Memory-Maze: Scenario Driven Benchmark and Visual Language Navigation Model for Guiding Blind People

Masaki Kuribayashi*[1], Kohei Uehara*[2], Allan Wang[2,3], Daisuke Sato[3],
Renato Ribeiro*[2], Simon Chu[3], Shigeo Morishima[4]

***Abstract*— Visual Language Navigation (VLN) powered robots have the potential to guide blind people by understanding route instructions provided by sighted passersby. This capability allows robots to operate in environments often unknown a prior. Existing VLN models are insufficient for the scenario of navigation guidance for blind people, as they need to understand routes described from human memory, which frequently contains stutters, errors, and omissions of details as opposed to those obtained by thinking out loud, such as in the R2R dataset. However, existing benchmarks do not contain instructions obtained from human memory in natural environments. To this end, we present our benchmark, Memory-Maze, which simulates the scenario of seeking route instructions for guiding blind people. Our benchmark contains a maze-like structured virtual environment and novel route instruction data from human memory. Our analysis demonstrates that instructions data collected from memory were longer and contained more varied wording. We further propose a VLN model better equipped to handle the scenario by leveraging Large Language Models (LLM) and show that existing state-of-the-art models perform suboptimally on our benchmark.**

I. Introduction

Visual language navigation (VLN) is a task where an agent with visual access to the surroundings navigates under a human's instructions [1]. Recently, navigation robots for blind people have been developed to help them gain independence [2], [3], [4], [5], such as robots that allow users to choose destinations within prebuilt maps [2], [4], [3]. One scenario in which such robots would benefit from the VLN technology is where blind people request instructions to their destinations from sighted passersby in unfamiliar buildings [6]. In this scenario, the VLN technology deployed on navigation robots may assist their blind users by understanding verbal instructions from the passersby and then autonomously guiding them to their destinations. VLN technology could also allow robots to operate autonomously without relying on building infrastructure or prebuilt maps, which is crucial for allowing robots to assist blind people in navigating various new environments [5], [7].

However, direct application of existing VLN models to the blind people navigation scenario is currently limited, as there is a need for a benchmark that reflects the blind users' demands realistically. Many VLN tasks have been addressed in environments such as static houses [1] or roadways [9]. Nonetheless, it is also most important for blind individuals

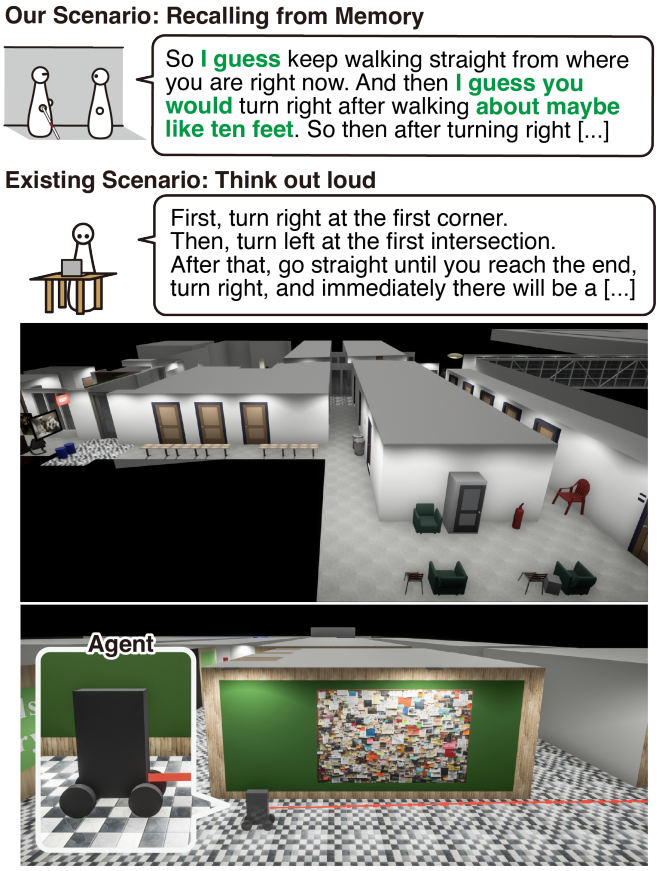


Fig. 1: **Memory-Maze Benchmark.** Top: the instructions obtained in the memory-based scenario contain unique phrases, highlighted in green, in contrast to those collected in traditional think-out-loud settings. Middle: Our benchmark environment based on the CARLA simulator [8]. Bottom: the VLN agent that navigates within the environment.

to navigate large public spaces such as shopping malls or university hallways. Compared to existing environments, these environments are characterized by physical turning points and intersections, resembling a maze. Besides the environmental difference, in existing VLN literature, natural language instructions are provided by thinking out loud. In other words, annotators visually navigate a virtual environment and type out instructions for constructing routes concurrently. In our scenario, sighted passersby must describe the route from their memory, which often contains errors such as inaccurate estimates of distances, hallucinations of landmark objects, and omissions of key turning points. To the best of our knowledge, our benchmark is the first to address the

* Equal Contribution [1]Waseda University [2]Miraikan - The National Museum of Emerging Science and Innovation [3]Carnegie Mellon University [4]Waseda Research Institute for Science and Engineering, Corresponding Author Email: rugbykuribayashi@toki.waseda.jp

scenario of a blind user seeking memory-based instructions from sighted passersby in maze-like public spaces.

