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 name like ”television” or ”blue rug”. Here, we consider more complex free-text queries with spatial relationships, such as ”find the remote on the table” while still leveraging robustness of a semantic map. We present DIV-Nav, a real-time navigation system that efficiently addresses this problem through a series of relaxations: i) Decomposing natural language instructions with complex spatial constraints into simpler object-level queries on a semantic map, ii) computing the Intersection of individual semantic belief maps to identify regions where all objects co-exist, and iii) Validating the discovered objects against the original, complex spatial constrains via a LVLM. We further investigate how to adapt the frontier exploration objectives of 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 using a Jetson Orin AGX. More details and videos are available at https://anonsub42.github.io/reponame/.**

I. Introduction

Robots operating in human environments must interpret natural language commands that go beyond simple object identification. While a command like ”find a chair” requires handling simple object classes only, real-world search instructions often specify spatial relationships: ”go to the chair next to the desk,” ”find the towel in the bathroom,” or ”get the book on the nightstand.” These spatially-constrained queries are intuitive for humans, who naturally decompose them into constituent objects, reason about their spatial arrangements, and navigate to locations where these relationships are satisfied. However, enabling robots to perform this same reasoning remains a significant challenge.

Recent advances in Vision-Language Models (VLMs) have transformed how robots understand their environments. These models demonstrate impressive capabilities in semantic understanding, visual reasoning, and natural language comprehension. However, even large visual-language-models (LVLMs) like GPT5, that can display impressive image understanding capabilities, do not yet have a sufficiently robust understanding of space and motion to steer a robot alone. Previous work has therefore successfully integrated lightweight VLMs like CLIP [1] into conventional spatial representations such as frontier maps used for open-vocabulary single- and multi-object navigation [2][3]. Yet, existing object navigation methods typically cannot handle tasks that require explicit reasoning about spatial relationships between objects. As a result, they often fail on queries such as *“the potted plant next to the TV”*, where navigation requires resolving multi-object spatial constraints.

To this end, current zero-shot object navigation methods face two limitations when handling spatial queries. First,

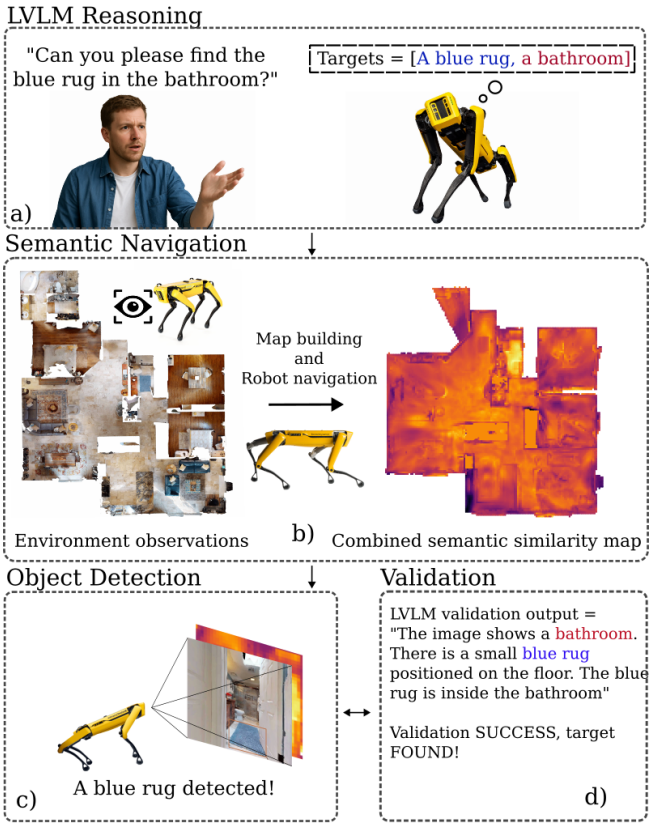

Fig. 1: Overview of the object searching process. From a spatially-constrained navigation query given in natural language, our system: (a) employs an LVLM module to reduce the complex query into simpler spatial proximity queries, (b) projects these targets into a reusable semantic belief map to guide the search towards regions in the physical space where these targets are more likely to co-exist, (c) detects potential targets through open-set object detection, and (d) continuously validates candidate objects against the complex spatial query via visual inference of an LVLM.

