SDA-LLM: Spatial DisAmbiguation via Multi-turn Vision-Language Dialogues for Robot Navigation

Kuan-Lin Chen, Tzu-Ti Wei, Ming-Lun Lee, Li-Tzu Yeh, Elaine Kao*, Yu-Chee Tseng, and Jen-Jee Chen

Abstract— **When users give natural language instructions to service robots, positional information is often referenced relative to objects in the environment rather than absolute coordinates. However, humans naturally use relative references. For example, in "Go to the chair and pick up empty bottles" , where the positional reference is the chair, ambiguity arises when multiple similar objects co-exist in the environment or when the robot's view is limited, resulting in multiple possible interpretations of the same command and affecting navigation decisions. To address this issue, we propose a two-level framework that integrates a large language model (LLM) and a vision-language model (VLM), allowing the robot to engage in multi-turn dialogues for spatial disambiguation. Our method first utilizes a VLM to map the semantic meanings of dialogues to a unique object ID in images and then further maps this object ID to a 3D depth map, enabling the robot to accurately determine its navigation target. To the best of our knowledge, this is the first work leveraging foundation models to address spatial ambiguity.**

Keywords: mapping and routing, large language model, multi-modal model, navigation, vision-language model

I. Introduction

With the advancements in sensors and computer vision, significant progress in robotics has been made in autonomous mapping [1], [2] and obstacle avoidance [3], [4], [5]. Machine learning techniques for LiDAR and cameras have been developed for better decision-making in dynamic environments [6], enabling more reliable and efficient navigation for robots in various fields. These advancements include simultaneous localization and mapping (SLAM) [7], [8], enhanced sensor fusion [9], and real-time object recognition [10], [11], which largely improve robots' navigation capabilities in complex environments.

On the other hand, recent advances in artificial intelligence have led to significant breakthroughs in large language models (LLM) and large multi-modal models (LMM), such as GPT [12] and BERT [13], which revolutionize natural language understanding and generation in various domains. These advances include generating coherent and contextually relevant text, understanding complex instructions, and even reasoning on language-based tasks. In particular, LMMs infer and align information from vision, language, and even sensory data [14], [15], [16], significantly moving forward human-to-robot and robot-to-physical world interactions.

Following the trend, an emerging direction is to apply language models to interact with robots or even program their behaviors. Given natural language instructions and visual observations, Vision-Language Navigation (VLN) [17] is designed to navigate a robot. By incorporating interactive human feedback, [18], [19], [20] extend VLN to allow clarification or additional information when the robot faces uncertainty, thereby improving the task completion probability. Building on this, Zero-Shot Object Navigation (ZSON) [21], [22] leverages the prior knowledge and reasoning capabilities of LLMs, allowing robots to navigate to open-vocabulary objects in unseen environments via natural language commands.

The above studies [17], [18], [19], [20], [21], [22] often assume that the spatial descriptions are uniquely identifiable or the target objects are visually distinct in type, color, or shape. However, humans are not good at describing precise absolute positions. The spatial information in natural language commands is often embedded in the referenced object in the environment. The positional ambiguity problem arises when multiple similar objects co-exist, resulting in multiple possible interpretations of the same command and affecting navigation decisions. Despite cutting-edge technologies in autonomous robots, there remains a spatial gap between human dialogues and robots' perceptions. For example, the commands

```
Grab the cup on the table.
Help me go next to the chair.
```

would confuse a robot when there are two cups/chairs meeting the criteria. Therefore, second commands like

```
Pick up the cup next to the vase.
Go to the chair facing the window.
```

may clarify the ambiguity. Then the robot can map the unique object on its LiDAR map, navigate to the right position, and perform its task.

This work aims to address the disambiguation problem within natural language when the position information is contextually dependent on an object in the environment. The objective is to differentiate objects, whenever necessary, by their spatial relationships with other objects or landmarks. We propose a two-level framework called SDA-LLM, where SDA means spatial disambiguation. The natural language commands may consist of time, position, and action components. We demonstrate how to disambiguate such commands by modern LLM and VLM through multiple dialogues and ultimately map human intentions to a target position in the environment. Fig. 1 outlines our framework. First, the *Level-1 Mapping* conducts 3D scanning and object detection on the space to obtain an accurate LiDAR map with all objects uniquely mapped to the LiDAR map. Also, a unique ID

The authors are with the College of Artificial Intelligence, National Yang Ming Chiao Tung University (NYCU), TAIWAN. *Elaine Kao worked as an intern at NYCU.

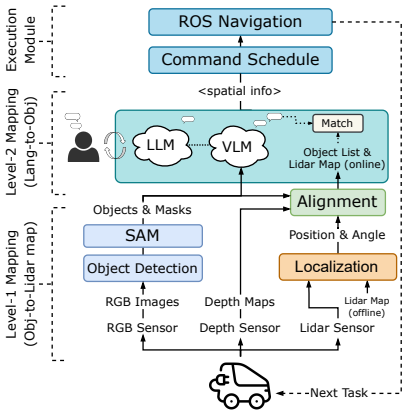


Fig. 1: The SDA-LLM disambiguation framework.

is assigned to each object. Second, the *Level-2 Mapping* leverages the power of LLM and VLM, which takes user's dialogues and photo snapshots from the robot's current position as inputs, to map the positional meanings in the dialogues to a unique object in photos. Spatial disambiguation is done by the VLM with the annotations of object IDs on these images as inputs. Then a precise target position can be issued to the *Execution Module*. Finally, the proposed method, SDA-LLM, effectively addresses the problem of spatial ambiguity and enables the robot to navigate to the designated target point.