We present *Memory-Maze* (Fig. 1), a benchmark that reflects the blind user navigation scenario. Memory-Maze contains virtual environments of real-world public spaces. It is based on CARLA [8], which enables us to simulate various sensor data (*e.g.*, LiDAR) from robots. It also contains instructions data gathered from two studies from sighted individuals. In the first study, instructions were gathered through online questionnaires by observing walk-through videos from a first-person perspective. This is similar to the annotation method used in existing research. In the second study, instructions were collected in-person by asking sighted passersby to describe the same routes from their memories. This reflects the novel scenarios envisioned in our benchmark. We observed different characteristics among the two studies in terms of length, number of errors, variety, etc.

Alongside our benchmark, we developed a novel VLN model that performs competitively in the Memory-Maze benchmark. Given the differences in environments and instruction annotation, directly applying existing supervised models poses a challenge due to their limited performance in unseen settings [10]. In order to realize a scalable navigation robot that could be used in various unseen environments for blind people, our benchmark scenario needs to be handled without retraining or fine-tuning. Hence, we leverage a Large Language Model (LLM) due to its high capability to understand complex natural language and its potential for generalization in the new task without training. Upon receiving an instruction, our VLN model utilizes the LLM to convert the instruction into Python code based on the defined robot control API (Sec. III-B) for route navigation. This code generation approach modularizes low-level commands such as path-planning for collision avoidance, eliminates the need for prebuilt maps (*i.e.*, navigation graphs), and enables the model to be applied across unseen environments. Our experiment showed that our model performed satisfactorily on our benchmark while outperforming the state-of-the-art methods [11], [12]. Through the study, we demonstrated the difficulty of our benchmark and a tendency that onsite instructions are more difficult for VLN models to handle.

We summarize our contributions below.

1) We constructed Memory-Maze, a benchmark containing virtual environments of a university building and a museum, and gathered two sets of instructions, one collected by thinking out loud and one obtained from human memory.
2) We developed an LLM-based VLN model and our model outperforms the state-of-the-art models on our benchmark. We revealed the gap between the instructions collected based on memory and those collected by thinking out loud.

The benchmark and the VLN model will be made open-source upon acceptance.

II. Related Work

A. Assistive Navigation Systems for Blind People

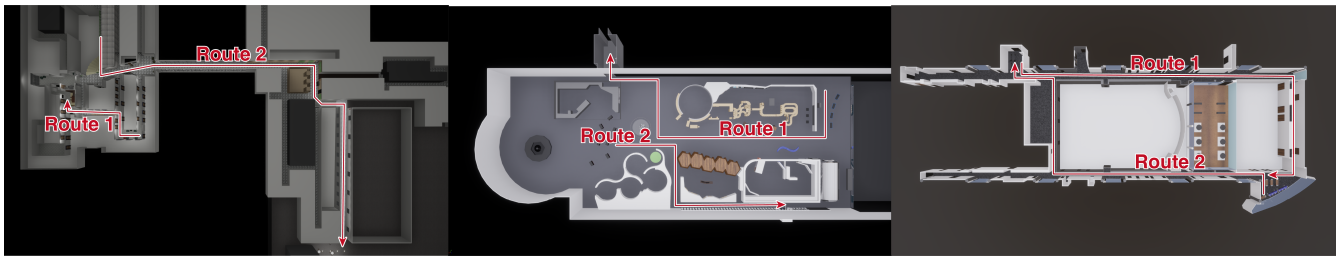
Recently, navigation robots have been explored to aid blind people in avoiding obstacles while navigating [2], [3], [4], [13]. A common practice is to prepare prebuilt maps and infrastructure for localization and manual destination selection [2], [3], [4]. This practice poses a limitation for these systems, as prebuilt maps and infrastructure are costly to obtain and maintain. Consequently, a map-less approach was also proposed [5]. Based on the instruction from sighted passersby [6], users input navigation directions through the buttons on the robot's handle. However, because the system needs users to understand and memorize the instructions, high cognitive loads are placed on the users. To address this, our work aims to present a system that directly interprets instructions from sighted passersby and navigates users autonomously to their destinations.

B. Benchmarks in VLN tasks

The VLN task has been conducted in various benchmarks, ranging from indoor [1], [14], [15] to outdoor [9], [16], [17] settings. Most of the instruction annotations of these benchmarks were created by annotators who typed while concurrently observing a virtual environment or by researchers who constructed them manually. This way of obtaining instructions is not suitable for our purpose, as it does not reflect the scenario of people describing routes from their memories. In our case, we gathered natural language instructions through both an online and an onsite study, where the onsite study contains instructions provided from human memories. In addition, most indoor virtual environments do not feature large public areas where blind people navigate, such as shopping malls or university hallways. These areas contain both static and dynamic obstacles and are characterized by the existence of turning points and intersections (Fig. 2). A similar environment is Touchdown [9]. However, its map structure is represented by a navigation graph (*i.e.*, an undirected graph that represents navigable points with nodes), whereas we assume no access to navigation graphs. Furthermore, our environment is fully configurable with various static and dynamic obstacles.