*This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, as well as Digital Futures.

The authors are with the Division of Robotics, Perception, and Learning, KTH Royal Institute of Technology, Sweden. Contact: {jgop, flbusch, timonh, quantao, olovand}@kth.se.

they typically construct query-specific representations that do not capture relationships between multiple objects. For instance, VLFM [2] builds similarity maps for individual queries but does not explicitly reason about "next to" or "inside" relationships using the map. Second, while methods like OneMap [3] build queryable semantic maps for multi-object search, they only allow to obtain correlations between the map and individual objects, and do not offer a straightforward way to reason about spatial relationships between them.

In this work, we present DIV-Nav, a real-time navigation system that bridges this gap by combining LVLM-based reasoning with incremental semantic exploration and mapping. The key insight of DIV-Nav is that LVLMs can serve as a tool that **D**ecomposes natural language queries with complex multi-object spatial relationships into simpler, relaxed object proximity queries, while semantic maps can provide the spatial grounding necessary to robustly search in space for candidate regions where such semantic object beliefs **I**ntersect, and for each such region the LVLM can again be used to **V**erify in image space if the original, complex spatial constraints are satisfied.

The contributions of our work include:

- We propose a method for open-vocabulary object search with complex spatial constraints that uses a LVLM to relax the constraints into individual object targets and proximity constraints that can efficiently be searched for in space with a spatial-semantic belief map, and the candidates later verified against the original complex constraint with the LVLM.
- We introduce a method to combine individual similarity maps from the decomposed object targets to identify map regions that likely satisfy the spatial relationships specified in the original complex query, and further investigate how to incorporate such spatial constraint information also into a frontier exploration objective without overly constraining exploration before the objects have been seen.
- We demonstrate the system's effectiveness through comprehensive evaluation on the MultiON benchmark as well as a series of real-world deployments with a Boston Dynamics Spot robot.

II. Related Work

A. Visual-Language Object Navigation

Visual-language navigation approaches [4], [5], [6], [7] enable agents to follow step-by-step instructions to reach a target destination. Yokoyama et al. [2] proposed Vision-Language Frontier Maps (VLFM), a zero-shot framework that combines occupancy mapping with semantic reasoning from a pretrained VLM such as BLIP-2 [8]. 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. [9], [10] 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. Shah et al. [11] demonstrated that composing multiple pre-trained models can enable navigation without language-annotated trajectory datasets. Similarly, Long et al. [12] 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

These works collectively illustrate the potential of combining vision-language understanding with spatial mapping and planning. More recently, Busch et al. [3] presented OneMap, a real-time, open-vocabulary mapping framework that constructs a belief map over CLIP-aligned features [1].

Building on these ideas, our method integrates a VLM-based semantic reasoning module with a real-time spatial-semantic map to produce grounded, executable navigation plans. Our work focuses on decomposing complex spatial queries into simpler subgoals that can be individually queried against a semantic belief map, while employing VLMs to validate candidate objects.

B. Policy Learning for Robot Navigation

Learning-based approaches [13], [14], [15], [16] have made progress in robot navigation by predicting actions directly from raw sensor data. [17] proposes to decompose navigation into hierarchical chain-of-thought reasoning steps and incorporate confidence-aware learning to handle noisy observations and uncertain semantic associations. However, it relies on fine-tuning a VLM using human demonstration trajectories and does not explicitly address spatial reasoning for object navigation tasks.

