2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Language-Driven Navigation Strategy Integrating Semantic Maps and Large Language Models

Zhengjun Zhong[1], Ying He[1], Pengteng Li[1], Fei Yu[2], and Fei Ma[3]

***Abstract*— Accurate perception of semantic and spatial information is crucial for robots performing language-driven navigation tasks. Existing approaches utilize visual-language models to extract semantic information from the environment and construct maps. However, constrained by the generalization and accuracy of these models themselves, the constructed maps may not be accurate and comprehensive, thereby affecting the accuracy of navigation tasks. Inspired by foundational models' outstanding classification and segmentation capabilities, this study introduces a semantic map constructed using foundational models. We leverage a foundational model to semantically segment objects in the robot's video stream and fuse semantics onto the map. Furthermore, this map is used in conjunction with large language models (LLMs) that receive natural language instructions to complete the navigation task. A substantial number of experiments in a simulated environment demonstrate that our method outperforms existing ones in language-driven navigation tasks.**

I. Introduction

A very basic task for a robot is to navigate from an initial point to a specified point. The robot encounters many challenges in performing semantically-driven tasks, which are related to its ability to complete tasks based on natural language instructions from humans. Obviously, a robot that performs tasks according to human instructions is more economically and socially valuable than one that only performs tasks according to preset procedures [1], [2], [3]. Therefore, we aim for the robot to understand natural language commands such as "find a painting" and make corresponding actions in an indoor scene. To achieve this goal, it is crucial for the robot to have a profound understanding of the semantic information present in its environment as well as the commands given by humans.

Recently, some approaches employ visual-language models to encode the environment into visual features and utilize text-visual alignment to query regions of interest within the environment. CoW [4] integrates CLIP [5] with exploration algorithms to search for specified objects, while VLMap [6] directly encodes the environment into feature maps point by point using LSeg [7]. These methods exhibit impressive performance in language-driven navigation tasks and possess the capability of being open-vocabulary queryable. However, they also have certain limitations. The classification accuracy of the CLIP is relatively low, and the generalization performance of LSeg is relatively poor, which may affect the accuracy and generalization of scene representations, consequently impacting navigation accuracy. Therefore, we hope to enable robots to perform language-driven navigation tasks more accurately in more categories of scenes as shown in Fig. 1.

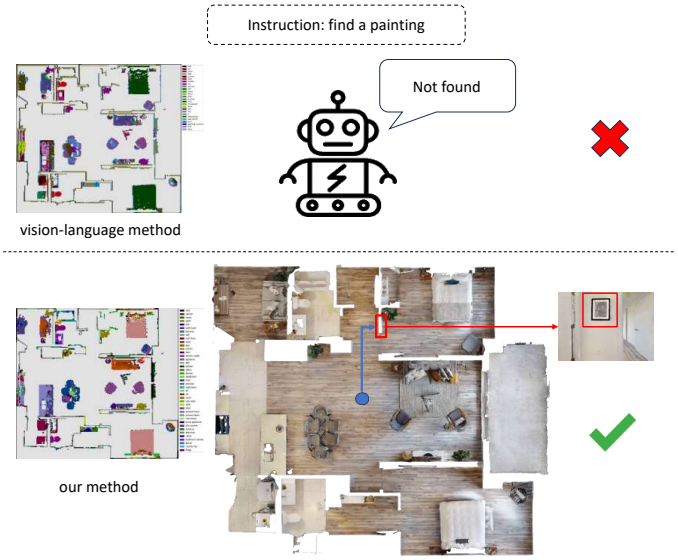

Fig. 1. The maps we construct are more comprehensive and accurate, and can better complete language-driven navigation tasks.

In order to construct a more accurate and comprehensive scene representation. To be specific, this work proposes a semantic map that combines Recognize Anything Model (RAM) [8] and Segment Anything Model (SAM) [9]. Specifically, we instruct the robot to explore the environment, fully gathering information about it. The robot labels and scores the collected RGB frames via RAM, and segments them via SAM to obtain semantic masks. These semantic masks are then merged with the point cloud generated from the corresponding depth images to create a semantic point cloud. The semantic map is obtained by filtering this semantic point cloud and projecting it from top to bottom. After obtaining the semantic map, LLMs are used to receive natural language instructions and generate executable codes for the robot. The robot utilizes the semantic map to execute these codes and complete navigation tasks.

In summary, our contributions are as follows:

[1]Z. Zhong, Y. He, and P. Li are with the College of Computer Science and Software Engineering, Shenzhen University, P.R. China 2210273070@email.szu.edu.cn; heying@szu.edu.cn; 2110276192@email.szu.edu.cn
[2]F. Yu is with the College of Computer Science and Software Engineering, Shenzhen University, P.R. China, and also with Carleton University, Canada yufei@szu.edu.cn
[3]F. Ma is with the the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), P.R. China mafei@gml.ac.cn
[2]F. Yu is the corresponding author.

- To construct a more accurate and comprehensive scene representation, we introduce a semantic mapping framework that integrates foundational models capable of extracting environmental semantics more comprehensively and accurately.
- For executing language-driven navigation tasks, we incorporate the semantic map into a LLM-based navigation framework. This integration is achieved by constructing a local API library, which enables LLMs to generate executable codes for the robot.
- We conduct a substantial number of experiments in a simulated environment to validate the efficacy of our approach. The experimental results demonstrate that our method outperforms the existing method.