C. VLN Models

Researchers have explored solutions for VLN tasks using supervised models [1], [18], [19], which learn from a sequence of observations and actions to take. These supervised models often do not transfer well in unseen environments [10]. With the recent advancements in LLM, researchers have also explored methods that do not require retraining [11], [20], [21], [22], [23]. One such approach was to use LLMs to extract landmarks from instructions and follow chronologically [20], [21], [23]. Another approach was to utilize LLM to flexibly determine actions at each step. NavGPT [11] is a model that uses LLM iteratively to select the node to navigate to within a navigation graph. Additionally, researchers have explored approaches that utilize the code generation capability of LLM [24], [25]. In the



(a) University (b) Museum 5F (c) Museum 7F

Fig. 2: **Bird's-Eye Views of Memory-Maze Environment.** The benchmark contains three environments. The university includes features such as classrooms, offices, hallways, a kitchen, and a library. The 5th floor of the museum mainly contains exhibits. The 7th floor contains conference rooms, hallways, and a terrace area. Each environment includes two routes, totaling six routes.

method proposed by Biggie *et al.* [25], given a prebuilt 3D map, images from their robot, and a Python API, the model generates codes that locate a target object [26], maps the object's location on the 3D map, and navigates to the mapped location [25]. While these methods are effective when the given instructions include sufficient landmarks, instructions recalled from memory often contain insufficient landmarks, potentially leading to failure. Furthermore, these methods are limited by the need for a navigation graph or 3D map, which is difficult to construct for every unseen environment. To eliminate this requirement, models have been proposed to predict navigation graphs [27] or low-level actions [18], [12] iteratively. However, the need for iterative inference prolongs inference time, which may affect navigation by not reacting to dynamic obstacles responsively. Our model utilizes LLM to produce navigation codes that follow a specified path in a single iteration, and allows flexible integration of low-level planning algorithms for obstacle avoidance. This direct generation of navigation codes, coupled with existing low-level planning algorithms, allows operation without the need for navigation graphs.

III. Memory-Maze

Here, we describe our benchmark's virtual environment and the robot simulation program. To simulate our scenario, we selected a floor of a university building and two floors in a museum building (Fig. 2), which is characterized by the existence of multiple turning points.

A. Selecting and Building the Simulator

To simulate a scenario where a robot guides a blind person, it is necessary to simulate high-fidelity egocentric visuals that are realistic enough to run an image recognition algorithm. In robot simulation software such as Gazebo [28], it is difficult to create visually detailed environments, which limits the functionality of the image recognition algorithm. Furthermore, simulators used in existing VLN research, such as AI2-THOR [29] and Matterport3D [30], are designed for indoor simulation and are limited in their ability to customize environments and the types of sensors that can be simulated. Thus, we built a novel virtual environment from scratch on top of the CARLA [8] simulator. While primarily developed for autonomous driving simulations, CARLA's flexibility and compatibility with the Unreal Engine allowed us to create a detailed 3D model of the experimental site. CARLA also offers the ability to configure the existence of static and dynamic obstacles and to simulate various sensors like RGB cameras, depth sensors, and LiDAR sensors.

We created a 3D model of the experimental site using Fusion 360 [31] and imported it into CARLA. This 3D model accurately reproduces the experimental site, both visually and in terms of floor layout. It also includes major objects along the route (doors, chairs, a statue, etc.).

B. Implementation of the Control Program

Our next step was to develop a control program for the robot in the simulator. Utilizing CARLA's Python API to control the navigation robot, we implemented various control functions. We describe four major functions implemented.

We implemented functions for the agent to move forward (`move_forward(distance)`), find a turning point (`detect_turning_point()`), and turn (`turn(direction)`) using CARLA's `vehicle.apply_control` API. When using the `move_forward(distance)` function, to ensure the robot moves along the path without colliding with walls, we implemented a feature that makes the robot navigate as closely to the center of the corridor as possible. We calculate the central path based on the coordinates of the four corners of the corridor in the 3D model. The central path tracking is realized through PID control, which adjusts the robot's steering angles. When the `detect_turning_point()` function is used, it determines if the robot is in the pre-annotated areas of turning points and returns navigable directions if the robot is in one of them. Once the robot is at the turning point, it could change its direction using the `turn(direction)` function. In our experiment, coordinates of the corridor's corners and the turning point areas are acquired from the virtual environment to reduce errors resulting from noise in perception or control, and focus on executing instructions. However, these can be obtained using prior well-established methods [5].

Additionally, we implemented an image recognition module `detect_from_RGB_image(object)`, to manage landmark-related instructions such as *"turn after finding*

TABLE I: **Data Analysis.** The table presents the route length (RL), mean, median, and standard deviation (SD) of word counts in the collected instructions, and their failure rates (FR). For the onsite instructions, we also report the alternative rate (AR), the rate of describing alternative routes.

