneMap [4] provided with the primary target.

Approach	SRAT ↑	Pr ↑	SR ↑	FP (↓)	TNF (↓)
MOPA [38]	0.05	0.02	0.00	-	-
IntelliGO Labs [31]	0.23	0.10	0.03	-	-
OneMap [4]	0.24	0.10	0.03	0.76	0.0
DIV-Nav (Ours)	**0.30**	**0.14**	**0.04**	0.44	0.25

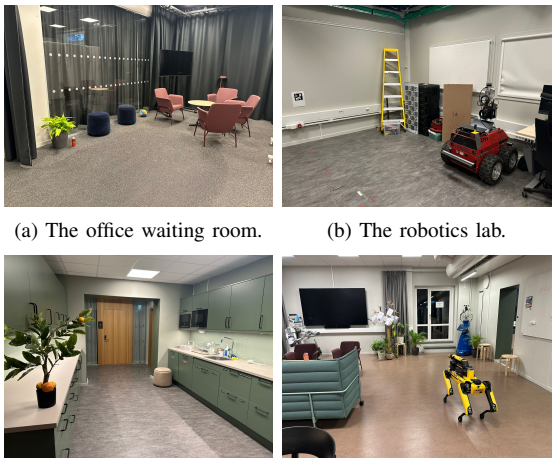
TABLE I: Comparison of navigation performance across Success Rate per Attempted Target (SRAT), Progress (Pr), and Success Rate (SR). To further understand different failure cases, we report the False Positive Rate (FP), where the agent called the FOUND action on a wrong target, and the Terminate-Not-Found Rate (TNF) where the agent terminates the search without calling the found action.

DIV-Nav achieves the best performance across all metrics, with a 30% improvement in SRAT over the baselines and a 40% improvement in Progress. While absolute performance remains challenging due to the complexity of the benchmark, our approach demonstrates advantages in handling spatially-constrained queries. We further observe that our method reduces false positives compared to [4], which we attribute to our method being able to reason about spatial constraints, instead of just searching for the primary target. In contrast to that, [4] might call the FOUND action on a target that matches the primary target, but does not satisfy the full constraint.

The relatively low absolute performance across all methods can be attributed to several factors: (1) Low-resolution observations: The 256×256 RGB-D images limit detailed object recognition; (2) LVLM performance in simulation: Vision-language models show reduced effectiveness on the simple graphics used in the benchmark compared to real-world images; (3) Highly challenging queries: Using our complexity characterization from Section IV, the benchmark's hardest queries are predominantly fine-grained single-object descriptions (one target, zero spatial relationships) such as *'Find the freestanding bath/shower mixer'* and *'find the cream chair by Joanna Gaines'*, which challenge CLIP's fine-grained discrimination capacity rather than the spatial reasoning pipeline, and represent limitations of the underlying semantic map rather than the DIV-Nav framework itself. Lastly, our semantic map struggles with small or occluded objects in the feature space.

Despite these challenges, our semantic intersection approach shows consistent improvements over baselines, validating our core hypothesis that decomposing complex spatial queries and combining similarity maps leads to better navigation performance.

B. Real World Experiments



(a) The office waiting room. (b) The robotics lab.

(c) The kitchen area. (d) The office lounge.

Fig. 3: Four real-world experiment scenes. We conducted 15 multi-object navigation episodes across these environments.

We conducted 15 multi-object navigation episodes across four realistic environments as described in the evaluation setup. The environments are shown in Fig. 3. Table II compares our method against [4] on real-world navigation tasks.

Our system achieves an 88% success rate, successfully finding 40 out of 45 total targets across all experimental runs. This represents a 66% relative improvement over the

Approach	SR / # Found (↑)	FP (↓)	TNF (↓)
OneMap[4]	53% / 24	40%	6.66%
DIV-Nav (Ours)	**88% / 40**	0%	11.11%

TABLE II: Comparison of real-world performance. Metrics: Success rate (SR) per object, number of objects found (# Found), false positive rate (FP), and rate of attempts where the robot terminates the search without finding the object (TNF).

OneMap baseline [4] (and is statistically significant, 95% Clopper-Pearson bounds). We further report the false positive rate (FP), which shows that [4] fails a substantial number of attempts due to misidentifying wrong objects, which we attribute to it not accounting for the spatial relationships. As a consequence, [4] might call the FOUND action on objects that match the primary object, but do not satisfy the spatial constraints, leading to false positives. We also observe significantly increased performance on real-world experiments for both methods, which we attribute to the challenges encountered in the benchmark that are not present in real-world. Moreover, the performance gap between [4] and DIV-Nav is larger in real-world experiments, since the proportion of queries containing spatial relationships is larger, which requires the system to possess capabilities to take spatial constraints into account. The 11.11% TNF rate has two contributing causes. First, LVLM validation occasionally rejects correct detections, as in the case of the red car in the robotics lab (Fig. 3b). Second, when a stated spatial relationship is factually unsatisfied in the environment, for instance when no television is present near any blue rug, S_{comb} (Eq. 1) never produces a high-scoring region satisfying both targets simultaneously. Note that while the maximum term in Eq. 1 still guides the robot toward the individually highest-scoring object, the subsequent LVLM validation will reject it as the spatial constraint remains unmet. Both cases result in a TNF outcome, and handling unsatisfiable spatial constraints remains an important direction for future work.

While the quantitative gains in Tables I and II represent the combined framework, our real-world experiments highlight the distinct role of the Intersection module in driving search efficiency. Specifically, by identifying regions of proximity where sub-targets likely co-exist, the system creates a focused search area that allows the robot to bypass isolated object instances. This is evidenced by the *'birthday robot'* task, where the robot was directly guided towards the intersection region rather than naively validating every detected instance of a robot or tree in the environment.

For one of the experiments in the environment depicted in Fig. 3 (d), we provide the system with the query: *'Find me the birthday robot with his Christmas tree'*. In the experiment, the system correctly: (1) **Decomposes the query**: Identifies *'robot'*, *'Christmas tree'*, and their proximity relationship; (2) **Builds combined similarity map**: Creates intersection regions where both objects are likely to co-exist; and (3) **Validates spatially**: Confirms both the presence of individual objects and their spatial relationship through LVLM reasoning.

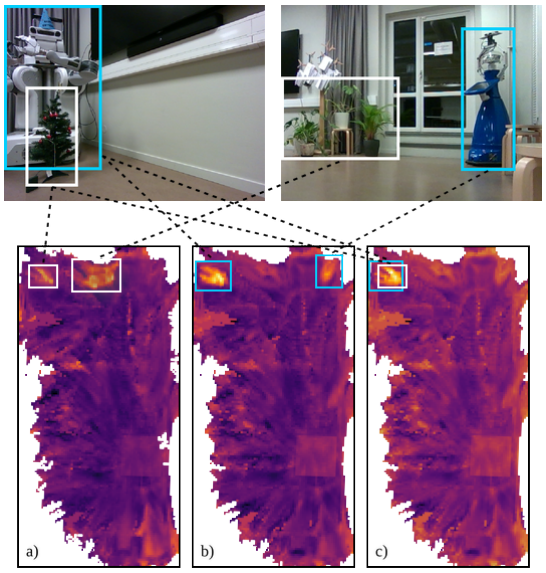


Fig. 4: Data of a real world experiment: (top) RGB frames, (a) similarity map for query *'plant'* , (b) similarity map for query *'robot'* (c) and the resulting map of value intersection scores.

To illustrate the capability of our value intersection scoring module to highlight specific object instances based on spatial proximity, Fig. 4 shows data collected during this experiment. We compute similarity maps for the queries: *'plant'* and *'robot'*, (Fig. 4 (a) and (b)) and the corresponding value intersection score (Fig. 4 (c)).

We identified two primary failure modes in our real-world experiments. First, **small or hidden objects**: Objects like a volleyball hidden in the waiting room or an electric kettle on a shelf failed to be detected due to limited visibility or small size in the semantic map's feature space. Second, **LVLM validation failures**: In some cases, objects with high similarity scores in the semantic map failed LVLM validation. For example, the red car in the robotics lab (Fig. 3b) was correctly mapped but incorrectly rejected during the visual validation step.

VI. Conclusion

In this work, we presented DIV-Nav, a multi-object navigation method that extends zero-shot navigation frameworks to handle spatially-constrained targets specified through natural language commands. By decomposing spatial queries into simpler semantic components via LVLMs and combining the resulting similarity maps, our approach enables robots to navigate to objects defined not just by what they are, but by where they are found relative to other objects.

Several limitations point to future directions. The semantic mapping struggles with small or occluded objects due to CLIP feature space limitations and map resolution constraints, VLM validation occasionally fails even when the semantic map correctly identifies target locations, and our approach is currently limited to proximity-based spatial relationships — generalizing to *'far from'*, *'between'*, or directional relationships would broaden applicability. Finally, integrating more advanced local planners to handle dynamic

obstacles remains an avenue beyond the semantic reasoning framework proposed here.

REFERENCES

[1] W. Ma, Y.-C. Chou, Q. Liu, X. Wang, C. M. de Melo, J. Xie, and A. Yuille, "Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. [Online]. Available: https://openreview.net/forum?id=hFaXVjRFHI

[2] 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*. PmLR, 2021, pp. 8748–8763.

[3] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 42–48.

[4] F. L. Busch, T. Homberger, J. Ortega-Peimbert, Q. Yang, and O. Andersson, "One map to find them all: Real-time open-vocabulary mapping for zero-shot multi-object navigation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 14 835–14 842.

[5] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 15 439–15 449.

[6] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *European Conference on Computer Vision*. Springer, 2020, pp. 104–120.

[7] W. Wu, T. Chang, X. Li, Q. Yin, and Y. Hu, "Vision-language navigation: a survey and taxonomy," *Neural Computing and Applications*, vol. 36, no. 7, pp. 3291–3316, 2024.

[8] 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*, 2018, pp. 3674–3683.

[9] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[10] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," *arXiv preprint arXiv:2210.05714*, 2022.

[11] ——, "Multimodal spatial language maps for robot navigation and manipulation," *The International Journal of Robotics Research*, p. 02783649251351658, 2025.

[12] D. Shah, B. Osiński, S. Levine *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on robot learning*. PMLR, 2023, pp. 492–504.

[13] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "Instructnav: Zero-shot system for generic instruction navigation in unexplored environment," *arXiv preprint arXiv:2406.04882*, 2024.

[14] S. Raychaudhuri, E. Cancelli, T. Campari, L. Ballan, M. Savva, and A. X. Chang, "Mlfm: Multi-layered feature maps for richer language understanding in zero-shot semantic navigation," 2025. [Online]. Available: https://arxiv.org/abs/2507.07299

[15] D. Choi, A. Fung, H. Wang, and A. H. Tan, "Find everything: A general vision language model approach to multi-object search," 2025. [Online]. Available: https://arxiv.org/abs/2410.00388

[16] L. Nanwani, K. Gupta, A. Mathur, S. Agrawal, A. H. A. Hafez, and K. M. Krishna, "Open-set 3d semantic instance maps for vision language navigation – o3d-sim," *Advanced Robotics*, vol. 38, no. 19–20, p. 1378–1391, Aug. 2024. [Online]. Available: http://dx.doi.org/10.1080/01691864.2024.2395926

[17] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," *arXiv preprint arXiv:2302.07241*, 2023.

[18] M. A. Mirzaei, P. Amoie, A. Ekhterachian, M. Mirzababaei, and B. Khalaj, "Core-3d: Context-aware open-vocabulary retrieval by embeddings in 3d," 2025. [Online]. Available: https://arxiv.org/abs/2509.24528

[19] P. Paul, A. Garg, T. Choudhary, A. K. Singh, and K. M. Krishna, "Lego-drive: Language-enhanced goal-oriented closed-loop end-to-end autonomous driving," 2024. [Online]. Available: https://arxiv.org/abs/2403.20116

[20] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," in *International Conference on Learning Representations*.

[21] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4247–4258, 2020.

[22] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 829–14 838.

[23] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "Pirlnav: Pretraining with imitation and rl finetuning for objectnav," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 896–17 906.

[24] X. Yu, S. Zhang, X. Song, X. Qin, and S. Jiang, "Trajectory diffusion for objectgoal navigation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 110 388–110 411, 2024.

[25] N. Hirose, C. Glossop, A. Sridhar, D. Shah, O. Mees, and S. Levine, "Lelan: Learning a language-conditioned navigation policy from in-the-wild videos," in *Conference on Robot Learning*, 2024.

[26] Y. Cai, X. He, M. Wang, H. Guo, W.-Y. Yau, and C. Lv, "Cl-cotnav: Closed-loop hierarchical chain-of-thought for zero-shot object-goal navigation with vision-language models," *arXiv preprint arXiv:2504.09000*, 2025.

[27] R. Fu, J. Liu, X. Chen, Y. Nie, and W. Xiong, "Scene-llm: Extending language model for 3d visual understanding and reasoning," *arXiv preprint arXiv:2403.11401*, 2024.

[28] P. Marza, L. Matignon, O. Simonin, and C. Wolf, "Multi-object navigation with dynamically learned neural implicit representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 004–11 015.

[29] Microsoft, A. Abouelenin, A. Ashfaq *et al.*, "Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras," 2025. [Online]. Available: https://arxiv.org/abs/2503.01743

[30] B. Xie, J. Cao, J. Xie, F. S. Khan, and Y. Pang, "Sed: A simple encoder-decoder for open-vocabulary semantic segmentation," 2024. [Online]. Available: https://arxiv.org/abs/2311.15537

[31] (2024) Multion eai challenge (cvpr 2024). First place: IntelliGO (Francesco Taioli, Marco Cristani, Alberto Castellini, Alessandro Farinelli, Yiming Wang). [Online]. Available: https://multion-challenge.cs.sfu.ca/

[32] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 7464–7475.

[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[34] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 16 901–16 911.

[35] S. Wani, S. Patel, U. Jain, A. X. Chang, and M. Savva, "Multi-on: Benchmarking semantic map memory using multi-object navigation," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[36] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[37] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: fast direct lidar-inertial odometry," *CoRR*, vol. abs/2107.06829, 2021. [Online]. Available: https://arxiv.org/abs/2107.06829

[38] S. Raychaudhuri, T. Campari, U. Jain, M. Savva, and A. X. Chang, "Mopa: Modular object navigation with pointgoal agents," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 5763–5773.