II. Related Work

For language-driven navigation tasks, previous work has proposed numerous methods for constructing maps and navigation schemes. These methods leverage various modalities like visual perception, semantic segmentation, and depth estimation to obtain spatial representations and structured spatial information. The proposed navigation schemes span from classical path planning algorithms to incorporating language instructions into the decision process. Overall, these previous works laid a rich technical foundation, exploring the integration of different maps and navigation strategies to address the challenges of language-guided navigation in complex environments.

A. Map Representation For Navigation.

Sensing the environment through map representations is crucial for navigation tasks. Early research predominantly focuses on spatial navigation, emphasizing solely the spatial information of the environment, and proposed various map representations for specific application scenarios. The occupancy map [10], [11], [12] indicates whether a point is occupied, the cost map [13], [14] displays the cost of traversing each area, and the topological map [15], [16], [17] describes the environment's structure and connections. In recent years, to meet the demands of more advanced navigation tasks, researchers start incorporating semantic information into maps. This integration allows robots to interpret and interact with their surroundings in a more human-like manner. Works [18], [19], [20] use visual models to construct top-down semantic maps for modeling navigation environments. Additionally, some research enhances traditional spatial maps by adding object nodes [21], [22], [23] or object-centric scene maps [24], [25]. These methods achieve success in semantic-driven navigation tasks. However, traditional object detection and semantic segmentation models cover a relatively limited range of categories, which can lead to degraded performance in scenes with larger categories or necessitate finetuning the model when encountering a new scene. EmbodiedScan [26] proposes a multimodal, ego-centric 3D scene understanding. ConceptGraphs [27], an open-vocabulary graph-structured representation for 3D scenes, is built by leveraging 2D foundation models and fusing their output to 3D by multiview association. More recently, studies [4], [6], [28] employ visual language models to encode environments, enabling object querying in the environment using open vocabulary. For instance, achieves target navigation based on zero-shot language understanding by integrating the saliency map, based on CLIP, with traditional exploration methods. Similarly, VLMap combines pre-trained visual language model features with the geometric reconstruction of scenes. However, they construct less accurate and comprehensive representations of the scene. It motivates us to utilize models with more accurate and comprehensive classification for the reconstruction of the environment to be more accurate and comprehensive.

B. Language-Driven Navigation.

Language-driven navigation [29], [30] garner increased attention in the embodied AI domain. This task, which requires seamless integration of natural language understanding, visual perception, and decision-making, emerge as a challenging yet promising research direction. Some research adopts an end-to-end framework [29], [31], [32], simplifying the process by directly extracting navigation strategies from visual signals. This integrated approach combines perception and decision-making into a single model. While this simplifies the structure, it also reveals a strong dependLence on extensive data and a limited ability to generalize. Conversely, other studies prefer a modular design [28], [6], [33], aiming to enhance system adaptability and flexibility by separating perception, planning, and execution. This phased approach aids in mitigating complexity and enhances the model's generality across different environments. Recently, hybrid approaches [34], [35], which merge the perceptual advantages of deep learning with the stability and interpretability of traditional navigation strategies, begin to gain traction. Our approach uses a similarly distributed approach to make the system more flexible and improve generality in different environments.

III. Method

In order to enhance the robot's ability to perform language-driven navigation tasks more accurately in scenes, we propose a more accurate and comprehensive semantic map conbine RAM and SAM. Firstly, we detail the construction of a semantic map (Section III-A). Next, we explain how to localize objects within the map (Section III-B). Finally, we describe the integration of the semantic map with LLMs into our navigation framework (Section III-C). The overall pipeline of our approach is visualized in Fig. 2.

A. Constructing a Semantic Map

In our framework, the most important thing is to construct a global semantic map that effectively integrates object semantics with spatial information. We obtain segmentation masks of objects from RGB images using existing models and derive the point cloud of these objects from depth images. By merging these masks with the corresponding point cloud, we create a semantic point cloud. This semantic