	Route	RL	Iteration	Mean	Median	SD	FR	AR
Online Study	University R1	40.27m	1	51.8	47.0	17.8	0.0%	-
			2	69.8	64.0	19.9	0.0%	-
	University R2	156.68m	1	81.3	81.0	24.9	9.09%	-
			2	98.3	99.0	31.2	3.03%	-
	Museum 5F R1	71.18m	1	81.4	78.0	36.3	17.39%	-
			2	88.9	90.0	32.3	17.39%	-
	Museum 5F R2	44.05m	1	60.1	53.5	21.5	9.09%	-
			2	71.0	61.0	33.9	4.55%	-
	Museum 7F R1	86.10m	1	98.2	91.5	42.5	13.64%	-
			2	96.7	90.0	42.2	18.18%	-
	Museum 7F R2	79.40m	1	71.3	68.0	25.8	4.35%	-
			2	95.0	85.0	47.4	0.00%	-
Onsite Study	University R1	40.27m	1	73.9	74.5	36.6	25.0%	10.0%
			2	102.9	94.5	51.1	25.0%	10.0%
	University R2	156.68m	1	131.0	115.5	73.2	40.0%	15.0%
			2	147.3	143.0	65.0	35.0%	15.0%
	Museum 5F R1	71.18m	1	68.2	64.0	27.4	76.19%	0.0%
			2	97.1	92.0	27.0	9.52%	0.0%
	Museum 5F R2	44.05m	1	65.5	51.0	42.7	45.45%	59.1%
			2	83.4	68.5	39.7	4.55%	63.6%
	Museum 7F R1	86.10m	1	68.7	69.5	27.5	54.54%	4.5%
			2	89.0	84.0	24.0	13.64%	9.0%
	Museum 7F R2	79.40m	1	79.5	69.0	40.0	52.38%	85.7%
			2	99.0	96.0	37.5	23.81%	90.1%

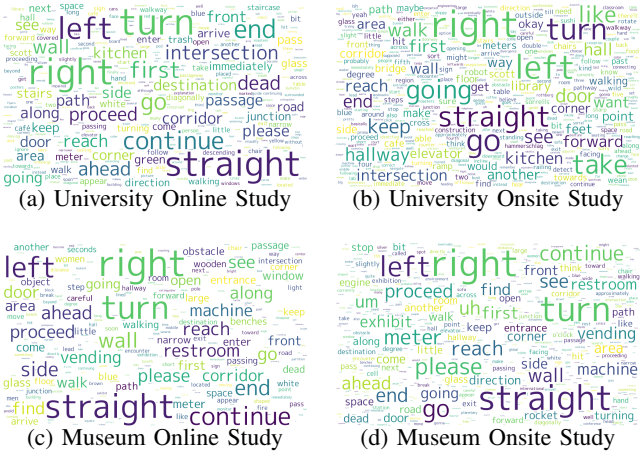
a chair." While most existing object detection models are designed to identify objects from predefined classes, they are not capable of detecting arbitrary objects. Therefore, we used Grounding DINO [32], an open-vocabulary object detection model. Open-vocabulary object detection models output bounding boxes for any object by using the object's name as a query. With the object detection model selected, we then used CARLA's robot ego-centric RGB sensors to capture images. To address tasks requiring the robot to identify an object multiple times (*e.g.*, *"turn after passing four doors"*), we added tracking algorithms to avoid counting the same object in different frames as distinct entities.

We further assume that in instructions that require finding landmark objects, the objects are located in close vicinity. For example, in the instruction *"turn after finding [object],"* although the camera could capture the object at a considerable distance, such instructions typically imply that *"[object]"* is near the robot. Therefore, we utilized the depth sensors available in CARLA to measure the distance to each object in the image, filtering out objects that are far away to ensure only those at close range are detected. We set the distance threshold to be four meters.

IV. Instruction Data Collection

A. Procedure

We conducted two studies, one online and one onsite, to collect natural language instruction data for routes at three locations: a floor across three buildings in a university and two floors in a museum. We designed the route as shown in Fig. 2. The studies were approved by our institutional review board (IRB), and informed consent was obtained from all participants. For each route, we obtained two rounds of instructions: one asking participants to describe the route to a blind person with a navigation robot naturally (first



(a) University Online Study (b) University Onsite Study

(c) Museum Online Study (d) Museum Onsite Study

Fig. 3: **Word Clouds.** The onsite instruction data contains unique phrases that come from talking while recalling from the memory, such as *"uh," "maybe,"* and *"okay."*

iteration) and another asking participants to describe the route after providing them with a brief description of the capability of the navigation robot (second iteration). The second instruction was collected to obtain more accurate memory-based instructions given by passersby. This was achieved by explaining the robot's capability (*e.g.*, being able to detect objects) to the participants. It simulates a scenario where a blind user may provide sighted passersby with robot information to obtain refined instructions. We expect that telling them about robots' capabilities would enable VLN models to achieve better performance.

In the first study, participants completed an online questionnaire designed to gather instructions that were similar to those in prior works. They were first presented with a scenario in which they communicated with a blind person accompanied by a navigation robot capable of following natural language instructions and 360° video walkthroughs of two routes. They were then asked to type instructions to the destination. They were allowed to re-watch the walkthrough videos at any time. We collected four instructions per participant. In total, 78 participants participated in the study, resulting in 312 instructions. The participants were gathered through university recruitment or through an online survey platform, and all were unfamiliar with the shown routes. The study was conducted in Japan, and the instructions were translated into English using GPT-4.

In the second study, we conducted an onsite in-person study. The aim of this study was to gather data that reflects the realistic scenario of sighted people describing the route from their memory. Thus, they did not watch the walkthrough video or experience the route during the study. The experimenter roleplayed as blind individuals, asked them for directions to the route destinations, and instructed them to describe the route verbally in two rounds. For the first iteration, we asked participants to describe the routes as naturally as possible. For the second iteration, to obtain more accurate instructions for the benchmark, in addition to explaining the robot's capabilities, the experimenter pointed

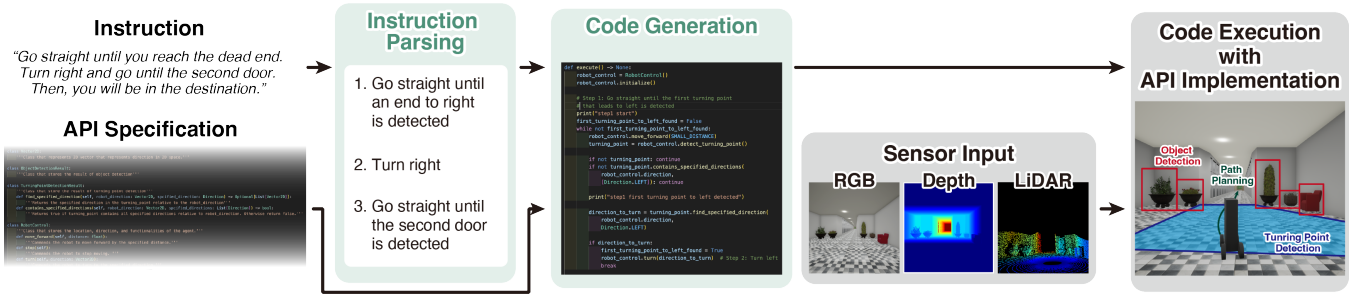


Fig. 4: **Method Overview.** Given a set of instructions from a sighted passerby, the LLM first parses it into an itemized format. Then, combined with the API specification, the LLM generates Python code directly to control the robot, which runs in the virtual environment using the simulated sensor inputs.

out errors in the participants' given instructions, such as a missing turn, and asked them to explain the route again. For the university routes, we recruited sighted passersby and ensured that all participants were familiar with the route by using a pre-study check survey. In this study, each participant described a single route, resulting in two instructions per participant. In total, 40 participants participated in the study at the university, contributing 80 instructions. For the museum routes, we recruited staff or recent visitors who were familiar with the museum layouts. In this study, each participant described two routes, resulting in four instructions per participant. In total, 43 participants participated in the study at the museum, contributing 172 instructions.

B. Benchmark Statistics

The mean, median, and standard deviation (SD) for the length of collected instructions are reported in Table I. First, we observed that the mean length and SD are longer for the second iteration in most cases, as participants tended to add more information on the second iteration. Also, we observed a tendency for instructions collected onsite to have higher lengths and more length variation. This is because, in the online study, participants described relevant and mostly accurate information about landmarks and turning points, while in the onsite study, many participants tried to be descriptive, relying on their memory, such as adding audio, olfactory cues, and conversational phrases such as *"I'm not 100% sure about this, but I think..."*.

The average instruction length and route distance in our benchmark are greater than those in previous datasets. For example, the R2R dataset includes instructions averaging approximately 30 words and route distances of about 10 meters [1], and the RxR dataset features instructions averaging 78 words and route distances of 14.9 meters [33].

The word clouds of the collected instructions are shown in Fig. 3. For the university environment, although the samples collected in the onsite study are fewer, they include 521 different words compared to the 381 words found in the samples from the online study. The same trend was noted in the museum environment, with 611 different words found in the onsite study and 586 words in the online study. This shows the greater diversity in the instructions' wording when described from memory. Although the instructions from the online study were translated using LLM, we believe that these results hold in the instructions' original language.

In Table I, we also manually analyzed each instruction to determine if it contained significant errors, *i.e.*, the number of failures in describing the route correctly. One author first conducted an initial failure review, after which multiple authors engaged in a discussion to reach a consensus on all samples. Reasons for the instructions in the online study that were classified as failures are turning to the wrong direction at a turning point, instructing a turn at the incorrect turning point, and suggesting unnecessary extra turns. For instructions collected in the onsite study, the reasons for the failures were containing extra turns, directing to an incorrect direction, leading to a wrong destination, lacking essential turn information, turning to incorrect directions at a turn, and containing inaccurate environmental details.

Interestingly, while examining the instructions, we realized that in the real world, humans may be able to correct errors in the instructions. For example, according to some passersby, the robot should go through a corridor between the hexagon exhibitions and the rectangular exhibition immediately to the right for museum 5F R2. However, there is actually no corridor between these two exhibitions. But it is possible to imagine where the nonexistent corridor might lead and try to find a detour. Being able to recognize errors in memory-based instructions is vital for aiding blind people to follow instructions provided by passersby. This is a unique aspect of our benchmark.

Finally, as shown in Table I, during the onsite study, we observed that participants sometimes described alternative routes compared to those we had initially anticipated and illustrated in Fig. 2, when they described the routes from their memory. Surprisingly, some participants described a different route in the second iteration compared to their initial description. This highlights the potential of humans to describe alternative routes in real-world scenarios and the need for VLN models to perform equally well in these alternative routes, thereby underscoring the naturalness of our benchmark.

V. VLN Model Implementation

In this section, we describe our proposed VLN model used to test our benchmark. To satisfy the requirements of our scenario, it contains two characteristics. First, we

utilized LLM's capability to generalize to various tasks and comprehend complex natural language instructions, so that no additional training is required when deployed in a new environment. Second, our method requires only a single inference iteration to generate low-level navigation code for robot control, in contrast to existing models that perform repeated inferences during navigation, which may prolong navigation time. It also eliminates the need for a navigation graph by generating codes that directly interface with low-level navigation modules. The generation of navigation code potentially leads to the flexibility of integrating existing, well-established methods into various modules, such as for obstacle avoidance or turning point detection.

We define the following as inputs to the agent: *natural language instruction*, the *sensor input* which includes the details obtained from sensors, *API specification* which includes the commands and their explanations in Python that the agent can use as described in Sec. III-B, *API implementation* which is the actual implementation of the API specification, and the *initial orientation* of the robot. We assume the initial orientation is predetermined, as the blind user can adjust it in place. We used the GPT-4 model for the LLM, which is provided via a public API. For the initial setup of the prompt, instructions from five participants in the online study were used as references to construct the prompt for the proposed systems. The remaining were used for the evaluation in Sec. VI. The implementation overview is shown in Fig. 4.

A. Parsing Instruction

The system first parses a natural language instruction to step-by-step instructions using LLM. This was done to organize our benchmark's diverse natural language instruction and make it more interpretable before generating navigation codes. To achieve this, we prompt LLMs with a set of rules they should follow, such as the requirement to describe when and which turning point to turn, and which object the robot should detect, along with examples of possible input and expected output. After parsing, each navigation step is returned as a brief sentence. We employ a two-stage prompting process with the LLM to generate more accurate outputs. Initially, we prompt LLM to provide a thought to guide the generation of the first output. Following this, we prompt LLM to refine the output by incorporating a second thought, leading to the finalized output.

B. Navigation Code Generation

After the parsed instructions have been obtained, the LLM now generates the navigation codes in Python. We prompt LLM with an API specification that includes a range of commands for robot operations (*e.g.*, `move_forward(distance)` function). These commands are complete with docstrings of their usage explanation [25], [26] and instructions to generate Python codes that follow the provided specification. For `detect_from_RGB_image(object)`, our model uses an open vocabulary object detector internally (Sec. III-B) and flexibly determines which object to detect by generating an object argument. The API specification was formatted to the similar format of the previous work [25], [26], but with additional notes, such as how each function should be and not be used. We employ the same two-stage refining method as in the parsing stage. Finally, we execute the generated code using the API implementation, given the initial orientation.

TABLE II: **Performance of VLN Models.** We compare our method with state-of-the-art VLN models that fulfill the requirements relevant to our scenario.

Condition			Online Study Data				Onsite Study Data			
Method	Parser	Route	SR↑	OSR↑	SPD↓	CLS↑	SR↑	OSR↑	SPD↓	CLS↑
NavGPT		University R1	0.04	0.09	37.54	0.05	0.02	0.04	40.58	0.05
NaVid		University R1	0.00	0.00	35.73	0.03	0.00	0.00	36.67	0.02
Proposed		University R1	0.20	0.24	**17.01**	**0.56**	0.23	0.33	21.04	0.46
Proposed	✓	University R1	**0.30**	**0.35**	19.66	0.49	**0.30**	**0.38**	**18.32**	**0.54**
NavGPT		University R2	0.00	0.00	162.10	0.01	0.00	0.00	161.13	0.01
NaVid		University R2	0.00	0.00	149.79	0.00	0.00	0.00	151.47	0.00
Proposed		University R2	0.00	0.00	93.66	0.32	**0.03**	**0.03**	117.52	0.20
Proposed	✓	University R2	**0.04**	**0.04**	**81.59**	**0.38**	**0.03**	**0.03**	**98.13**	**0.29**
NavGPT		Museum 5F R1	0.00	0.00	50.76	0.00	0.00	0.00	51.30	0.00
NaVid		Museum 5F R1	0.00	0.00	54.59	0.01	0.00	0.00	55.50	0.01
Proposed		Museum 5F R1	0.11	0.20	35.46	0.44	0.02	0.02	43.35	0.32
Proposed	✓	Museum 5F R1	**0.20**	**0.26**	**26.71**	**0.60**	**0.05**	**0.07**	**29.14**	**0.54**
NavGPT		Museum 5F R2	0.00	0.07	37.47	0.07	0.00	0.07	35.67	0.06
NaVid		Museum 5F R2	0.00	0.00	43.74	0.01	0.00	0.00	43.82	0.01
Proposed		Museum 5F R2	**0.05**	0.18	23.17	0.25	0.00	**0.02**	29.08	**0.37**
Proposed	✓	Museum 5F R2	**0.05**	**0.32**	**16.43**	**0.29**	0.00	**0.02**	**24.78**	**0.37**
NavGPT		Museum 7F R1	0.00	0.00	54.70	0.08	0.00	0.00	36.00	0.14
NaVid		Museum 7F R1	0.00	0.00	73.76	0.06	0.00	0.00	71.30	0.07
Proposed		Museum 7F R1	**0.02**	**0.02**	55.99	0.31	0.05	0.05	46.67	0.42
Proposed	✓	Museum 7F R1	**0.02**	**0.02**	**42.59**	**0.48**	**0.09**	**0.09**	**25.22**	**0.67**
NavGPT		Museum 7F R2	0.00	0.00	61.64	0.01	0.00	0.00	60.43	0.01
NaVid		Museum 7F R2	0.00	0.00	67.39	0.02	0.00	0.00	69.18	0.00
Proposed		Museum 7F R2	**0.15**	0.26	47.41	0.17	0.00	0.10	52.81	0.16
Proposed	✓	Museum 7F R2	0.07	**0.35**	**36.78**	**0.25**	**0.02**	**0.12**	**46.74**	**0.22**

VI. Experiment

A. Baselines

In our scenario, VLN agents are expected to demonstrate strong transferability, as blind users may navigate across diverse unseen locations by asking sighted passersby for directions. To evaluate this capability, we compare our model with two prior state-of-the-art methods that leverage foundation models and exhibit strong zero-shot performance: NavGPT [11] and NaVid [12].

NavGPT [11] demonstrates strong zero-shot transfer capability by leveraging an LLM, visual foundation model, and an object detector to iteratively select destinations within a navigation graph until the agent determines it has reached the goal. We used GPT-4o-mini for the LLM and for the visual foundation model, and the same Grounding DINO [32] for the object detector. As NavGPT requires a navigation graph to operate, we constructed navigation graphs over the environments following the R2R dataset [1].

NaVid [12], a state-of-the-art VLN model that demonstrates strong generalization to unseen environments, employs a visual foundation model and operates without a navigation graph, relying solely on camera input, similar to our model. We strictly controlled the agent by following NaVid's established pipeline. We initialized its weights using an open-sourced checkpoint.

B. Metrics

For the metrics, we employ success rate (SR), oracle success rate (OSR), and shortest path distance (SPD) [1], [34], and coverage weighted by length score (CLS) [35].

TABLE III: **Effect of Instruction Refinement.** While in most cases refining instruction leads to an increase in performance, in certain cases, it was not always the case, due to redundant referral to surrounding objects.

Condition		Online Study Data				Onsite Study Data			
Route	Iteration	SR↑	OSR↑	SPD↓	CLS↑	SR↑	OSR↑	SPD↓	CLS↑
University R1	1	**0.43**	**0.48**	**16.37**	**0.56**	0.25	**0.40**	20.41	0.52
University R1	2	0.17	0.22	22.94	0.41	**0.35**	0.35	**16.23**	**0.56**
University R2	1	**0.04**	**0.04**	86.82	0.36	0.00	0.00	**93.32**	**0.31**
University R2	2	**0.04**	**0.04**	**76.36**	**0.41**	**0.05**	**0.05**	102.95	0.28
Museum 5F R1	1	**0.26**	**0.30**	**25.80**	**0.60**	0.00	0.00	**27.21**	**0.56**
Museum 5F R1	2	0.13	0.22	27.61	0.61	**0.10**	**0.14**	31.07	0.52
Museum 5F R2	1	**0.05**	0.27	17.77	0.27	0.00	0.00	25.75	0.32
Museum 5F R2	2	**0.05**	**0.36**	**15.09**	**0.30**	0.00	**0.05**	**23.82**	**0.43**
Museum 7F R1	1	0.00	0.00	46.57	0.43	**0.09**	**0.09**	**23.77**	**0.68**
Museum 7F R1	2	**0.05**	**0.05**	**38.61**	**0.53**	**0.09**	**0.09**	26.66	0.65
Museum 7F R2	1	0.00	0.26	37.24	**0.26**	0.00	0.05	**46.71**	0.21
Museum 7F R2	2	**0.13**	**0.43**	**36.33**	0.24	**0.05**	**0.19**	46.76	**0.23**

As CLS computes the similarity of the path on the graph, it requires a dense navigation graph to map the navigated trajectory onto. Thus, we divided passable corridors into 50 cm square grids to serve as nodes on a graph and mapped predicted and ground truth paths onto it to calculate this metric [36]. For routes where participants described an alternative path, we used the participant-described route as the ground truth.

VII. Results and Discussion

A. Performance of the Proposed Method

As shown in Table II, our model outperforms NavGPT and NaVid. The baselines' suboptimal performances can be attributed to two factors: their deviation from the correct direction, and their premature decision that they had reached the goal. This is due to the fact that the baselines refer to the environment at each navigation step with an LLM. For example, if NavGPT makes a mistake even once during this process, it will be challenging for the model to recover the agent back to the correct path. Additionally, NaVid tends to make unnecessary frequent turns after initially following the route correctly for several iterations, likely due to the longer sequence of turns and longer instructions in our benchmark, which NaVid was not trained to handle. In contrast, our method achieves the desired outcome through a single iteration of code generation inference, removing the need to initiate inferences at every intermediate step for instructions like *"go straight for 100m and then turn right."*. We also observed that the instruction parsing module boosted the performance of our method in most cases.

B. Difficulty of the Benchmark

In Table II, it is observed that the performances from onsite instructions tended to be lower than those from online instructions, as it is more likely for the route instructions to contain errors due to human memory, and it is harder for the system to recover from errors. Overall, our results demonstrate the difficulty of the instruction data from human memory and the value of our benchmark.

Across all routes, both our model and the baselines showed suboptimal or low performance. One major reason was the difficulty in handling the varied and inaccurate input instructions. In longer routes, participants tended to inaccurately estimate lengths for certain path segments and not include sufficient information about the destination.

Upon closer inspection, many instructions in our benchmark contained phrases that required a combined understanding of both natural language and the building's structure, which our proposed model failed to follow. One example was a phrase such as *"go along this path and turn right in the first intersection,"* which was often described at the starting point of University R1. The instruction skips the right turn in the first turning point by describing it as *"go along this path,"* because the building structure only allows a right turn at the immediate corner. As a result, the instruction starts by describing the first left turn where there are two possible directions to proceed. Similar instructions also frequently appeared to describe the third and fourth turns in University R2. While humans can naturally follow such instructions, our model was unable to handle this data, as it explicitly takes into account turning points. This variation in the levels of topological details further highlights the difficulty of our benchmark, which imitates real-world scenarios of blind people seeking navigation instructions.

C. Effect of Refining Instruction

In Table III, we report the performance of our proposed method across different instruction iterations. The first iteration corresponds to the most natural, memory-based instruction, while the second represents memory-based instructions that are more accurate and contain features that may better assist the VLN agent in navigation. Generally, we found that refining the instruction led to a slight performance improvement. However, surprisingly, it was not always the case, particularly Museum 5F R1 and University R1 of the online study data. This happened because participants tended to describe more objects during the second iteration, which contained greater variation in object descriptions. As such, the two routes contained much richer object landmarks along the routes for participants to describe. For example, one participant described only turning point-related information at the first iteration, while in the second iteration, the participant also described objects to ignore, such as, *"then, you will come across an intersection with a door on the left and an intersection with doors on both sides, but ignore them and continue straight ahead."* Nonetheless, the tendency of refined instructions to yield better performance suggests that, when deploying VLN-equipped robots, it is beneficial to assist sighted passersby in recalling routes more and in conveying environmental information in a format that is more compatible with robotic interpretation.

VIII. Conclusion and Future Work

This work proposed Memory-Maze and a VLN model. Our VLN model operates with a single inference, can achieve zero-shot transfer, and can operate without a prebuilt map. It is a first step towards utilizing VLN technology for

navigation robots for blind people. Our experiment demonstrated that our model could complete the task while also outperforming prior methods. The results also reflect the difficulty of our benchmark. More importantly, we found that realistic instructions collected in the onsite environment, where participants had to rely on human memories, were longer with greater variation in words, and contained more errors compared to the instructions collected online. Through the study with the baseline and our proposed VLN model, we also observed a gap between the instructions collected onsite and those collected in traditional settings. Upon qualitative inspection, we observed evidence of the tendency for onsite instruction to be more difficult for the model to handle, such as the ones that required understanding of *"go along this path."* This suggests that future works on VLN should consider a more adaptive map representation where nodes and turns are not strictly defined, or a more flexible approach to accommodate varying topological descriptions.

For future work, we aim to implement additional modules that detect potential errors in the navigation so that the user can ask for the route again. Also, while in this study, we only used instructions from participants, for future practical usage, we aim to also explore the interactive aspect with users and robots. For example, the method could also benefit a blind user, which could guide the instruction from passersby to be better or rephrase it themself, potentially leading to better performance of the robot. Additionally, we are in the process of including more locations to our benchmark. We believe that our benchmark based on the scenario of a navigation robot for blind people provides a new perspective to the VLN task, as it contains greater variety in descriptions, is of higher difficulty, and most importantly, reflects a real-world application scenario.

IX. Theme Relevance

This paper seeks to bring attention to human understanding of route instructions in visual language navigation (VLN), particularly in scenarios where the robot seeks route instructions from humans in the real world. In traditional VLN settings, route instructions are annotated by humans with perfect knowledge of the routes, as the human annotators have the luxury of experiencing the routes repeatedly to refresh their memories. Additionally, traditional route instructions are shorter and are mostly limited to small, simpler spaces such as houses or offices. By collecting route instruction data from the real world, we discovered that memory-based instructions often contain ambiguities and errors. However, humans may be able to detect these errors in the instructions, whereas existing leading VLN models fail to do so. Additionally, existing VLN models struggle to follow long route instructions in large public spaces. This is because existing models lack a recovery mechanism should they fail while following the routes.

Our paper contributes to the workshop's theme by adding discussion to the following topics. **AI alignment**: We observed a significant difference between VLN route instruction data obtained in the traditional idealized setting and those obtained in the real-world memory-based setting. Future VLN models need to address this gap. **Data accessibility**: We proposed a new paradigm to collect VLN route instruction data more aligned with human interpretations. **Applications in human-robot interaction**: We based our studies on the real-world scenario where a blind person uses a navigation robot to obtain route instructions from passersby and proceed towards a destination, when they enter an area they have never been to before.

ACKNOWLEDGEMENT

We deeply thank Hironobu Takagi for engaging in the discussion of this project. We also thank Miraikan staff members for their support of our studies. This work was supported by JSPS KAKENHI No. 23KJ2048 and 21H05054.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018.

[2] J. Guerreiro, D. Sato, S. Asakawa, H. Dong, K. M. Kitani, and C. Asakawa, "Cabot: Designing and evaluating an autonomous navigation robot for blind people," in *ASSETS*, 2019.

[3] S. Liu, A. Hasan, K. Hong, R. Wang, P. Chang, Z. Mizrachi, J. Lin, D. L. McPherson, W. A. Rogers, and K. Driggs-Campbell, "Dragon: A dialogue-based robot for assistive navigation with visual language grounding," *RA-L*, 2024.

[4] S. Kayukawa, D. