

L-SNI: A Language-Driven Semantic Navigation System for Inspection Tasks

Jiawang Ma^{*}, Weichen Guo^{*}, Xuan Wu, Zinan Zhuang, Rongxiang Zeng, Yongliang Shi and Gan Ma[†]

Abstract—For inspection robots to achieve generalizability, stability, and ease of use, it is crucial that they understand natural language commands and navigate accurately to specified target objects. We propose L-SNI, a semantic navigation system adapted for inspection tasks, offering generalizability, robust stability, and practical ease of use. In the perception phase, L-SNI constructs a precise geometric depth map of the environment using LiDAR, while RGB images are employed to extract object categories, which are then combined with depth data to generate a semantic map. To enable the large language model (LLM) to interpret the environment, L-SNI encodes the 3D semantic map into a plain text representation. During single-task execution, L-SNI decodes human commands into inspection primitives using an LLM constrained by system initial prompts. These inspection primitives guide the robot's low-level planner for task execution. To address the challenge of traditional 3D LiDAR localization and navigation systems in accurately positioning the robot around target objects during inspection tasks, we propose a target cost gradient to assist in optimizing the robot's target point selection and attitude control in maps with semantic information. Upon reaching the target, L-SNI uses a visual language model (VLM) to describe the scene, which is simplified by the LLM into a user-friendly response. Through testing on 18 indoor scenes from the Matterport 3D dataset, L-SNI achieves a 46.9% improvement in Success Rate (SR) and a 58.3% increase in Success weighted by Path Length (SPL) over existing state-of-the-art (SOTA) solutions, while also demonstrating superior target image understanding. Moreover, it can be easily deployed on real-world robots without complex initialization.

I. INTRODUCTION

Inspection robots have been applied in various indoor scenarios, and to expand their use while reducing the complexity of operation, these systems should be language-driven. The robot must be capable of navigating to specified objects within the environment based on arbitrary human instructions and accurately interpreting visual information.

In developing such robots, foundational models that replicate human reasoning processes could be particularly valuable. Existing systems such as GOAT [1] and VLFM [2] have integrated these models to support semantic navigation, allowing robots to explore and navigate to semantic objects in unfamiliar environments. However, these systems fall short of meeting the demands of practical applications in terms of navigation efficiency and success rate, primarily due to three factors: 1) these methods prioritize open exploration and place heavy emphasis on extracting semantic information, but they often overlook the high spatial accuracy needed for inspection tasks. As a result, they do not ensure that the robot can accurately reach the target point and perform precise localization and

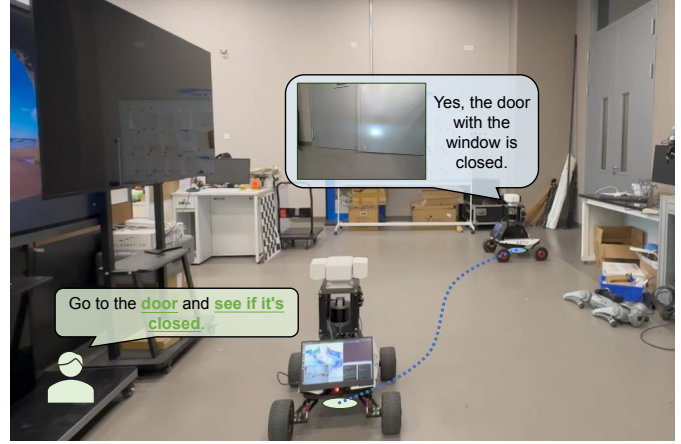


Fig. 1: L-SNI processes human language instructions, extracting target object instances using LLM and semantic priors, and converts these instructions into navigation goal points. Upon reaching the designated target point, it provides interpretative feedback based on the target object within its field of view that aligns with the original human instructions.

navigation in complex environments. 2) Navigation methods based on deep neural networks or reinforcement learning require extensive training resources and large data sets, which poses significant limitations in practical applications. 3) These approaches do not incorporate downstream task planning for navigation and human interaction in inspection scenarios.

To address the needs for generalizability, ease of use, and stability in inspection robots operating on limited computing platforms, we introduce L-SNI, a real-time deployable semantic navigation system tailored for inspection tasks. L-SNI integrates the strengths of LLM and VLM to enhance semantic reasoning and scene understanding, thereby improving the generalizability of the robot and the ease of use while maintaining the high stability of the traditional perception-prediction-planning navigation framework. As depicted in Fig. 2, L-SNI comprises two main components: perception and task planning. In the perception phase, L-SNI combines stable depth estimation from 3D LiDAR point clouds with rich semantic information from RGB images to construct a 3D semantic map of the environment. To facilitate understanding by the LLM, we employ a textualization process, encoding these 3D semantic maps into plain text representations. In the task planning phase, L-SNI uses an initial system prompt along with the textualized semantic map to enable the LLM to interpret human language commands and convert them into executable inspection primitives. These primitives are then used by a low-level planner for task execution. In addition, to achieve an accurate target position for inspection, we introduce a target cost gradient. After reaching the target pose, the robot generates and simplifies a description of the scene,

^{*}Equal contribution, [†]Corresponding author

J. Ma, W. Guo, X. Wu, R. Zeng and G. Ma are with the Sino-German College of Intelligent Manufacturing, Shenzhen Technology University, China. Y. Shi is with Tsinghua University, China. Corresponding author: Gan Ma(magan@sztu.edu.cn)

which is then used for feedback. We conducted comparative experiments using four state-of-the-art (SOTA) methods: CLIP on Wheels (CoW) [3], ZSON [4], and VLFM [2], along with a supervised method, PONI [5], in 18 indoor scenes from the Matterport 3D [6] dataset. The results show that L-SNI outperforms existing methods in terms of navigation Success Rate (SR) and Success weighted by Path Length (SPL) [7] for inspection tasks, also, only our method is equipped with target image understanding.

The key contributions of this paper include:

- A semantic navigation system designed for inspection tasks, characterized by high stability, generality, ease of use, and suitability for practical applications.
- A method for constructing 3D point cloud semantic maps using a pre-trained semantic segmentation network to provide rich semantic information and stable depth estimation, then encoding the maps into plain text representations for direct LLM understanding.
- A target point selection method based on the semantic object cost gradient, addressing the limitations of traditional 3D LiDAR localization and navigation systems in accurately identifying target points around semantic objects for inspection tasks.
- A zero-shot task planner and human-robot interaction method driven by a large multimodal model, enhancing the generalizability of inspection tasks in complex scenarios and reducing the difficulty of robot use.
- A novel evaluation metric, Goal Description Accuracy (GDA), is introduced to assess the robot's ability to accurately recognize and describe target object images. This metric addresses the existing evaluation system's insufficient focus on semantic accuracy in inspection tasks.

II. RELATED WORK

A. Semantic Mapping

RGB-based semantic mapping methods [8]–[12] leverage rich color and texture information for object and scene distinction. However, these methods are limited by the absence of depth data and sensitivity to lighting. RGB-D methods [13]–[17] help mitigate these limitations. However, RGB-D approaches still struggle with stable depth estimation in large-scale environments.

Besides, with the development of neural networks, semantic maps can be built using only LiDAR. RangeNet++ [18] applies a 2D CNN to point clouds for segmentation, Pointnet++ [19] adds hierarchical structure on the top of Pointnet [20]. Many subsequent LiDAR-based methods [21]–[23] have used these two networks. Despite these advantages, relying solely on LiDAR for semantic segmentation requires extensive neural network training, and the range of recognizable objects is still limited.

To address these challenges, we propose fusing RGB data with LiDAR point clouds to generate semantic map. Rich semantic information can be obtained from RGB data, and LiDAR can provide stable depth estimation.

B. Semantic Navigation

Early semantic navigation methods [24], [25], relied heavily on deep reinforcement learning (DRL) and image-based inputs. Later developments [26], [27], aimed to reduce the dependence on explicit training signals, but they still operated without language comprehension. These approaches lacked the ability to understand scenes or process language commands.

In recent years, the development of LLMs like BERT [28] and GPT [29] have enabled more advanced multimodal semantic navigation, combining visual inputs with language processing, such amount of methods [30]–[34] are based on BERT. Approaches such as CoW [3], GOAT [1], ZSON [4] and VLFM [2] integrate object detection and language understanding, making semantic navigation can be driven by language. CoW [3] explores nearby boundaries until a target object is detected using CLIP [35] features or an open vocabulary object detector, then navigates directly to the identified target. GOAT [1] facilitates multimodal navigation by utilizing category labels, target images, and language descriptions, while continuously updating an instance-aware semantic memory through lifelong learning in real-world environments. ZSON [4], an open-vocabulary method, leverages CLIP to transfer zero-shot techniques from ImageNav tasks to ObjectNav tasks. It is trained on ImageNav, where images serve as targets instead of object names. VLFM [2] employs a pre-trained visual language model to extract semantic information from RGB images, creating a semantic value map that helps identify promising boundaries to explore and locate the target object. However, these methods require extensive training resources and large datasets, making them lack generalizability in data-scarce scenarios.

Our approach overcomes these limitations by integrating the potential of LLM for zero-shot high-level task planning with the strengths of traditional lower-level motion planning framework, resulting in a more general solution.

III. METHOD

A. System Overview

An overview is showed in Fig. 2. The system is divided into two main components: perception and task planner. In perception module, L-SNI takes RGB images and LiDAR point clouds as inputs. The RGB images are segmented using the YOLOv8 [36] model to obtain object masks, which are then associated with the LiDAR point cloud to generate semantic point clouds. The semantic point clouds are combined with Fast-LIO [37], where it is used to incrementally construct a semantic map. This 3D semantic map constructed in this method effectively combine the stable depth estimation from 3D LiDAR point clouds with the rich semantic information from RGB images. The semantic maps are encoded into a plain text representation for scene understanding by the LLM, which details the IDs, names, positions, and 2D size of objects in the environment. In the task planner module, the textualized semantic map generated by the perception module serves as input to the LLM for environmental understanding.

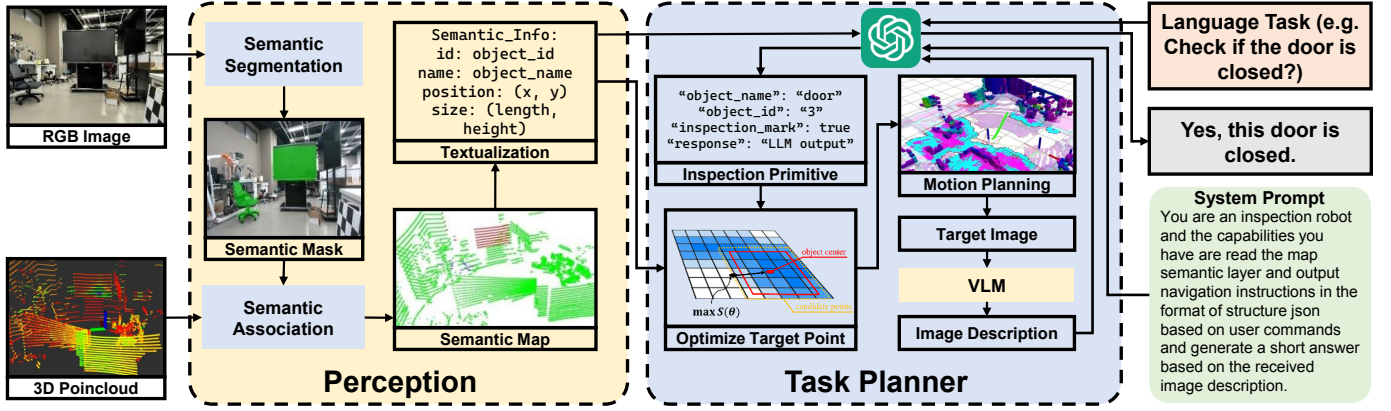


Fig. 2: The L-SNI system consists of two main components. The perception module processes RGB images and 3D LiDAR point clouds to generate a semantic map using a pre-trained semantic segmentation model. The semantic map are then encoded into plain text representation. In the task planning module, LLM takes system initialization prompts and textualized semantic map representation as input. It also translates human language commands into inspection primitives that can be executed by the robot, including navigation commands and invoking visual language models for image understanding. Ultimately, LLM generates anthropomorphic responses for the user.

Additionally, predefined system initial prompt and the user's language commands are provided to the LLM to facilitate task understanding and planning. During the task planning phase, the LLM interprets the user's language instructions to identify the target object and encodes these instructions into a specific format known as the inspection primitives (detailed in III-D). Subsequently, the system calculates the cost gradient around the target object (detailed in ??) to determine the optimal arrival point relative to the robot's current position. An executable trajectory is calculated using the search-based global planning algorithm and DWA [38] local planning algorithm to guide the robot to the target point, where it captures an image of the object, then the VLM generates a description of the image, which is often complex and not immediately interpretable by the user. Finally, the LLM processes the image description produced by the VLM, providing the user with a more human-friendly interpretation, tailored to the specific language task.

B. Semantic Mapping and Textualization

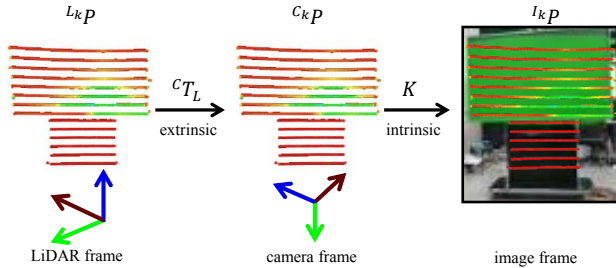


Fig. 3: Semantic mapping workflow. $L_k P$ are transformed into $C_k P$ via extrinsic C_{T_L} , and $C_k P$ are transformed into $I_k P$ via intrinsic K . $I_k P$ are then matched with the segmentation mask.

To generate a semantic map for the robot's environment, we integrate LiDAR and RGB data, applying semantic segmentation to enrich the global map with semantic information. This process ensures that the map not only provides spatial details

but also associates objects with meaningful labels, which is crucial for downstream navigation tasks. Details of this method are described in following paragraphs.

Given a frame of point clouds $L_k P$, where L_k is the LiDAR local frame at time k , $L_k P$ are first transformed from the LiDAR frame to the camera frame via the extrinsic matrix $C_{T_L} = ({}^C R_L, {}^C p_L)$, obtaining $C_k P$. Subsequently, the camera intrinsic matrix K is utilized to project the points from the camera frame to the image frame, obtaining $I_k P$.

Simultaneously, a pre-trained YOLOv8 model is applied to the image captured at time k to perform semantic segmentation, generating the corresponding segmentation mask. The points $I_k P$ are then matched with the segmentation mask. If any points fall within the segmentation mask, corresponding labels are assigned to the points in $L_k P$, then we combine Fast-LIO to generate a global map containing semantic information. Fig. 3 shows our semantic mapping workflow and Fig. 4 shows the semantic map.

Textualization of semantic map is necessary, since LLM cannot directly interpret 3D semantic maps. Considering potential errors in the extrinsic matrix C_{T_L} , some points may be misclassified. Thus, outliers must be filtered out. For points sharing the same label, assume there are m points in total. The distance between each point and its k nearest neighbors is calculated using a k-d tree, and the average distance D_i is computed for each point. The mean $\bar{D}$ and standard deviation D_{std} of all m points are then calculated. If $D_i > \bar{D} + D_{std}$, the point is considered an outlier, and its label is changed to -1, indicating that it does not belong to any identified object.

After filtering, points with the same labels are clustered. From these clusters, we extract the object names, the coordinates of the cluster centers along the x and y axes (i.e., position), as well as the lengths and widths of the clusters (i.e., size). This allows the position and size of the segmented objects in 2D space to be determined, and these parameters are then input into the LLM for use in navigation tasks.

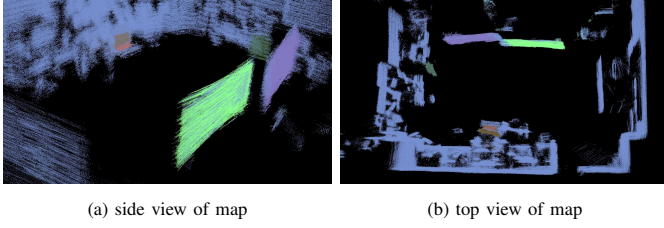


Fig. 4: (a) and (b) are semantic maps observed from different perspectives, where purple, bright green, dark green, orange, and brown point clouds represent the TV, whiteboard, chair, keyboard, and monitor, respectively

C. Target Point Selection

Conventional 3D LiDAR-based localization and navigation frameworks are not well suited for inspection tasks, particularly when precise viewpoint positioning is required. In top-down semantic maps, robots often fail to localize correctly around target objects, especially when the objects are occluded or placed adjacent to walls. For example, a robot may approach the wrong side of a wall-mounted object, yielding an invalid inspection pose despite spatial proximity.

To address this issue, we propose a Target Cost Gradient (TCG) method that leverages semantic geometry to compute an optimal navigation goal near the object. This method aims to select a candidate point near the object boundary that maximizes both accessibility and inspection suitability, by analyzing the spatial gradient of cost values and the robot's relative position.

1) *Problem Formulation*: Each object is represented by a rectangular region in 2D space, characterized by its center coordinates $\mathbf{p}_{\text{obj}} = (x_{\text{obj}}, y_{\text{obj}})$, and its width and height (w, h) . Let $\mathbf{p}_r = (x_r, y_r)$ be the current position of the robot. Our goal is to find an optimal approach point $\mathbf{p}^* \in \partial\mathcal{O}_d$, where $\partial\mathcal{O}_d$ denotes a safety-offset boundary around the object, such that the score function $S(\theta)$ is maximized over all direction angles $\theta \in [0, 2\pi)$:

$$\mathbf{p}^* = \arg \max_{\theta \in [0, 2\pi)} S(\theta) \quad (1)$$

The candidate point in direction θ is defined as:

$$\mathbf{p}_\theta = \mathbf{p}_{\text{obj}} + (r_{\mathcal{O}} + d_{\text{safe}}) \cdot \mathbf{u}_\theta \quad (2)$$

where $\mathbf{u}_\theta = (\cos \theta, \sin \theta)$ is a unit direction vector, and $r_{\mathcal{O}} = \max(\frac{w}{2}, \frac{h}{2})$ is the maximum half-length of the object.

2) *Cost Gradient Computation*: We define a spatial cost map function $C(\cdot)$ that quantifies the local cost (e.g., obstacle proximity, terrain difficulty) at any map location. The gradient in direction θ is computed as the normalized cost difference between the object center and the candidate point:

$$G(\theta) = \frac{C(\mathbf{p}_\theta) - C(\mathbf{p}_{\text{obj}})}{\|\mathbf{p}_\theta - \mathbf{p}_{\text{obj}}\| + \epsilon} \quad (3)$$

where ϵ is a small positive constant added to avoid division by zero. This term captures the relative change in environmental traversability between the object and the potential approach point.

3) *Distance Metric*: To incorporate efficiency into the navigation planning, we define a proximity term $D(\theta)$ representing the squared Euclidean distance from the robot to the candidate point:

$$D(\theta) = \|\mathbf{p}_\theta - \mathbf{p}_r\|^2 \quad (4)$$

Lower values indicate shorter travel distances, thus promoting energy efficiency and speed.

4) *Combined Scoring Function*: The final score function $S(\theta)$ combines cost gradient and distance via a weighted sum:

$$S(\theta) = w_g \cdot G(\theta) + w_d \cdot \frac{1}{D(\theta) + \delta} \quad (5)$$

where $w_g, w_d \in \mathbb{R}^+$ are tunable weights controlling the balance between safety (gradient-based accessibility) and efficiency (travel distance), and $\delta > 0$ prevents division by zero. By evaluating $S(\theta)$ across discrete angles θ , the system selects the direction yielding the highest score.

5) *Goal Pose and Orientation*: Once the optimal angle θ^* is identified, the robot's goal position is set to $\mathbf{p}_{\theta^*}$, and its orientation is aligned to face the object center:

$$\phi^* = \text{atan2}(y_{\text{obj}} - y_{\theta^*}, x_{\text{obj}} - x_{\theta^*}) \quad (6)$$

This ensures the robot not only arrives at a valid inspection location, but also faces the target object to maximize camera visibility and image interpretation accuracy.

The proposed method provides a lightweight yet effective approach to high-precision inspection positioning. It avoids extensive training and instead exploits geometric semantics and cost-based heuristics, improving robustness in complex, partially occluded indoor environments.

D. LLM as Zero-Shot Task Planner

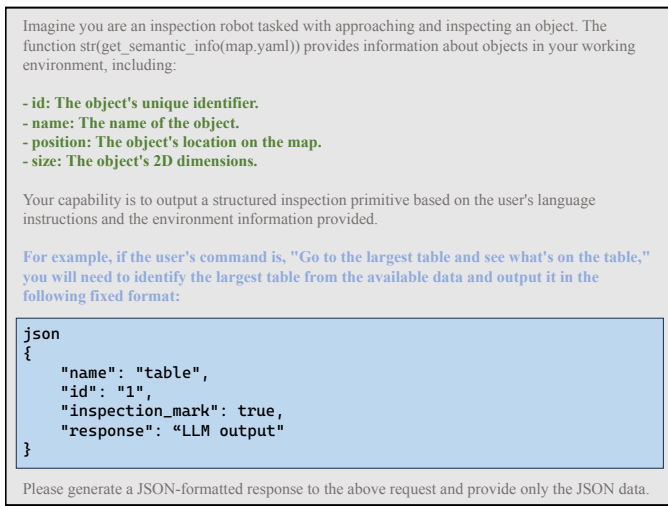
In order to reduce the difficulty of using the robot, we describe a high-level task planner for inspection tasks, given tasks described by natural language, such as

Go to the desk and see if the computer's on the desk.
Go to the door and see if it's closed.
Go to the whiteboard and see what's on it.

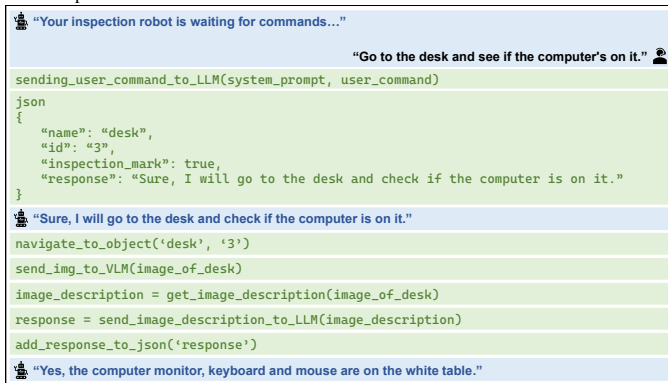
The robot will perform the appropriate tasks and give answers. Specifically, we first utilize the LLM to interpret and extract target objects for navigation from natural language commands, while also determining whether inspection tasks are required. The LLM generates structured inspection primitives that are comprehensible to the robot, enabling both navigation and the invocation of the VLM for image understanding. In this paper, we define the basic unit of a single inspection task as an "inspection primitive". The inspection primitive is a structured JSON-formatted statement that includes the target object's name, its ID, a flag indicating whether image comprehension is required, and the corresponding LLM response. The specific format is illustrated in the blue box in Fig. 5a. These generated inspection primitives can be seamlessly recognized and executed by the robot's built-in navigation system.

To enable the LLM to generate more accurate responses, we provide clear and concise task details along with contextual

descriptions in the system initial prompt. These contextual descriptions include: 1) environment information, where we encode the semantic map into text and supply it to the LLM via a parameter profile to foster an understanding of the environment; 2) solution examples, where we present the LLM with instances of how to complete inspection tasks, covering the full process from receiving user commands to generating responses. Furthermore, we impose structured, parametric constraints on the LLM's output to ensure the correct generation of inspection primitives that the robot can execute in real time. Fig. 5a depicts a system initial prompt of a robot completing an inspection task. Fig. 5b depicts the workflow of the robot in completing an inspection task guided by the LLM.



(a) The system initialization prompt consists of several components: gray text clearly outlines the robot's tasks and capabilities, green text describes the semantic information of the map after it has been converted into text, and blue text provides an example task for the robot. In this example, blue squares represent the strict structure required for the LLM output.



(b) The robot's internal workflow for completing an inspection task under LLM guidance begins with the LLM extracting target object information from user commands and matching it with semantic map data to generate an inspection primitive. The robot then navigates to the optimal point, captures an image, which the VLM processes into a description that the LLM simplifies and returns as a response.

Fig. 5: (a) Illustrates the initial system prompts provided to the robot. (b) Illustrates the robot's workflow for completing the inspection task, guided by the LLM.

IV. EXPERIMENTS

The primary objectives of our experiments are twofold: 1) to evaluate whether our approach demonstrates superior performance in navigating to spatial inspection targets specified by natural language commands and generating accurate interpretations of the visual scene, compared to existing methods; 2) to assess the feasibility of deploying L-SNI in real-world environments. To achieve these goals, we set up the experiments as follows.

A. Experimental Set Up

Basic Setup. The robot's radius and height were set to 0.3m and 1.2m respectively. A navigation task was considered successful if the robot's center was within 0.5m of the target object edge and oriented towards the object center ($\pm 30^\circ$ tolerance).

Datasets. We conducted experiments using the Matterport 3D [6] dataset, a comprehensive 3D scan of real-world environments, within the Gazebo simulator in ROS Noetic. Our approach was validated across 18 indoor scenes and 8 object categories derived from the Matterport [6] 3D dataset.

Evaluation Metrics. We conducted a comparative evaluation of all methods using three key metrics: Success Rate (SR), Success weighted by Path Length (SPL), and Goal Description Accuracy (GDA). Among these, SPL serves as a comprehensive measure for assessing the robot's navigation performance by combining task success rate with path efficiency. SPL not only evaluates whether the robot successfully reaches the target, but also measures the efficiency of its navigation by comparing the actual path taken with the shortest possible path. A value closer to 100% indicates higher system efficiency. GDA reflects the method's ability to comprehend the content within the target's field of view, a crucial metric for inspection tasks. GDA was computed by comparing VLM-generated descriptions with ground-truth annotations from 3 human evaluators.

Baselines. We evaluated our approach by comparing L-SNI against several state-of-the-art (SOTA) semantic navigation methods: CLIP on Wheels (CoW) [3], ZSON [4], and VLFM [2]. Additionally, we compare our approach with a supervised method, PONI [5], which constructs maps during navigation and employs task-specific policies trained for semantic reasoning to predict the locations of potential target objects for exploration.

B. Results

Table I provides a comparison of experimental result of the performance of our method against other approaches on the Matterport 3D [6] dataset. Blank cells indicate that certain methods were not evaluated on specific metrics for this dataset. The results demonstrate that our method surpasses the existing state-of-the-art (SOTA) approaches across all benchmark tests.

L-SNI outperformed existing zero-shot methods in baseline testing, achieving a 46.9% improvement in Success Rate (SR) and a 58.3% improvement in Success weighted by Path Length (SPL) compared to VLFM [2]. Relative to the

TABLE I: Comparison of experimental result

Methods	SR (%)	SPL (%)	GDA (%)
CoW [3]	7.4	3.7	-
ZSON [4]	15.3	4.8	-
PONI [5]	31.8	12.1	-
VLFM [2]	36.4	17.5	-
Ours w/o TCG	68.2	59.7	70.1
Ours	83.3	75.8	86.7

supervised method PONI [5], SR improved by 51.1% and SPL by 63.7%. Our approach demonstrates this advantage by leveraging the L-SNI perception module, which provides stable geometric information and rich semantic information about the environment. Additionally, we employ a motion planning algorithm based on explicit geometric rules and cost functions, ensuring the optimality and interpretability of the generated paths. In contrast, the compared methods prioritize open-ended scene exploration, leading to suboptimal stability and spatial accuracy for inspection tasks. Additionally, our approach demonstrates superior understanding of the target object, a capability not possessed by the existing methods.

Fig. 6 illustrates an example of a semantic navigation and inspection task. In this task, the robot interprets human language commands, identifies the target semantic category, computes the optimal navigation point goal, and subsequently analyzes the visual data within its field of view. The robot then provides feedback, offering a response that aligns with the user's requirements.

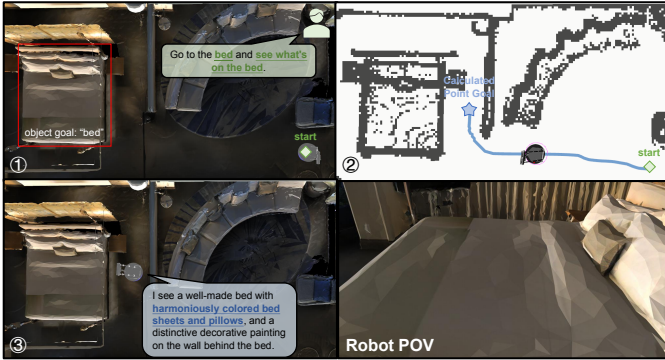


Fig. 6: The two figures on the left show a top-down view of the environment. The upper-right figure displays the obstacle map and navigation path used by the robot, while the lower-right figure shows the robot's observations upon reaching the goal point.

C. Real-World Deployment

As shown in Fig. 7, we deployed our approach on a Scout mini robot to assess the practical implementation of L-SNI for real-world target navigation and comprehension. The computational hardware is Intel NUC with an i5 2.1GHz CPU and 16GB memory. RGB input is provided by a RealSense D455 camera, while a 32-line LiDAR offers 3D point clouds data. For map creation, we employ an off-the-shelf LiDAR SLAM solution: Fast-LIO to estimate sensor positions. During localization, we use HDL-localization [39] to initialize the robot's position.

Platform

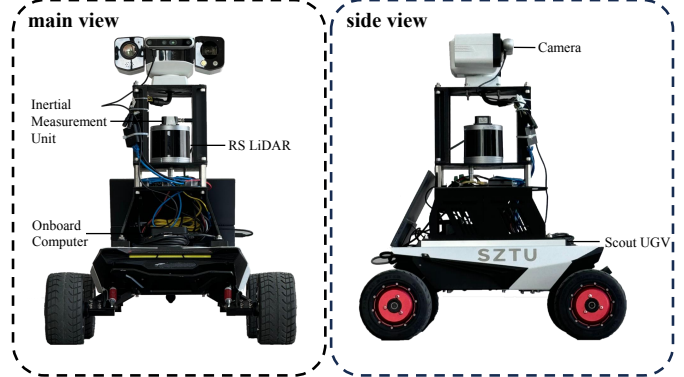


Fig. 7: The vehicle platform used for deploying L-SNI is equipped with an RS32-line LiDAR and a camera. Both the onboard computer and the LiDAR are powered by batteries mounted on the chassis.

L-SNI was tested in semantically rich indoor environments containing over 10 object categories. The system's navigation and goal understanding capabilities were evaluated using various language inspection commands, such as "Go to the robotic arm workspace to check.", "Go to the desk to see if the computer is on the desk." and "Go to check if the door is closed.". While the robot successfully completed all test tasks, we observed that the understanding of scenarios in repeated instruction tests was occasionally biased. This bias arises because LLM and VLM generate responses based solely on contextual cues from the current task and lack long-term memory capabilities, which can affect feedback results. Nevertheless, these results indicate that L-SNI is capable of performing open language-driven inspection tasks in real-world settings.

V. CONCLUSION

In this paper, we present the Language-Driven Semantic Navigation System for Inspection Tasks (L-SNI), a general, stable and user-friendly approach designed for inspection applications. L-SNI combines 3D LiDAR data and RGB observations to construct a semantic map. By integrating a LLM, the system enables a robot to efficiently navigate towards a target object based on human natural language commands and generate a concise image description of the target upon arrival. Our system demonstrates substantial improvements in task success, navigation efficiency, and stability, highlighting its potential for real-world deployment. Additionally, L-SNI requires minimal initialization, making it a practical solution for various inspection tasks. Future work will focus on dynamically updating semantic information in maps so that robots can achieve robust navigation even in changing scenarios.

REFERENCES

- [1] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavitha Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. Goat: Go to any thing. *arXiv preprint arXiv:2311.06430*, 2023.

- [2] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48, 2024.
- [3] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*, 3(4):7, 2022.
- [4] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multi-modal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022.
- [5] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18890–18900, 2022.
- [6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.
- [7] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7606–7623, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [8] Ziwei Liao, Yutong Hu, Jiadong Zhang, Xianyu Qi, Xiaoyu Zhang, and Wei Wang. So-slam: Semantic object slam with scale proportional and symmetrical texture constraints. *IEEE Robotics and Automation Letters*, 7(2):4008–4015, 2022.
- [9] Berta Bescos, Cesar Cadena, and José Neira. Empty cities: A dynamic-object-invariant space for visual slam. *IEEE Transactions on Robotics*, 37(2):433–451, 2021.
- [10] Yanmin Wu, Yunzhou Zhang, DeLong Zhu, Yonghui Feng, Sonya Coleman, and Dermot Kerr. Eao-slam: Monocular semi-dense object slam based on ensemble data association. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4966–4973, 2020.
- [11] Tong Qin, Tongqing Chen, Yilun Chen, and Qing Su. Avp-slam: Semantic visual mapping and localization for autonomous vehicles in the parking lot. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5939–5945, 2020.
- [12] Jingwen Wang, Martin Rünz, and Lourdes Agapito. Dsp-slam: Object oriented slam with deep shape priors. In *2021 International Conference on 3D Vision (3DV)*, pages 1362–1371, 2021.
- [13] Kaiqi Chen, Jialing Liu, Qinying Chen, Zhenhua Wang, and Jianhua Zhang. Accurate object association and pose updating for semantic slam. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):25169–25179, 2022.
- [14] Zhentian Qian, Kartik Patath, Jie Fu, and Jing Xiao. Semantic slam with autonomous object-level data association. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11203–11209, 2021.
- [15] Berta Bescos, Carlos Campos, Juan D. Tardós, and José Neira. Dynaslam ii: Tightly-coupled multi-object tracking and slam. *IEEE Robotics and Automation Letters*, 6(3):5191–5198, 2021.
- [16] Edgar Sucar, Kentaro Wada, and Andrew Davison. Nodeslam: Neural object descriptors for multi-view shape reconstruction. In *2020 International Conference on 3D Vision (3DV)*, pages 949–958, 2020.
- [17] Dongjiang Li, Xuesong Shi, Qiwei Long, Shenghui Liu, Wei Yang, Fangshi Wang, Qi Wei, and Fei Qiao. Dxslam: A robust and efficient visual slam system with deep features. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4958–4965, 2020.
- [18] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220, 2019.
- [19] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [20] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [21] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguère, Jens Behley, and Cyrill Stachniss. Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537, 2019.
- [22] Juncong Fei, Kunyu Peng, Philipp Heidenreich, Frank Bieder, and Christoph Stiller. Pillarsegnet: Pillar-based semantic grid map estimation using sparse lidar data. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 838–844, 2021.
- [23] Guibin Chen, Bosheng Wang, Xiaoliang Wang, Huanjun Deng, Bing Wang, and Shuo Zhang. Psf-lo: Parameterized semantic features based lidar odometry. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5056–5062, 2021.
- [24] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364, 2017.
- [25] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurobh Gupta. Neural topological slam for visual navigation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12872–12881, 2020.
- [26] Meera Hahn, Devendra Singh Chaplot, Shubham Tulsiani, Mustafa Mukadam, James M Rehg, and Abhinav Gupta. No rl, no simulation: Learning to navigate without navigating. *Advances in Neural Information Processing Systems*, 34:26661–26673, 2021.
- [27] Ziad Al-Halah, Santhosh K. Ramakrishnan, and Kristen Grauman. Zero experience required: Plug play modular transfer learning for semantic visual navigation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17010–17020, 2022.
- [28] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [29] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018.
- [30] Alessandro Suglia, Qiaozi Gao, Jesse Thomason, Govind Thattai, and Gaurav Sukhatme. Embodied bert: A transformer model for embodied, language-guided visual task completion. *arXiv preprint arXiv:2108.04927*, 2021.
- [31] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. *arXiv preprint arXiv:1909.02244*, 2019.
- [32] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.
- [33] Arjun Majumdar, Ayush Srivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 259–274. Springer, 2020.
- [34] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. *arXiv preprint arXiv:2011.13922*, 2020.
- [35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [36] Rejin Varghese and Sambath M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 1–6, 2024.
- [37] Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.
- [38] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1):23–33, 1997.
- [39] koide3. Hdl localization. https://github.com/koide3/hdl_localization.