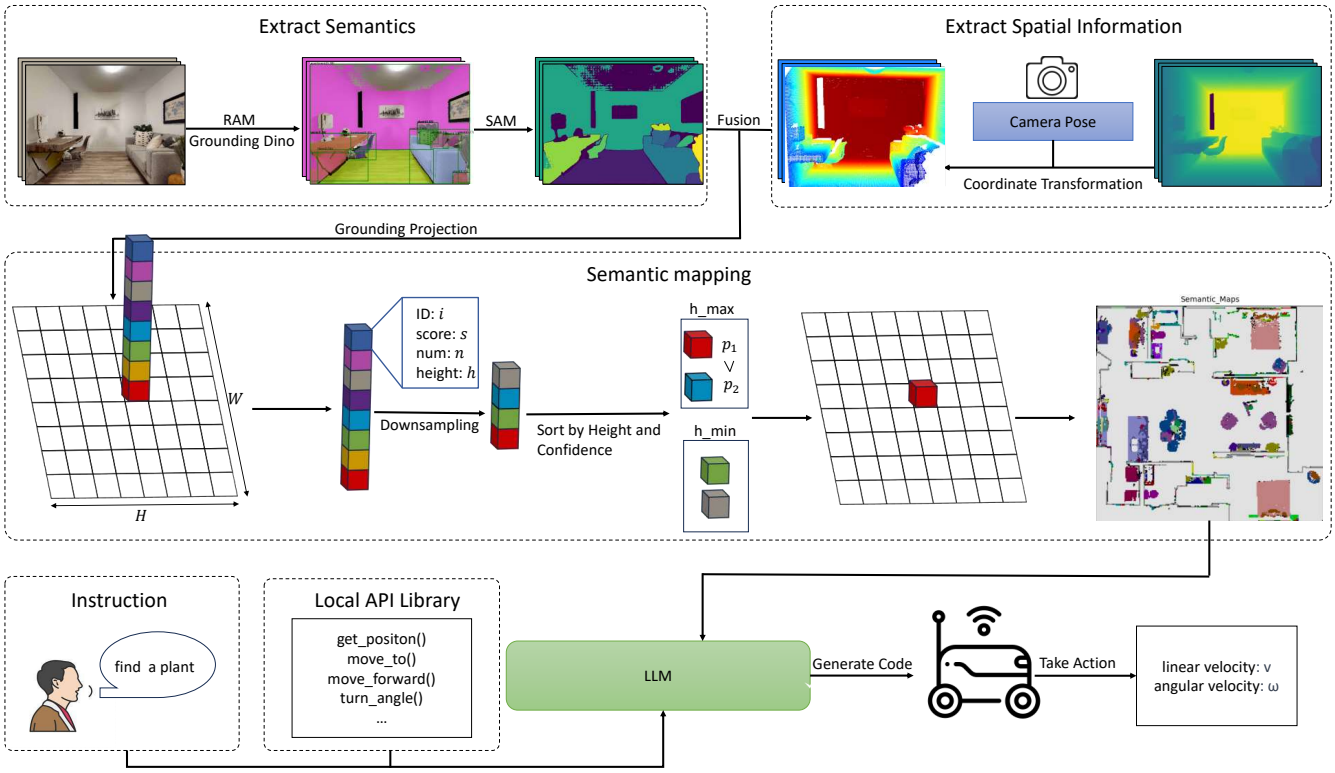

Fig. 2. The overall pipeline. At the core of this method is the construction of a semantic map through RAM and SAM. LLMs then receive instructions and generate codes within a local API library. The robot executes these codes, utilizing the semantic map to effectively navigate and complete the assigned tasks.

point cloud is then filtered and projected downwards onto the ground to achieve our desired semantic map. In our work, we combine RAM, Grounding DINO [36] and SAM to get the semantic mask of the image.

We define the semantic map $M \in R^{H \times W}$, H and W represent the height and width, respectively, of a grid map viewed from a top-down perspective. Each grid cell contains a unique identifier i representing an object label. The robot is equipped with both RGB and depth cameras. We direct the robot to conduct a comprehensive exploration of the environment, during which it collects RGB-D frames and records its current pose information. The total number of data sequences is represented by N. We then extract spatial and semantic information from the collected data. For accurate identification and localization of objects in the image, we use RAM, Grounding Dino and SAM to extract the object mask in the RGB frame. And a matrix A of the same size as the image is uesd to record the corresponding ID i and score s of the objects in the mask as follows:

$$A(x_f, y_f) = (i, s) \tag{1}$$

where x_f and y_f represent the positions of pixels in the image, $x_f \in (0, W')$, $y_f \in (0, H')$, W' and H' represent the height and width of image.

At the same time, all the depth pixels $d = (x_f, y_f)$ in the depth map are converted to point P_k in the current camera coordinate system. The point $P_w = (x, y, h)$ in the world coordinate system is then calculated through the transformation matrix as follows:

$$P_k = Q(d)K^{-1}\tilde{d} \tag{2}$$

$$P_w = T_{Wk}P_k \tag{3}$$

where $\tilde{d} = (x_f, y_f, 1)$, K is the intrinsic matrix of the depth camera, $Q(d) \in R$ is the depth value of the pixel d. T_{Wk} is the transformation from the world coordinate frame to the k-th camera frame.

After obtaining the semantic information from the RGB frame and the spatial information from the depth map, we combine the two tightly. We match the object mask with the depth map on a pixel-by-pixel basis. For each pixel in the same position, we integrate the A with the P_w corresponding to d to get the semantic point P'_w. And P'_w is mapped to the corresponding grid cell on the grid map M to obtain P_g which can be formulated as:

$$P'_w = P_w \oplus A(x_f, y_f) = (x, y, h, i, s) \tag{4}$$

$$x_{\mathcal{M}} = \lfloor \frac{W}{2} + \frac{x}{a} + 0.5 \rfloor, y_{\mathcal{M}} = \lfloor \frac{H}{2} - \frac{y}{a} + 0.5 \rfloor \tag{5}$$

$$P_g = (x_{\mathcal{M}}, y_{\mathcal{M}}, h, i, s) \tag{6}$$

where $x_{\mathcal{M}}$ and $y_{\mathcal{M}}$ represent the locations on the grid map

corresponding to x and y in the world coordinate system, respectively, and a is the size of one grid cell on the map.