Sec. II reviews some related work. We discuss the challenge of position disambiguation in Sec. III. Sec. IV presents our SDA-LLM framework. Experiment results are shown in Sec. V. Sec. VI concludes this paper.

II. Related Work

A. Vision-and-Language Navigation (VLN)

Vision-and-Language Navigation (VLN) has garnered increasing attention recently. Previous work has addressed data augmentation [23], [24], memory mechanisms [25], [26], and pre-training [27], [28] issues. These work mainly focused on effectively utilizing simulator data during training and the objectives were to enhance simulation performance. The recent development of LLMs has made training zero-shot VLN agents possible, thus enabling these agents to navigate in real-world environments. LM-Nav [29] utilized GPT-3 for navigation in real-world topological maps; however, it struggled to represent spatial relationships between objects, potentially leading to the loss of detailed information. In contrast, CoW [21] and VLMaps [16] used top-down semantic maps to model the navigation environment, allowing for more accurate representations of spatial relationships. However, these methods were limited by predefined semantic labels, restricting their applicability.

Different from the above methods, NavGPT [30] constructed a framework for translating visual scene semantics into prompts, allowing an LLM to directly execute VLN tasks. Since NavGPT relies solely on text descriptions, its performance in spatial interpretation was limited. Object disambiguation was addressed in [31] by identifying attributes and removing distractors through multi-round interactions. However, it assumed the existence of identifiable intrinsic attributes (e.g., color or pattern) of objects. As can be seen, spatial ambiguity caused by visually identical instances of a target object has not been explicitly addressed in previous works. Our work leverages the power of VLM to jointly process visual inputs and language cues and employs multi-turn dialogues to resolve spatial ambiguities in human commands, thereby improving the accuracy of VLN.

B. LLM and LMM

The breakthroughs in LLM have demonstrated broad and impressive capabilities across various domains, such as summarization, code generation, and task planning. Inspired by the fine-tuning of LLMs with instructions, researchers have made progress in integrating visual instruction fine-tuning, thereby enhancing VLMs' ability to follow instructions. Vision-language models like GPT-4V [12], LLaVA [32], and InstructBLIP [33] have excelled in responding to image content, detailed semantics, and image-to-text artistic creation. These advancements inspired us to leverage VLM for precise navigation.

III. Challenge of Position Disambiguation

Let us assume that a human and a robot conduct a sequence of k dialogues $D_1 \rightarrow R_1 \rightarrow D_2 \rightarrow R_2 \rightarrow \dots D_k \rightarrow R_k$. Through the conversations, the human assigns a mission (i.e., his intentions) to the robot, where $D_i, i = 1..k$, are the dialogues of the human and $R_i, i = 1..k$, are those of the robot. Our objective is to derive a conversation model f which produces the response R_i at the ith round

$$R_i = f(D_j, o_j | j = 1..i), \tag{1}$$

where o_j is the observation made by the robot after receiving dialogue D_j. In Eq. 1, if D_i contains no ambiguity, R_i can be "OK, I will perform the mission." Otherwise, R_i may ask for further clarification from the human.

With the advancement of multimodal models, R_i and D_i may encompass various types of data. In some cases, humans can efficiently resolve ambiguities by pointing to a target or clicking on a screen. For example, when a robot presents an image containing multiple candidate objects or locations, the user can directly click on the correct item without requiring additional language descriptions. However, in the following scenarios, language-based interaction may be more effective than direct selection:

- When there are too many similar objects in a scene, presenting all candidates may lead to decision paralysis. For example, if a space contains 500 chairs, the user may need to zoom in and check them one by one, which is time-consuming and error-prone. Contrarily, if the user only wants to find the chair with his denim jacket draped over it, a simple verbal description can quickly narrow down the search range and improve efficency.
- Multi-turn dialogue can serve as a "filter" to help identify candidate items and eliminate errors caused by lighting, occlusion, or motion blur. For example, if a

user is looking for a laptop with a bear sticker, it may be difficult to identify it from a photo due to angle or lighting conditions. Through conversation, the robot can assist the user in gradually narrowing down the possibilities. Similarly, if a user's wallet was on the living room table yesterday but is now in the bedroom, multi-turn dialogue can help filter out discrepancies in human memory and improve localization accuracy.

- In warehouse environments, where goods appear similar but have subtle differences, confirmation through dialogue is necessary. For example, in the instruction "send out the blue packaging box," the robot can ask for clarification "Do you mean the blue box labeled 'Fragile'?" This prevents misdelivery.

In summary, while computer vision technology enhances interaction efficiency in some applications, multi-turn dialogue proves to be an effective means of resolving ambiguities in complex, redundant, or repetitive environments. Text and vision each have their own strengths in different aspects. Our aim is to study the representation power of natural language and the capability of modern vision-language models. So we assume that both R_i and D_i are purely texts. Nevertheless, the observation o_i may be obtained from all sensors available to the robot, so o_i may contain various types of data, such as images, depths, and thermals.

Existing approaches to disambiguating the position information within dialogues are mostly *attribute-based* or *spatial-based*. In CoWs [21], a user can issue commands "small, black, metallic alarm clock" and "house plant on a dresser near a spray bottle" via appearance and spatial descriptions. In FindThis [31], a user can issue a command "find my laptop on the desk, there is a sticker on it" for the robot to resolve. However, both CoWs and FindThis make a strong assumption that the target object has uniquely identifiable attributes that separate it from others. In this work, we assume that the target object does not necessarily have unique attributes and thus further contexts need to be specified (for example, if two laptops of the same type are on a table, we have to identify the one "nearby the corner"). We refer to this as *context-based* approach. As a pre-assumption, our following discussions will focus on the positional contexts within dialogues and ignore those out-of-scope contexts.

IV. SDA-LLM Framework

A. Task Definition

This work addresses the position disambiguation issue during robot navigation through a multi-turn dialogue interaction. We assume that an autonomous robot relies on the "navigate-and-find" paradigm to identify and reach its targets. Fig. 1 shows our navigation and dialogue framework. The robot is equipped with a LiDAR and a RGB-D camera. We assume that before the robot starts to navigate, it has already explored the space and obtained an offline 2D LiDAR map M_{lidar}. Recall Eq. 1 and the dialogue sequence $D_1 \rightarrow R_1 \rightarrow D_2 \rightarrow R_2 \rightarrow \dots D_k \rightarrow R_k$.

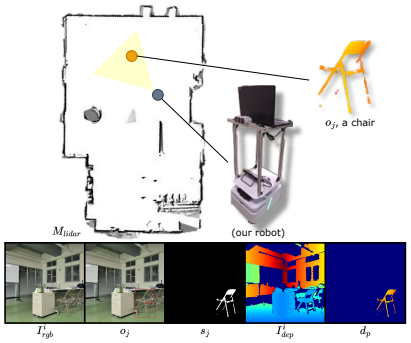


Fig. 2: An example of level-1 mapping.

For each dialogue $D_i, i \leq k$, the robot has to interpret it into a candidate position/object set T in the environment and respond in text. The role of SDA-LLM is to address the ambiguity issue, if any, and produce a response R_i by forming a mission if $|T| = 1$ or asking for further clarification if $|T| > 1$. After the disambiguation process, the mission should contain a precise navigation target. Below, we describe the processing of one mission, which involves two mapping stages. However, we ignore the navigation details as it is out of our scope.

B. Level-1 Mapping: Object-to-LiDAR Map

During navigation, the robot will continuously execute two tasks: (i) position itself in the LiDAR map and (ii) collect object information in its current environment. Task (i) can be done with SLAM by comparing to the offline map M_{lidar} (we omit the details here). Let the robot's current position be P_{cur} and its current angle be A_{cur}.

Task (ii) is done with the assistance of SAM (Segmentation Anything Model) [34]. The robot will rotate 360° at its current position P_{cur} and take ω snapshots by its RGB-D camera. (In our implementation, $\omega = 8$ with angle space of 45 degrees.) Let the RGB images be $I^1_{rgb}, I^2_{rgb}, \dots, I^\omega_{rgb}$ and depth images be $I^1_{dep}, I^2_{dep}, \dots, I^\omega_{dep}$. Each $I^i_{rgb}, i = 1..\omega$, is processed by an object detection model (such as YOLO [35]) to obtain a list of object bounding boxes. A unique ID is then assigned to each object. As the robot knows the direction, angle, and distance of each object, duplicate objects appearing in multiple images can be eliminated based on their extremely close distances. This potentially results in a list O_{list} of uniquely identifiable objects. Then, the image of each object $o_j \in O_{list}$ is cropped by its bounding box and sent to SAM to find its mask, named s_j.

With O_{list} and each o_j's mask s_j, we can map o_j to M_{lidar}. For each pixel (x_p, y_p) within mask s_j, we compute:

$$\begin{aligned} \Theta &= \frac{FoV_x}{w} \times (x_p - x_c) \\ \Phi &= \frac{FoV_y}{h} \times (y_p - y_c) \\ D_h &= d_p \times \cos\Phi \\ \Delta x &= D_h \times \cos(\Theta + A_{cur}) \\ \Delta y &= D_h \times \sin(\Theta + A_{cur}) \end{aligned} \tag{2}$$

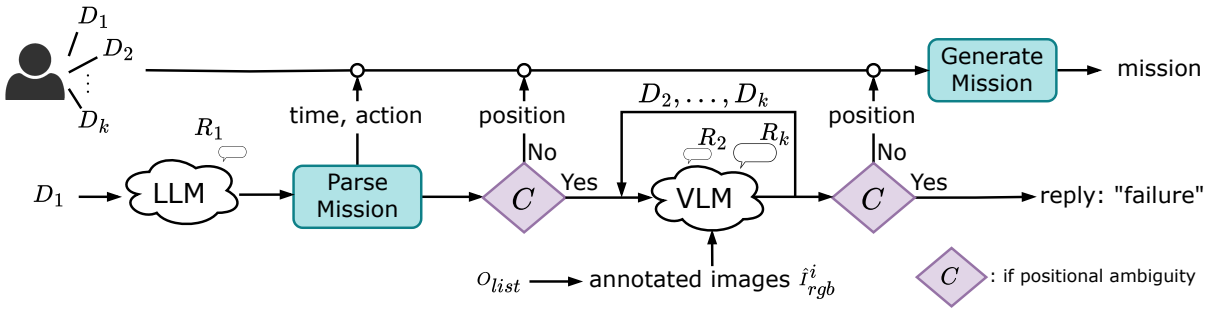


Fig. 3: The language-to-object (level-2) mapping module.

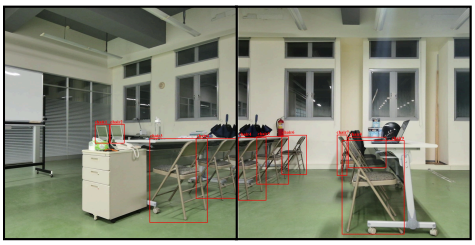


Fig. 4: The annotated images $\hat{I}^x_{rgb}, x = 1..\omega$, with object bounding boxes and object IDs shown in red.

Θ and Φ are respectively the azimuth angle and elevation angle with respect to the robot's camera. FoV_x represents the horizontal field of view, FoV_y represents the vertical field of view, w is the width of the sensor, h is the height of the sensor, (x_c, y_c) is the center of the image, and d_p is the depth of pixel (x_p, y_p) existing in a certain (I^i_{rgb}, I^i_{dep}) pair. Then we compute $P_{cur} + (\Delta x, \Delta y)$ as the position of (x_p, y_p) in M_{lidar}.

By repeating the above process for all $(x_p, y_p) \in s_j$, we can map object o_j onto M_{lidar}. By mapping each $o_j \in O_{list}$ onto M_{lidar}, we can obtain the online LiDAR map $\hat{M}_{lidar}$. An example is shown in Fig 2.

C. Level-2 Mapping: Language-to-Object

To initiate a mission, the user needs to exchange a sequence of dialogues, $D_1, D_2, \ldots, D_k$, $k \geq 1$, with SDA-LLM. The responses of SDA-LLM are denoted by $R_1, R_2, \ldots, R_k$, respectively. As our focus is positional ambiguity, we assume that D_1 contains time, position, and action information of the mission, while $D_i, i \geq 2$, are included, if necessary, for disambiguation.

Fig. 3 shows the architecture of the level-2 mapping. The first dialogue D_1 is sent to the LLM to interpret the user's intention. From the response R_1 of the LLM, we try to parse the details of the mission. If there is no ambiguity regarding the position, a mission can be created. Otherwise, the subsequent dialogues are processed by the VLM.

Suppose that the current dialogue is $D_i, i \geq 2$. We prompt the VLM with two inputs: (i) the text D_i and (ii) the annotated images $\hat{I}^1_{rgb}, \hat{I}^2_{rgb}, \ldots, \hat{I}^\omega_{rgb}$ obtained from the original RGB images $I^1_{rgb}, I^2_{rgb}, \ldots, I^\omega_{rgb}$, respectively, by drawing bounding boxes and object ID tags on the images.

System Prompt:
You are an assistant that helps the user **identify a specific target object** across 8 images. Each image may contain multiple objects, each marked with a **bounding box and an associated ID**. Your task is to conduct a **multi-turn conversation**, progressively narrowing down the selection based on the user's input until the exact target object is identified.

Response Guidelines:
1.First Response:
- •Based on the user's description, **identify all objects that match the initial criteria**.
- •For **each candidate object**, provide:
  - ○**ID**
  - ○**Detailed context** (e.g., position, color, shape, size, relation to nearby objects).
- •Ask a **specific follow-up question** to help refine the selection further.

2.Subsequent Responses:
- •**Filter the candidate list** based on new user-provided details.
- •In **each turn**, explicitly list **only the remaining candidate object IDs** and their updated **detailed context**.
- •Continue asking targeted follow-up questions to further eliminate incorrect objects.

3.Final Confirmation:
- •Once only one object remains, **clearly state the final identified object ID** with its full description.
- •Ask the user for **confirmation** before concluding the selection process.

Example Conversation Format:
User (Turn 1):
"Can you find the sofa near the window in the living room?"
GPT (Turn 1):
*"Based on your description, I have identified three possible sofas near the window:
•**Sofa 10** – A **dark-colored** sofa positioned **directly next to the window**.
•**Sofa 11** – A **light-colored** sofa, **slightly away** from the window, with its **side facing it**.
•**Sofa 12** – A **blue sofa**, **near the window**, but **angled differently**.
Could you provide more details, such as the sofa's **color, style, or size**?"*

User (Turn 2):
"It is next to the window and is a modern-style sofa."
GPT (Turn 2):
*"Thank you for the details! Based on this information, I have **narrowed it down** to the following candidates:
•**Sofa 10** – A **medium-sized modern dark-colored sofa** positioned **directly next to the window**.
•**Sofa 12** – A **large blue modern sofa**, positioned **near the window**, but slightly **angled differently**.
Could you confirm whether the sofa is **positioned straight in line** with the window or is **slightly angled**?"*

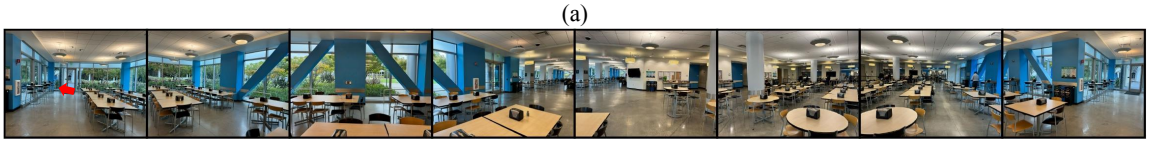
User (Turn 3):
"It's positioned directly in line with the window."
GPT (Turn 3):
*"Based on your full description, the target object is:
•**Sofa 10** – A **medium-sized modern dark-colored sofa**, positioned **directly in line with the window**.
Does this match the object you are looking for?"*

Fig. 5: Our system prompt to GPT-4o and an example.

Specifically, each $\hat{I}^x_{rgb}, x = 1..\omega$, is obtained as follows. Recall that we already obtained the object list O_{list}, in which each object is unique. For each object o_j appearing within I^x_{rgb}, we draw the bounding box of o_j according to its coordinates. We also add a tag of o_j's ID on top of the bounding box in the standard OpenCV style. This results in the annotated image $\hat{I}^x_{rgb}$. Several examples are shown in Fig. 4. We will verify modern VLMs' capability in Sec. V.

In addition to the above inputs, we also give a system prompt to the VLM in the beginning, as shown in Fig. 5. It consists of object identification instructions and contextual support for iterative dialogues. First, it instructs the VLM to identify a precise bounding box and a unique ID for each potential object that matches the user's description. Second, for every candidate object, it supplies additional contextual details, e.g., spatial relationships and attribute cues, to assist the user in disambiguating candidate objects. Last, when only one object remains, it confirms the unique ID with the user to finalize the selection. An example of responses is in the

(a)



(b)

Fig. 6: Examples of snapshots taken in (a) meeting room I and (b) cafeteria.

second section of Fig. 5.

In addition to text, the response R_i may include the snapshots taken by the robot. If images are provided to the user, a simple click on images can resolve the ambiguity. However, we are interested in investigating the spatial reasoning capabilities of the VLM. Therefore, we assume no images provided to prolong the dialogue with the user.

V. Dataset and Experiment Results

A. VisDial: Vision-Dialogue Dataset

The inputs to the VLM consist of $\omega = 8$ snapshots taken by the robot at a fixed position and a sequence of dialogues given by a user. To evaluate our SDA-LLM framework, we design a *Vision-Dialogue (VisDial)* dataset. Although the snapshots are easy to obtain, the dilemma is that the next dialogue D_{i+1} should depend on the current reply R_i, making D_{i+1} uncertain. To derive a fixed dataset, we designed a *gradual narrowing-down approach* when designing *VisDial*. There are two types of dialogues:

- Type A: The first dialogue D_1 refers to a unique specific object, but due to the robot's perspective or environmental complexity, the VLM may not be able to accurately identify the object. Therefore, subsequent dialogues $D_2, D_3, ..., D_k$ will provide additional information, progressively offering more attributes/contexts of the object until the robot can precisely determine the correct object.
- Type B: Suppose that there is a set of n_0 objects of the same type in the environment. The first dialogue D_1 itself is ambiguous and covers a subset of n_1 objects ($n_1 < n_0$). Each of the subsequent dialogues $D_2, D_3, ..., D_k$ will provide additional constraints or contextual details, progressively narrowing down the range of the candidate objects, until the final dialogue D_k, which can precisely identify a single target object ($n_k = 1$). Note that these numbers $n_i, i = 1..k$, are ground truth. A model's task is to try to identify as many of these n_i objects, from the given snapshots, as possible.

With the design, we are able to use fixed dialogue patterns to evaluate an implementation of SDA-LLM in resolving positional ambiguity. Note that the dialogue length, k, may vary in each data item.

The VisDial dataset was collected in 5 different spaces:

- Meeting room I: This space is about 50 m^2. The space is relatively simple and small, with 11 chairs, 1 table, 1 sofa set, 1 coffee table, 1 umbrella, 1 window, 1 cabinet, and 1 whiteboard. We designed 1 snapshot point and 10 Type-A dialogue sequences.
- Office: This space is about 150 m^2. The space is relatively large but simple, with 12 chairs, 4 tables, 5 windows, 1 cabinet, 4 whiteboards, and other office objects. We designed 1 snapshot point and 15 Type-A dialogue sequences.
- Meeting room II: This space is about 20 m^2. The space is relatively simple, with 10 chairs, 3 tables, and other everyday objects. We designed 3 snapshot points and 15 Type-B dialogue sequences.
- Classroom: This space is about 90 m^2. The space is somewhat complex, with approximately 40 chairs, 40 tables, classroom objects, and 2 sets of large whiteboards. We designed 1 snapshot point and 5 Type-B dialogue sequences.
- Cafeteria: This space is about 350 m^2. The space is highly complex, with approximately 150 chairs and 35 tables shown. We designed 4 snapshot points and 20 Type-B dialogue sequences.

For example, the following Type-A dialogue sequence is designed for the snapshots in Fig. 6a (the answer is the chair marked by a red arrow in the fourth snapshot).

```
D1: Help me find the chair closest to
the whiteboard.
D2: It also needs to be the closest to
the cabinet.
D3: That chair is also next to the
protruding wall.
D4: It is also at the edge of the
U-shaped table.
D5: Finally, the chair should be
directly in front of the white paper
with the QR code.
```

The following Type-B sequence is designed for Fig. 6b (the answer is the chair marked by a red arrow in the first snapshot).

```
D1: Please go to the chair.
```

TABLE I: Quantitative evaluation of the disambiguation capability of GPT-4o. ($\lambda_{sr} = 0.8$ and $\lambda_{as} = 0.2$ for Type-A dialogues; $\lambda_{ar} = 0.6$ and $\lambda_{ns} = 0.4$ for Type-B dialogues.)

		SR / AR	AS / NS	$T_{A/B}$
Type-A	Meeting room I	0.636	0.79	0.667
	Office	0.866	0.835	0.86
Type-B	Meeting room II	1	0.783	0.913
	Classroom-1	0.8	0.759	0.784
	Classroom-2	0.4	0.651	0.5
	Classroom-3	0.6	0.663	0.625
	Classroom-4	1	0.948	0.979
	Cafeteria-1	0.6	0.679	0.632
	Cafeteria-2	1	1	1.000
	Cafeteria-3	1	0.972	0.993

```
D2: Hmm, I mean a high chair.
D3: I think the high chair I need is
left of the door and closest to it.
```

The above dialogues in VisDial are manually generated. To augment our dataset, each dialogue D_i is further rewritten using LLM to expand diversity and increase generalizability. Specifically, the LLM is asked to generate alternatives that retain the same semantic meanings of D_i. (Alternatively, one may apply techniques such as syntactic transformations, synonym replacements, and style transfers.) For example, for a length-k dialogue, if each dialogue is augmented into 3 dialogues, 3^k sequences can be obtained. This significantly improves the diversity of VisDial.

B. Disambiguation Metrics and Evaluations

To evaluate a VLM's disambiguation capability, we define two metrics for each dialogue type. Let $\mathcal{S}$ be the dataset and $s \in \mathcal{S}$ be a data item of type A with a dialogue length of k. We define *success rate* (SR) and *accuracy score* (AS) with respect to s as follows:

$$SR(s) = (k - (\alpha - 1))/k$$
$$AS(s) = \begin{cases} \frac{1}{\alpha}\Sigma_{i=1}^{\alpha} \frac{\mathbf{1}(\text{found})}{\beta_i}, & \text{if } \alpha \leq k \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Here, α is the number of dialogues consumed during the test. If $\alpha \leq k$, the unique object is identified; otherwise, we assume that $\alpha = k + 1$. So a smaller α results in a higher $SR(s)$. β_i means the number of objects that the robot infers in response R_i. If the unique object is within the predicted set, $\mathbf{1}$(found) is 1; otherwise, it is 0. So a smaller β_i gives a higher $AS(s)$. We aggregate them into a total Type-A score:

$$T_A = \sum_{s \in \mathcal{S}} (\lambda_{sr} \times SR(s) + \lambda_{as} \times AS(s))/|\mathcal{S}| \quad (4)$$

where λ_{sr} and λ_{as} are weighting factors.

If $s \in \mathcal{S}$ is a Type-B data item, we define the *accuracy rate* (AR) and *narrowing score* (NS) as follows:

$$AR(s) = \begin{cases} 1, & \text{if } found \\ 0, & \text{otherwise} \end{cases}$$
$$NS(s) = \frac{1}{\alpha} \sum_{i=1}^{\alpha} \frac{x_i \cap x_i'}{x_i \cup x_i'} \quad (5)$$

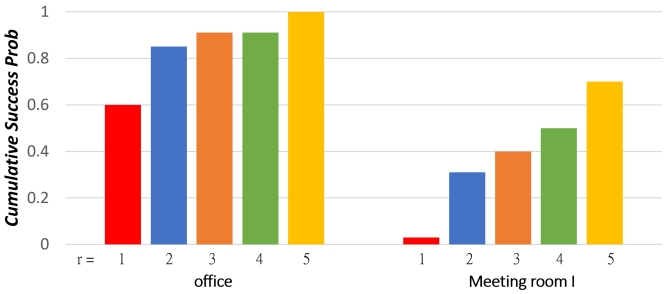


Fig. 7: The cumulative success probability when at most γ turns are allowed for Type-A data in Office and Meeting Room I.

where x_i is the set of actual objects in dialogue D_i and x_i' is the set of predicted objects by the model in R_i. We aggregate them into a total Type-B score:

$$T_B = \sum_{s \in \mathcal{S}} (\lambda_{ar} \times AR(s) + \lambda_{ns} \times NS(s))/|\mathcal{S}| \quad (6)$$

where λ_{ar} and λ_{ns} are weighting factors.

Based on the above metrics, we show our quantitative evaluation results in Table I. These experiments were conducted by using GPT-4o as the LLM and VLM. For Type-A dialogues, we observe that in the office setup, the SR, NS, and T_A scores are all slightly higher than those of the meeting room I setup. We suspect the reason to be that the types and numbers of objects in these two spaces are similar. However, the office space is relatively larger, making this task relatively easier for the VLM to solve. For Type-B dialogues, the meeting room II has consistently high scores since its setup is relatively simple. In contrast, we observe inconsistent scores under different cases for the classroom and cafeteria setups, which are more complex. For example, the scores of classroom-4, cafeteria-2, and cafeteria-3 are pretty high, while the scores for the other cases are much lower. This is due to the complexities of these spaces, including numbers, types, sizes, and diversities of the objects in these spaces. It is to be noted that due to our formulations, T_A scores will generally be lower than T_B scores.

To demonstrate why multi-turn dialogue is crucial, we show the cumulative success probability when at most γ turns are allowed to resolve positional ambiguity for Type-A data in Fig. 7. For Office, the success probability can reach 1.0 by five turns. For Meeting Room I, as the environment is more ambiguous, the single-turn success probability is very low, but it increases gradually as more turns are allowed, peaking at around 0.8.

C. Object Mapping Accuracy

Here, we test level-1 mapping accuracy for Type-A data in Office and Meeting Room I. We omit Type-B data because its first dialogue is already ambiguous. Using Eq. 2, we use the average depth of an object to estimate its position in the LiDAR map M_{lidar}. We then calculate the error between the predicted position to the ground truth center

TABLE II: Error analyses of level-1 mapping.

	Meeting room I	Office
Mean Error (m)	0.657	0.983
Standard Deviation (m)	0.566	1.159
Min Error (m)	0.058	0.121
Max Error (m)	1.848	4.923

of the object. The results reflect the accuracy of SAM and hardware limitations. As Table II shows, the errors are within an acceptable range. Since the space of Office is much larger than that of Meeting Room I, it has a larger error.

D. Implementation Details and Demos

Our robot testbed was developed based on ROS Melodic. The main control board was NVIDIA Jetson Nano, and the operating system used was Ubuntu 18.04 LTS. The robot was equipped with a LiDAR and two RGB-D cameras. The LiDAR was for constructing M_{lidar} and positioning the robot. The first RGB-D camera, ORBBEC Astra S, was used for obstacle avoidance and was installed at the bottom of the robot. The second RGB-D camera, ORBBEC Femto Bolt, was installed at the top of the robot for sensing the environment. We used YOLOv8 [36] for object detection and SAM [34] for object segmentation. We adopted GPT-4o [12] as both the LLM and the VLM in our method. The robot interacted with GPT-4o through its APIs. We have also tried other models; refer to Sec. VI for more discussions.

In Fig. 8, we demonstrate a successful prompting example, where the dialogue is of Type-A. A failure dialogue example of Type-B is demonstrated in Fig. 9.

Fig. 10a shows an example that an attribute-based approach (e.g., CoWs and FindThis) may successfully resolve. The towel has its unique attributes, while the two beds are the same. By referring to the towel, the bed are distinguishable. Fig. 10b shows an example where there is no attribute to distinguish those chairs, unless context information is referred to. In our test, only our SDA-LLM can interpret the context and successfully resolve the ambiguity.

VI. Conclusions and Limitations

This work has addressed a new spatial ambiguity challenge in natural language processing that is rooted naturally at the imprecision of human dialogues. Computer vision and natural language processing each has its own representation strengths. SDA-LLM addresses the gap between them through in the human-robot interaction scenario. Disambiguation has been addressed by previous work for attribute separation. SDA-LLM is the first to handle spatial ambiguity. Table III provides a comparative analysis of various methods, highlighting our contributions. The work contributes in developing a 2-level mapping framework, leveraging modern LLM and VLM for text and vision mapping, defining Type-A and Type-B dialogues, and validating on a ROS-based robot. In addition, we also contributed a new VisDial dataset for spatial ambiguity study.

As to limitations, our model requires to submit multiple snapshots. At the time of this study, GPT-4o was the only

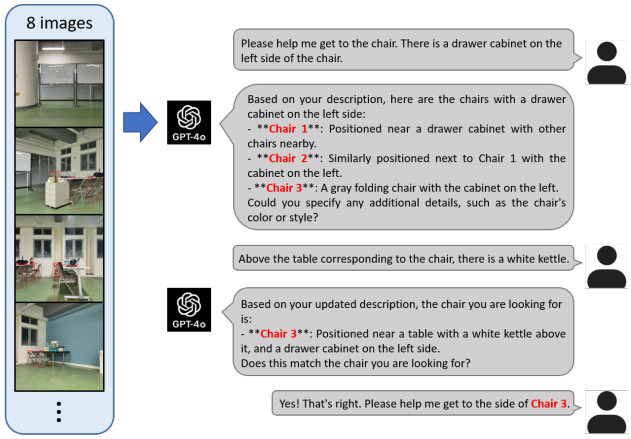


Fig. 8: A successful prompting example (Type-A). The target object, chair3, with a drawer cabinet on its left side and a white kettle on the corresponding table. The example is to test how much information is needed for the model to correctly identify the object.

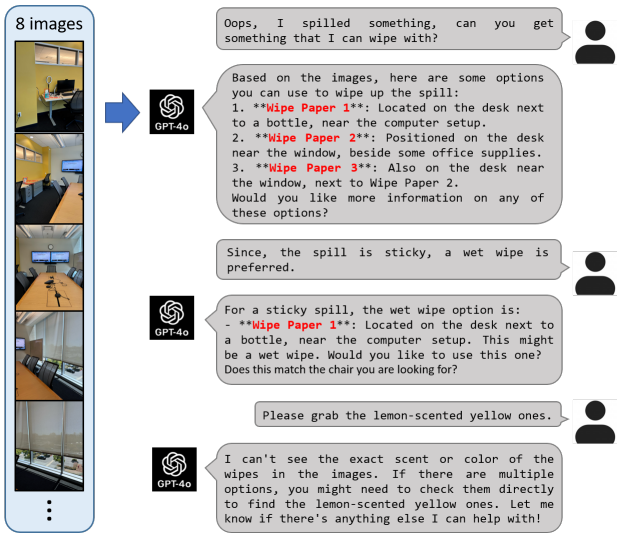


Fig. 9: A failed prompting example (Type-B). The target object is 'wipe paper 3'. This example is designed to test whether the model can analyze ambiguous human commands and correctly identify how many objects are referenced by the ambiguous commands.

Fig. 10: (a) An example resolvable by attributes. (b) An example resolvable by contexts. A dialogue "Go to the chair next to the fire extinguisher on the table" can identify the right chair.

TABLE III: Comparison of disambiguation methods. Using post-processing and allowing 2 turns, FindThis can handle simple spatial ambiguity. (DA=disambiguation, LDP=LLM-driven planning)

Method	no of turns	Spat. DA	Att. DA	LDP
CoW [21]	single	No	Yes	No
VLMaps [16]	single	No	Yes	No
LM-Nav [29]	single	No	Yes	No
NavGPT [30]	single	No	Yes	Yes
FindThis [31]	two	partial	Yes	Yes
ThinkActAsk [22]	many	No	Yes	Yes
SDA-LLM	many	Yes	Yes	Yes

VLM capable of handling multiple images at a time. Most VLMs accept only one image per conversation, which did not meet our needs. An alternative is to concatenate multiple images into a single large image. However, most models have restriction on image size, resulting in image resizing and blurry issues. Another way is to supply a sequence of images in multiple rounds of conversations. This leads to a longer dialogue, causing catastrophic forgetting and significantly impacting performance.

References

[1] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot." in *IROS*, 1991.

[2] C. Stachniss, *Robotic mapping and exploration*, 2009, vol. 55.

[3] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.

[4] J. Borenstein, Y. Koren, *et al.*, "Histogramic in-motion mapping for mobile robot obstacle avoidance," *IEEE Transactions on robotics and automation*, vol. 7, no. 4, pp. 535–539, 1991.

[5] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00022, 2017.

[6] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, pp. 54–70, 2023.

[7] B. Alsadik and S. Karam, "The simultaneous localization and mapping (slam)-an overview," *Journal of Applied Science and Technology Trends*, vol. 2, no. 02, pp. 147–158, 2021.

[8] S. Gobhinath, K. Anandapoorani, K. Anitha, D. D. Sri, and R. DivyaDharshini, "Simultaneous localization and mapping [slam] of robotic operating system for mobile robots," in *International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021.

[9] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39 830–39 846, 2020.

[10] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[11] J. Guo, P. Chen, Y. Jiang, H. Yokoi, and S. Togo, "Real-time object detection with deep learning for robot vision on mixed reality device," in *IEEE Global Conference on Life Sciences and Technologies (LifeTech)*, 2021.

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

[13] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G. A. Fink, and G. Sagerer, "Multi-modal anchoring for human–robot interaction," *Robotics and Autonomous Systems*, vol. 43, no. 2-3, pp. 133–147, 2003.

[15] Q. Tang, J. Liang, and F. Zhu, "A comparative review on multi-modal sensors fusion based on deep learning," *Signal Processing*, p. 109165, 2023.

[16] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[17] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *Conference on Robot Learning*, 2020.

[18] K. X. Nguyen, Y. Bisk, and H. D. Iii, "A framework for learning to request rich and contextually useful information from humans," in *International Conference on Machine Learning*, 2022.

[19] T.-C. Chi, M. Shen, M. Eric, S. Kim, and D. Hakkani-Tur, "Just ask: An interactive learning framework for vision and language navigation," in *Proceedings of the AAAI conference on artificial intelligence*, 2020.

[20] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur, "Teach: Task-driven embodied agents that chat," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[21] 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.

[22] Y. Dai, R. Peng, S. Li, and J. Chai, "Think, act, and ask: Open-world interactive personalized robot navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[23] C. Liu, F. Zhu, X. Chang, X. Liang, Z. Ge, and Y.-D. Shen, "Vision-language navigation with random environmental mixup," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[24] J. Li, H. Tan, and M. Bansal, "Envedit: Environment editing for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[25] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, "History aware multimodal transformer for vision-and-language navigation," *Advances in neural information processing systems*, vol. 34, pp. 5834–5847, 2021.

[26] H. Wang, W. Wang, W. Liang, C. Xiong, and J. Shen, "Structured scene memory for vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[27] W. Hao, C. Li, X. Li, L. Carin, and J. Gao, "Towards learning a generic agent for vision-and-language navigation via pre-training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[28] P.-L. Guhur, M. Tapaswi, S. Chen, I. Laptev, and C. Schmid, "Airbert: In-domain pretraining for vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[29] 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*, 2023.

[30] G. Zhou, Y. Hong, and Q. Wu, "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[31] A. Majumdar, F. Xia, D. Batra, L. Guibas, *et al.*, "Findthis: Language-driven object disambiguation in indoor environments," in *Conference on Robot Learning (CoRL)*, 2023.

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

[33] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," 2023. [Online]. Available: https://arxiv.org/abs/2305.06500

[34] 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," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[35] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[36] R. Varghese and M. Sambath, "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 2024.