Fu et al. [18] introduced Scene-LLM, which integrates both global scene-level and local egocentric 3D representations by spatially downsampling point clouds into voxel grids while preserving fine-grained spatial details. Such hybrid representations allow agents to interpret both large-scale layouts and small-scale object arrangements. Another language-conditioned approach [19], LeLaN, learns object navigation policies from unlabeled, action-free egocentric data by leveraging vision-language and robotic foundation models to autonomously label diverse real-world indoor and outdoor environments, enabling scalable training of policies that can follow natural language instructions. [20] proposes trajectory diffusion for object goal navigation that learns a distribution over future action sequences conditioned on the current observation and goal to enable temporally coherent, multi-step planning. PIRLNav [21] introduces a two-stage object goal navigation method that combines behavior cloning on human demonstrations with reinforcement learning finetuning to learn robust navigation policies. [22], [23] extensively explore learning implicit neural representations to encode semantic information for object navigation.

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 [24] (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 [1] 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 use the uncertainty-aware semantic belief mapping system of OneMap [3]. OneMap leverages patch-level image embeddings that are produced by encoding the RGB frames I_t using the backbone of a CLIP-aligned segmentation model [25][1] and grounds them spatially using depth frames D_t and corresponding poses. Querying the resulting semantic map with a text embedding yields a top-down grid S of similarity values.

While OneMap 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 LVLMs 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.

The 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}, incorporating proximity spatial relationships (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 OneMap's integrated navigation module (Fig. 2 (d)). 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.

Altogether, this method enables to exploit the capabilities of modern LVLMs to decompose natural language into basic navigation targets and to robustly assess navigation success. The queryable semantic representation of the physical environment allows utilizing previous observations to provide effective guidance to the robot.

The following sections detail each component of our approach.

C. Decomposing Natural Language Queries

To make use of our map, we need to extract a set of relevant objects and locations for which the map can be queried. To this end, 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, relationships like "in", "on top", "near", etc. satisfy this, but "not in", "far from", do not satisfy this. 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.

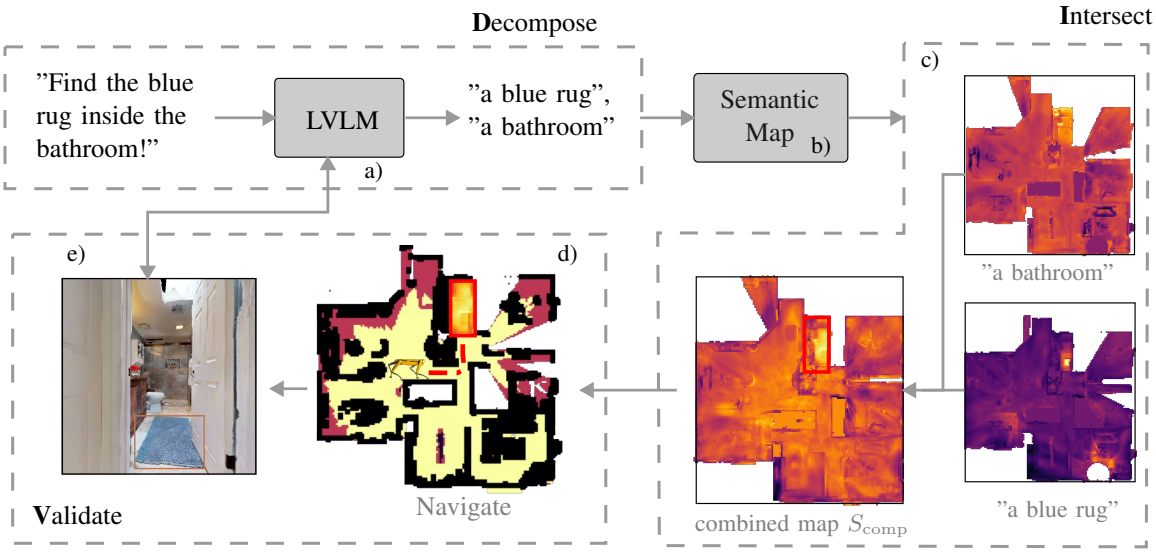

Fig. 2: Overview of DIV-Nav. (1) We **D**ecompose spatially-constrained instructions into object-level queries on a semantic map (a-b); (2) Compute **I**ntersections across similarity maps to find navigation regions where objects co-exist (c-d); (3) **V**alidate high-similarity regions by aproaching the region, and via LVLM reasoning to confirm individual objects and their spatial relationships (e).