When processing RGB-D streams, it is common for multiple semantic points to be projected onto the same grid cell location on the map. To ensure the most accurate labels are retained, we take into account both the frequency of label occurrences and their respective scores for filtering. Specifically, we first count the total frequency N_i and the average score S_i for each label during the semantic segmentation of all RGB frames. We consider labels with an extremely low frequency of occurrence throughout the entire perception process, or those with both low frequency and low average scores, to be unreliable. These labels are recorded in a list $i \in D$ as follows:

$$D = \begin{cases} D \cup \{i\}, & N_i < \alpha \\ D \cup \{i\}, & N_i < \beta \, and \, S_i < \gamma \\ D, & \text{otherwise} \end{cases} \tag{7}$$

where α, β and γ indicate the threshold values.

Then the labels in the grid cell are filtered according to D. After filtering the labels as a whole, we further filter the labels for each grid cell. While processing the data stream, we also accumulate the data of P_g and denote it as g. For example, when label c is in the location $(x_{\mathcal{M}}, y_{\mathcal{M}})$ of the grid cell:

$$\mathbf{n} = \sum_{k=1}^{N} 1_{[i=c]}, \mathbf{s} = \sum_{k=1}^{N} s_k^c, \mathbf{h} = \sum_{k=1}^{N} h_k^c \tag{8}$$

$$g(x_{\mathcal{M}}, y_{\mathcal{M}}) = (c, \mathbf{n}, \mathbf{s}, \mathbf{h}) \tag{9}$$

where s_k^c, h_k^c respectively represent the score and height of label c appearing for the k-th time in the same grid cell $(x_{\mathcal{M}}, y_{\mathcal{M}})$. **n**, **s**, and **h** respectively represent the total occurrence, cumulative score, and total height of a label in a grid cell.

After all frames are processed, we calculate the average score $\bar{s}$ and average height $\bar{h}$ of each label in the grid cell. We posit that a label detected multiple times at the same location with a high average score is considered reliable. Therefore, we calculate the confidence level p of each label in the grid cell using the following formula:

$$\bar{s} = \frac{\mathbf{s}}{\mathbf{n}}, \bar{h} = \frac{\mathbf{h}}{\mathbf{n}} \tag{10}$$

$$p = \begin{cases} w_1 \dfrac{\mathbf{n}}{10} + w_2 \bar{s}, & \mathbf{n} < 10 \\ w_1 \log_{10}(\mathbf{n}) + w_2 \bar{s}, & \mathbf{n} >= 10 \end{cases} \tag{11}$$

where w_1 and w_2 indicate the weights.

Sometimes, there are different objects at different heights in the same grid cell, such as a "stool" under a "table". To layer objects, labels in a grid cell are grouped based on their average height $\bar{h}$, with labels having similar $\bar{h}$ being grouped together. In each group, only the label with the highest confidence level p is retained. These selected labels are then sorted in descending order by their height. The label with ID i at the top of this hierarchy is then embedded in the grid map M to build a top-down semantic grid map, represented as $M(x_{\mathcal{M}}, y_{\mathcal{M}}) = i$. The complete process for constructing the semantic map is detailed in Algorithm 1, illustrating each step in the construction of the map's semantic representation.

Algorithm 1 Semantic Map Construction

Input:
I_{RGB}^k is the RGB image
I_{D}^k is the depth image
P_{c}^k is the camera pose
Parameter:
N is the total number of data in sequences
a is the size of one grid cell in map
α, β, γ is the threshold values for label deletion
w_1, w_2 is the weights used to calculate the confidence
H, W are the length and width of the map, respectively
Output:
$M \in R^{H \times W}$ is the semantic map

for $k = 0$ to N **do**
  $i, s \leftarrow Forward(I_{\text{RGB}}^k)$
  $A = (i, s)$
  world coordinate points: $P_w \leftarrow T(I_{\text{D}}^k, P_{\text{c}}^k)$
  semantic points in the grid map: $P_g \leftarrow C(P_w, A, a)$
  add up the data in each grid cell: $g \leftarrow g + P_g$
  the number of times the label i appears: $N_i = N_i + 1$
  the total score of label i: $S_i = S_i + s$
end for
delete some labels: $g \leftarrow \text{Del}(S_i, N_i, \alpha, \beta, \gamma)$
for $x_{\mathcal{M}}$ in W **do**
  for $y_{\mathcal{M}}$ in H **do**
    $i \leftarrow \text{filter}(g(x_{\mathcal{M}}, y_{\mathcal{M}}), w_1, w_2)$
    $M(x_{\mathcal{M}}, y_{\mathcal{M}}) = i$
  end for
end for
return M

B. Localizing Object

Next, we need to locate the location of a specific object in the constructed map. To effectively identify and locate objects within the semantic map, we construct a semantic dictionary $L = \{\text{lable} : i\}$ that records the names of all objects and their corresponding unique identifiers i. When searching for the location of a specific object, we first find its unique identifier i in the dictionary L. Subsequently, we search the semantic map M for all grid cells equal to i, considering these grid cells as potential locations of the object. We utilize a density-based clustering algorithm to group these grid cells into clusters, and subsequently compute the central coordinates p_{center} of the object on the map. Specifically, to address situations where the queried object names and the semantic labels recorded in the dictionary are similar but not identical, such as "TV" versus "television" and "couch" versus "sofa", we integrate LLMs into our system. LLMs excel at interpreting and correlating semantically similar terms, thereby facilitating the identification of the

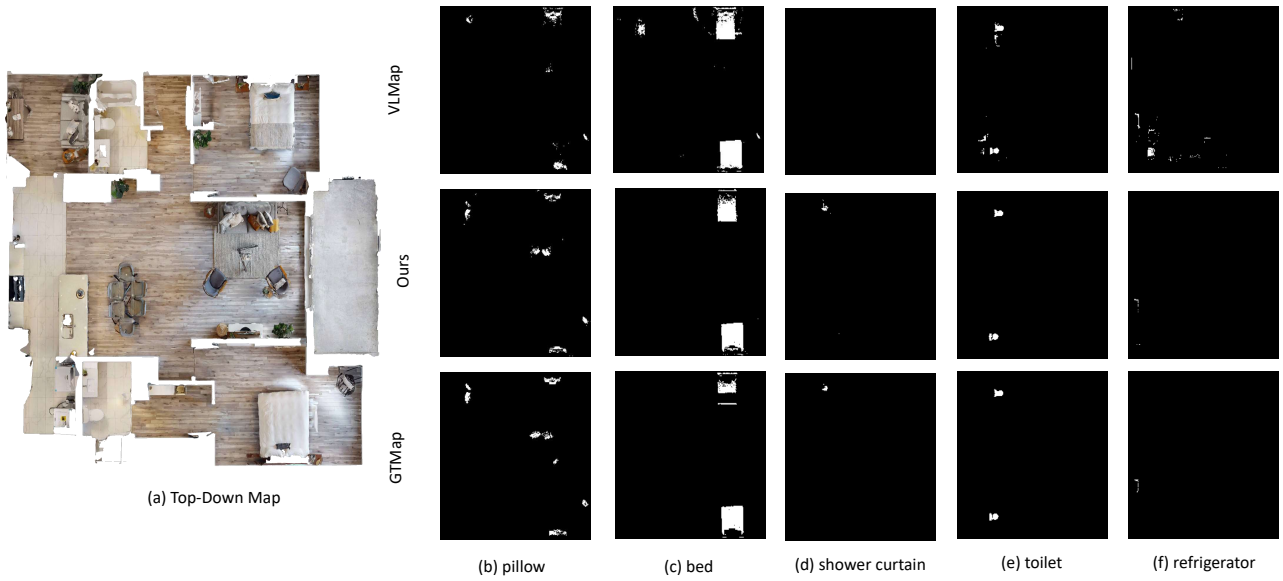

Fig. 3. Object mask on map. (a) shows the top-down map of the scene 1. (b), (c), (d), (e), and (f) illustrate the mask representations for the object types "pillow," "bed," "shower curtain," "toilet," and "refrigerator," respectively. The top row displays masks from VLMap, the middle row from our method, and the bottom row shows the ground truth masks from GTMap.

closest matching label in the dictionary for any given query. If an exact semantic match for the query object is not found in the dictionary, we will then inquire which object in the dictionary best matches the target word through LLMs.

C. Navigation from Language

In this section, we describe how to navigate based on given natural language instructions. We prepare some pre-written executable codes, encapsulated within the local API library, for the robot to call upon. we craft a prompt which explains the functionality of each API and pairs them with examples of natural language commands. During task execution, new natural language instructions are embedded into this prompt as input for the LLMs. The LLMs then generate the corresponding codes for these instructions. Subsequently, the robot accomplishes the task by executing these generated codes. For specific execution, we use the global plus local planning method and calculate the corresponding specific linear velocity v and angular velocity ω of the robot in each execution cycle.

IV. Experiments

A. Experimental Setups

In our experiment, we utilize Habitat simulator [37] and Matterport3D dataset [38] to evaluate navigation tasks. The Habitat Simulator is an open source, efficient simulation platform specifically designed for studying object navigation and visual navigation tasks. The Matterport3D dataset comprises a large number of high-quality 3D spatial scanning data, covering various types of indoor environments. To create the map in Habitat, we collect 8,280 RGB-D frames in 5 different scenes, recording each frame's camera pose, split mask, label, and its score. Additionally, to make the simulation more closely resemble real-world situations, we allow the robot to carry out continuous control rather than discrete operations in the Habitat continuous environment. We calculate the angular velocity ω and linear velocity v of the robot in each execution cycle. For the parameters in section III-A, we set $\alpha = 10$, $\beta = 40$, $\gamma = 0.36$, $w_1 = 0.35$ and $w_2 = 1$.

Baseline, we evaluate our approach against the baseline method VLMap. VLMap constructs a global semantic map by integrating pre-trained visual language model features into the geometric reconstruction of the scene. This method encodes spatial information and semantic information into a unified representation, which is convenient for navigation tasks. By exploiting the synergy between spatial geometry and language semantics, VLMap effectively interprets and follows complex navigation instructions through LLMs, transforming them into a series of spatial waypoints.

Ground truth map. In order to compare with the upper limit of the system, we call habitat's interface to obtain truth masks while collecting RGB-D frames. In the process of Map construction (Section III-A), the ID corresponding to the real label is added to the global point cloud, and the Ground Truth Map is obtained by projecting the point cloud from top to bottom.

B. Accuracy of Semantic Map

In the construction of semantic maps, ensuring the accuracy of semantic information is crucial. Misunderstandings of semantics at spatial locations may lead directly to errors in path planning, thereby preventing the navigation task from being executed correctly. For instance, if the

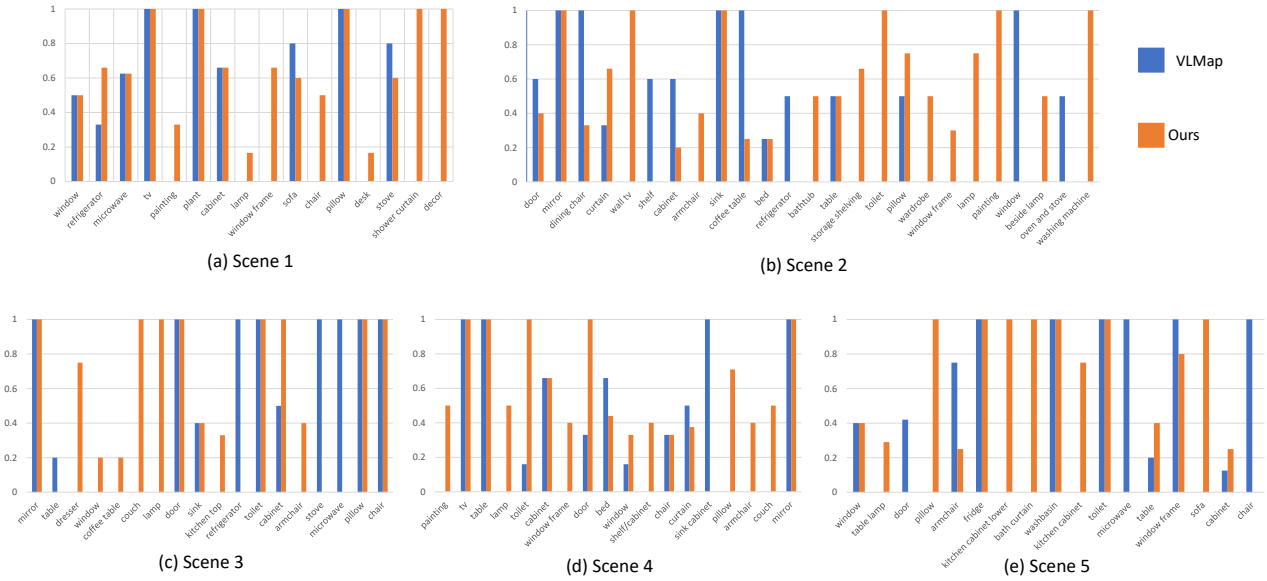

Fig. 4. (a), (b), (c), (d), and (e) represent the navigation success rate of each label in each of the five scenes (for simplicity, labels that fail in both algorithms are not included). The blue one on the left of each graph represents VLMap, while the orange one on the right represents our method.

TABLE I

SEMANTIC ACCURACY OF MAP IN DIFFERENT SCENES. IN GENERAL, OUR MAPS ARE CAPABLE OF LOCATING OBJECTS WITH GREATER ACCURACY THAN VLMAP.

Scene	VLMap			Ours		
	Macc	MIoU	FWIoU	Macc	MIoU	FWIoU
1	20.6	14.3	82.0	**29.7**	**20.9**	**83.2**
2	16.1	8.1	**82.0**	**17.6**	**8.9**	80.4
3	28.2	17.1	**77.2**	**32.5**	**19.3**	76.0
4	19.9	12.5	**83.7**	**26.8**	**18.7**	80.8
5	23.5	13.0	85.6	**27.9**	**18.0**	**85.8**

TABLE II

SUCCESS RATE (%) OF NAVIGATION IN DIFFERENT SCENES. OUR METHOD PERFORMS FAVORABLY OVER VLMAP.

Method	Scene				
	1	2	3	4	5
VLMap	30	32	31	34	35
Ours	**44**	**33**	**37**	**51**	**44**
GTMap	74	83	84	90	94

actual semantic object at location A is "book", but it is incorrectly identified and labeled as "apple", then executing the instruction "find an apple" will mislead the navigation system to the wrong destination. However, discrepancies in similar meanings (e.g., "sofa" and "couch", "chair" and "armchair") are less problematic. To accurately evaluate the effectiveness of semantic maps, we use popular semantic segmentation metrics for a quantitative assessment of their semantic accuracy.

We calculate Mean accuracy (Macc), Mean IoU (MIoU) and Frequency Weighted IoU (FWIoU) based on GTMap. To compute semantic masks for VLMap, we use the ground truth label set of GTMap as the input language list, and then convert this list into text features. We calculate the similarity between the features stored in VLMap and the text features, and assign the label with the highest similarity to each grid cell. For our semantic map, When comparing our label set with the ground truth label set of GTMap, the grid cell remains unchanged for same labels. If different labels are encountered, we employ LLMs to find the most similar label within the ground truth label set. If a similar label is found, we update the corresponding grid cell label to that similar label (e.g., "television" is converted to "tv"); if no suitable match is found, the grid cell is marked as "void" (represented by i=0). We calculate the three metrics across five different scenes and summarized the results in Table. I. It can be seen that our method surpasses VLMap in Macc and MIoU across all 5 scenes and exhibits better FWIoU in two scenes. These results demonstrate that our map will be more accurate and comprehensive for target location.

To visually demonstrate the differences between our method and VLMap in localizing objects on map, we visualize the masks of five types of objects within Scene 1 across three maps. The visual comparisons are presented in Fig. 3. It is apparent that VLMap produces a higher number of incorrect predictions (Fig. 3(c), Fig. 3(e), and Fig. 3(f)), which can lead to incorrect planning, or some objects cannot be located (Fig. 3(d)), making planning impossible. Both of these cases will reduce the success rate of navigation.

C. Navigation

To evaluate the performance of our entire framework, we direct a robot in the simulated environment to perform a

navigation task. Specifically, we provide a straightforward natural language instruction "find {object}," which prompts the LLMs to interpret the target and invoke the corresponding function from our pre-defined local API library to execute the task. We consider the navigation task successfully executed when the robot's final stopping position is within one meter of the correct target. Before starting the experiment, we conduct multiple target navigations on GTMap, selecting those labels that successfully navigate to form the candidate label set for this scene. This set serves as the query vocabulary list for VLMap, and we transform the features in VLMap into specific labels. Then, when the experiment officially starts, we conduct 100 experiments in each scene. In each experiment, the robot's starting position is randomly designated within a scene, and then one label is randomly selected from the candidate label set as the {object}. We navigate on all three maps using the method described in Section III-C. We calculate the success rate of task, as shown in Table. II. Compared to VLMap, our method demonstrates a higher success rate across all five scenes, indicating its enhanced performance and effectiveness. Combined with the visual analysis in Fig. 3, it is evident that the improvement in success rate is due to the higher accuracy of our method in target localization.

We also observe the navigation success rate for different labels during the experimental process, comparing the performance between VLMap and our method, as shown in Fig. 4. The data analysis indicates that our method possesses a much broader ability to recognize and navigate to different labels. As depicted in Fig. 4(a), VLMap's navigation success rate is zero on 7 labels, but our method can locate and navigate to these same labels with a relatively high success rate, demonstrating the advantages of our method in terms of processing range. Additionally, the same labels exhibit varying performance across different scenes. For example, in our method, the label "microwave" achieves a higher success rate in Scene 1, but in Scenes 3 and 5, the success rate drops to zero. This inconsistency might be due to the algorithm's tendency to misidentify the "microwave" label as other labels. Furthermore, by examining Fig. 4(a), Fig. 4(b), Fig. 4(c), Fig. 4(d), and Fig. 4(e), it is evident that VLMap's navigation success rate for labels associated with "lamp" is consistently zero. This may be because the visual language model used by VLMap does not contain the category of "lamp", making it impossible to accurately locate objects associated with "lamp" on the map. In contrast, our approach demonstrates superior performance in this regard, primarily due to the enhanced zero-shot capabilities of RAM, trained on a substantial volume of image-text pairs.

D. Ablation Study

To evaluate the effectiveness of our design, this section conduct ablation experiments. Specifically, parameters are adjusted in scene 1, and a semantic map is reconstructed. The accuracy of the map is evaluate using the methods described in section IV-B, with the results shown in Table. III.

Firstly, regarding the downsampling process associated with α, β, and γ, it is evident that retaining labels with extremely low frequencies, as well as retaining labels with relatively lower frequencies and scores, both result in a decrease in map accuracy. This decline becomes more pronounced when both aspects are retained. The reason for this situation is that when the object is far away, it may be misjudged due to unclear visibility, or when the object is too close and its complete appearance cannot be captured, resulting in misjudgment and the generation of some incorrect labels. Removing these misjudged labels can improve map accuracy.

TABLE III
ABLATION STUDIES

Parameters				Macc	MIoU	FWIoU
α	(β ,γ)	w_1	w_2			
0	(40,0.36)	0.35	1	27.0	19.3	82.4
10	(0,0)	0.35	1	27.1	20.0	82.7
0	(0,0)	0.35	1	25.8	18.8	82.4
10	(40,0.36)	0	1	24.6	18.2	81.9
10	(40,0.36)	0.35	0	28.2	20.4	82.5
10	(40,0.36)	0.35	1	**29.7**	**20.9**	**83.2**

Next, in the calculation of label confidence corresponding to w_1 and w_2, it is apparent that considering only the frequency of label occurrence or the score of the label individually fails to achieve optimal accuracy. At the same time, it can be observed that relying solely on the score for confidence calculation performs worse compared to relying solely on the frequency. This is because the scores assigned by Grounding Dino exhibit a certain bias, leading to certain labels being more likely to receive higher scores. Therefore, to retain more accurate labels, the optimal method is to comprehensively consider both the frequency and the score.

V. CONCLUSION

We develop a more accurate and comprehensive semantic map by integrating RAM and SAM, and use LLMs to enable robots to complete navigation tasks based on natural language instructions. Experiments demonstrate that our method excels in object localization and recognition, leading to better navigation task completion. However, our algorithm struggles with real-time data processing, dynamic map updates, and lacks map correction mechanisms during navigation failures. Future work will optimize algorithms for faster data processing, dynamic map updates, and use LLMs for navigation failure corrections. We also plan to test our framework in real-world scenarios to understand its practical applications and limitations.

VI. ACKNOWLEDGMENT

This work is supported in part by Shenzhen Science and Technology Program under Grant ZDSYS20220527171400002, the National Natural Science Foundation of China (NSFC) under Grants 62271324, 62231020 and 62371309, and the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011979.

References

[1] M. A. Goodrich, A. C. Schultz, *et al.*, "Human–Robot Interaction: A Survey," *Foundations and Trends® in Human–Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2008.

[2] T. B. Sheridan, "Human–Robot Interaction: Status and Challenges," *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[3] R. R. Murphy, T. Nomura, A. Billard, and J. L. Burke, "Human–Robot Interaction," *IEEE robotics & automation magazine*, vol. 17, no. 2, pp. 85–89, 2010.

[4] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 171–23 181.

[5] 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 *Proceedings of International conference on machine learning*, 2021, pp. 8748–8763.

[6] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual Language Maps for Robot Navigation," in *Proceedings of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 608–10 615.

[7] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[8] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu, *et al.*, "Recognize Anything: A Strong Image Tagging Model," *arXiv preprint arXiv:2306.03514*, 2023.

[9] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment Anything," *arXiv preprint arXiv:2304.02643*, 2023.

[10] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Mutual Information-Based Exploration on Continuous Occupancy Maps," in *Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 6086–6092.

[11] K. D. Katyal, A. Polevoy, J. Moore, C. Knuth, and K. M. Popek, "High-Speed Robot Navigation using Predicted Occupancy Maps," in *Proceedings of 2021 IEEE International Conference on Robotics and Automation*, 2021, pp. 5476–5482.

[12] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, "Occupancy Anticipation for Efficient Exploration and Navigation," in *Proceedings of Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, 2020, pp. 400–418.

[13] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered Costmaps for Context-Sensitive Navigation," in *Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 709–715.

[14] S. R. Avutu, D. Bhatia, and B. V. Reddy, "Voice Control Module for Low Cost Local-Map Navigation Based Intelligent Wheelchair," in *Proceedings of 2017 IEEE 7th International Advance Computing Conference*, 2017, pp. 609–613.

[15] E. Beeching, J. Dibangoye, O. Simonin, and C. Wolf, "Learning to Plan with Uncertain Topological Maps," in *Proceedings of European Conference on Computer Vision*, 2020, pp. 473–490.

[16] K. Chen, J. K. Chen, J. Chuang, M. Vázquez, and S. Savarese, "Topological Planning With Transformers for Vision-and-Language Navigation," in *Proceedings of Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 276–11 286.

[17] X. Meng, N. Ratliff, Y. Xiang, and D. Fox, "Scaling Local Control to Large-Scale Topological Navigation," in *Proceedings of 2020 IEEE International Conference on Robotics and Automation*, 2020, pp. 672–678.

[18] G. Georgakis, K. Schmeckpeper, K. Wanchoo, S. Dan, E. Miltsakaki, D. Roth, and K. Daniilidis, "Cross-Modal Map Learning for Vision and Language Navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 460–15 470.

[19] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "Gridmm: Grid Memory Map for Vision-and-Language Navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 625–15 636.

[20] 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.

[21] M. Zins, G. Simon, and M.-O. Berger, "OA-SLAM: Leveraging Objects for Camera Relocalization in Visual SLAM," in *Proceedings of 2022 IEEE International Symposium on Mixed and Augmented Reality*, 2022, pp. 720–728.

[22] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[23] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352–1359.

[24] 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.

[25] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "SceneGraphFusion: Incremental 3D Scene Graph Prediction From RGB-D Sequences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7515–7525.

[26] T. Wang, X. Mao, C. Zhu, R. Xu, R. Lyu, P. Li, X. Chen, W. Zhang, K. Chen, T. Xue, *et al.*, "Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai," *arXiv preprint arXiv:2312.16170*, 2023.

[27] 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.

[28] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary Queryable Scene Representations for Real World Planning," in *Proceedings of 2023 IEEE International Conference on Robotics and Automation*, 2023, pp. 11 509–11 522.

[29] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the Nav-Graph: Vision-and-Language Navigation in Continuous Environments," in *Proceedings of Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, 2020, pp. 104–120.

[30] 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.

[31] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-Follower Models for Vision-and-Language Navigation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[32] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets, "Waypoint Models for Instruction-Guided Navigation in Continuous Environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 162–15 171.

[33] S. Chen, X. Chen, C. Zhang, M. Li, G. Yu, H. Fei, H. Zhu, J. Fan, and T. Chen, "Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning," *arXiv preprint arXiv:2311.18651*, 2023.

[34] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "PONI: Potential Functions for ObjectGoal Navigation With Interaction-Free Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 890–18 900.

[35] P. Chen, D. Ji, K. Lin, R. Zeng, T. Li, M. Tan, and C. Gan, "Weakly-Supervised Multi-Granularity Map Learning for Vision-and-Language Navigation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 149–38 161, 2022.

[36] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding Dino: Marrying Dino with Grounded Pre-Training for Open-Set Object Detection," *arXiv preprint arXiv:2303.05499*, 2023.

[37] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, *et al.*, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9339–9347.

[38] 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," *arXiv preprint arXiv:1709.06158*, 2017.