D. Querying the Map

The goal of this module is to identify regions in the map that are likely to contain our extracted set of queries Q.

We follow [3] to query the map and start by encoding each query $q_i \in Q$ using the CLIP [1] text encoder to obtain an embedding $q_{\text{CLIP},i} \in \mathbb{R}^f$. We then compute the cosine similarity with the features of each map cell and obtain a query-specific similarity map $S_i(x, y)$ per query i.

To identify regions where all targets are most likely to be present, we are interested in the intersection of probable regions for all objects. Inspired by ConceptFusion [26], we combine the different similarity responses. But different from [26], we do not threshold the similarity values and choose to formulate a continuous value intersection score. We compute it as

$$S_{\text{int}} = \min_i S_i(x, y). \tag{1}$$

We do this for two reasons: 1) thresholding in 2D-map-space is difficult to tune; and 2) we are primarily interested in guidance for search, rather boolean labels.

This resulting map allows to infer regions where all targets are likely located close to each other (e.g., where both "the blue rug" and "the bathroom"" are present). We found that this operation is sufficient to capture relationships like *"next to"*, even though this does not necessarily correspond to an intersection. We attribute this to the fact that the mapped features are generally somewhat "blurred" spatially, in part due to OneMap's feature blurring procedure, which it performs during mapping to account for depth sensing noise and feature extraction uncertainty.

At certain stages of the search, some targets might not have been observed yet (e.g. we might have seen the bathroom, but not yet the sink in the bathroom). To still allow for meaningful guidance, we incorporate a maximum term to obtain the final similarity map

$$S_{\text{comb}}(x, y) = \alpha \cdot S_{\text{int}}(x, y) + (1 - \alpha) \cdot \max_i S_i(x, y), \tag{2}$$

where the weight α is a parameter. In practice, we select α to clearly prioritize S_{int}. An example of the natural language decomposition and the resulting queried similarity maps is depicted in Fig. 2

Note that for simple navigation goals, the list of targets is simply $\hat{T}$ if the LVLM cannot extract meaningful higher-level concepts, and the resulting similarity map is thus just $S_{\hat{T}}$.

E. Navigation and Validation

We use the combined similarity map S_{comb} to guide the robot to regions that are most likely to contain the target object described by the natural language query. Our navigation approach builds upon the OneMap navigation approach [3]. We adapt it to work with our combined similarity map that integrates multiple target queries. Besides the mapped features, OneMap maintains an estimate of uncertainty, and natively provides several derived binary maps that track the robot's exploration state:

- **Observed Map** $\mathcal{O}$: Regions that have been observed by any sensor update
- **Semantically Explored Map** $\mathcal{E}$: Areas where the map uncertainty falls below a threshold, indicating sufficiently reliable semantic features.
- **Searched Map** $\mathcal{C}$: Areas where the uncertainty falls below a threshold for the current query, reset when a new query begins.

- **Navigable Map** $\mathcal{N}$: Obstacle-free regions available for path planning.

We identify frontiers on the boundary between the semantically explored map $\mathcal{E}$ and the observed map $\mathcal{O}$. However, instead of scoring these frontiers using single-object similarity maps as in the original OneMap approach, we score them using our combined similarity map $S_{\text{comb}}(x, y)$. For each frontier, we compute the maximum similarity score $\max(S_{\text{comb}}(x, y))$ in reachable cells within $\mathcal{O} - \mathcal{E}$ (observed but not semantically explored regions). This guides exploration toward areas where all components of the spatial query are likely to co-exist.

We also cluster regions with high $S_{\text{comb}}(x, y)$ values within the semantically explored but not searched map ($\mathcal{E} - \mathcal{C}$). These represent previously observed areas that warrant revisiting under the current query context. Each cluster is scored by its maximum similarity value.

We greedily select the highest-scoring navigation goal from either frontier-based or cluster-based candidates. Path planning uses A* on the navigable map $\mathcal{N}$ to compute collision-free trajectories.

During navigation, we employ object detection with consensus filtering, requiring both detector activation and high $S_{\text{comb}}(x, y)$ values (top 5-percentile) for positive identification. Upon successful detection, the robot approaches the target location, which we obtain by projecting the detection into 3D space using the depth image. Moreover, we extend the validation step by prompting the LVLM to confirm that 1) the primary target is present and 2) the overall natural language prompt is satisfied. If the LVLM confirms that the conditions are satisfied, we approach the object within $0.5\,\mathrm{m}$ and terminate the search. We reset the searched map, and receive a new natural language query if available.

IV. Experimental Setup

We evaluate our method for multi-object navigation tasks given in natural language in the MultiON Challenge 2024 benchmark [27], as well as through real-robot navigation experiments. We use Phi-4-multimodal-instruct [24] as our LVLM model. For object detection, see Sec. III-E, we follow OneMap and use YOLOv7 [28] for MS-COCO [29] classes, and an open-set detector Yolo-World [28] otherwise.

Multi-Object Navigation in HSSD: The MultiON 2024 challenge extends the traditional object navigation framework presented in [30] 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) [31]. 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 $0.5\,\mathrm{m}$ 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 $0.25\,\mathrm{m}$, 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 [32] for odometry. 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.

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 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 [33] and IntelliGO Labs [27] which were trained for the task using RL. Since we utilize OneMap's semantic map and large parts of its navigation framework [3], we include OneMap as an additional baseline to evaluate our contributions in an isolated manner. Since OneMap can only handle simple object queries, we couple it with our VLM reasoning module and provide it with the primary target only, see III-C.

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 [33], IntelliGO Labs (the winning submission from the 2024 challenge) [27], and OneMap [3] provided with the primary target.

Approach	SRAT ↑	Pr ↑	SR ↑	FP (↓)	TNF (↓)
MOPA [33]	0.05	0.02	0.00	-	-
IntelliGO Labs [27]	0.23	0.10	0.03	-	-
OneMap [3]	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 OneMap, 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, OneMap 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: We observed that Vision-language models show reduced effectiveness on the rather simple graphics used in the benchmark compared to real-world images and (3) Highly challenging queries: The benchmark contains numerous highly challenging queries such as ”Find the freestanding bath/shower mixer” and ”Find the cream chair by Joanna Gaines”. Lastly, we found that 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

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 OneMap on real-world navigation tasks.

(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.

Approach	SR / # Found (↑)	FP (↓)	TNF (↓)
OneMap[3]	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).

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 OneMap baseline (and is statistically significant, 95% Clopper-Pearson bounds). We further report the false positive rate (FP), which shows that OneMap 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, OneMap 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 OneMap 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. Lastly, as a result of the LVLM validation step not confirming object detections, we note that DIV-Nav still misses some objects

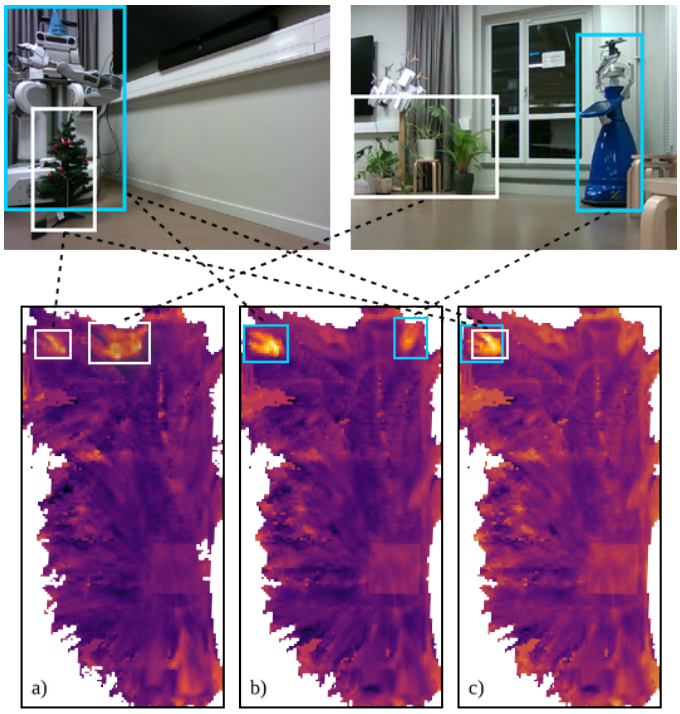

Fig. 4: Data of a real world experiment: (top) RGB frames, (a) similarity map for query *"plant"* , (b) similarity map for query *"robot"* (c) and the resulting map of value intersection scores.

entirely and terminates the search, leading to 11.11% of not-found cases.

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.

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 leveraging LVLMs to decompose spatial queries into simpler semantic components, relate them to the semantic map, 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 should be found relative to other objects.

When evaluated on tasks that require spatial relationship reasoning, our system achieves a 30% success rate per attempted target on the MultiON benchmark, outperforming existing methods (23%) that lack this spatial reasoning capability. This performance difference demonstrates that our proposed semantic intersection approach effectively helps when searching for targets with spatial relationships. In real-world deployments on a Boston Dynamics Spot robot, our system successfully completes 88% of multi-object navigation tasks with spatial constraints, compared to 53% for the baseline without spatial reasoning.

However, we find several limitations that point to possible future research directions. First, the semantic mapping on which we rely struggles with small or heavily occluded objects due to limitations in the CLIP feature space and limited map resolution, which potentially could be addressed by incorporating other additional representations. Second, VLM validation occasionally fails even when the semantic map correctly identifies target locations, indicating opportunities for more robust visual verification methods or confidence-calibrated validation strategies. Third, our current approach primarily handles proximity-based spatial relationships, generalizing to more complex spatial reasoning (e.g., "far from," "between," directional relationships) would broaden the system's applicability.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

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

[11] 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.

[12] 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.

[13] Y. Zhang, Z. Ma, J. Li, Y. Qiao, Z. Wang, J. Chai, Q. Wu, M. Bansal, and P. Kordjamshidi, "Vision-and-language navigation today and tomorrow: A survey in the era of foundation models," *arXiv preprint arXiv:2407.07035*, 2024.

[14] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, vol. 8, no. 79, p. eadf6991, 2023.

[15] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 63–70.

[16] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," *arXiv preprint arXiv:2306.14846*, 2023.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] ——, "Teaching agents how to map: Spatial reasoning for multi-object navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1725–1732.

[24] Microsoft, :, A. Abouelenin, A. Ashfaq, A. Atkinson, H. Awadalla, N. Bach, J. Bao, A. Benhaim, M. Cai, V. Chaudhary, C. Chen, D. Chen, D. Chen, J. Chen, W. Chen, Y.-C. Chen, Y. ling Chen, Q. Dai, X. Dai, R. Fan, M. Gao, M. Gao, A. Garg, A. Goswami, J. Hao, A. Hendy, Y. Hu, X. Jin, M. Khademi, D. Kim, Y. J. Kim, G. Lee, J. Li, Y. Li, C. Liang, X. Lin, Z. Lin, M. Liu, Y. Liu, G. Lopez, C. Luo, P. Madan, V. Mazalov, A. Mitra, A. Mousavi, A. Nguyen, J. Pan, D. Perez-Becker, J. Platin, T. Portet, K. Qiu, B. Ren, L. Ren, S. Roy, N. Shang, Y. Shen, S. Singhal, S. Som, X. Song, T. Sych, P. Vaddamanu, S. Wang, Y. Wang, Z. Wang, H. Wu, H. Xu, W. Xu, Y. Yang, Z. Yang, D. Yu, I. Zabir, J. Zhang, L. L. Zhang, Y. Zhang, and X. Zhou, "Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras," 2025. [Online]. Available: https://arxiv.org/abs/2503.01743

[25] 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

[26] 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.

[27] (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/

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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

[33] 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.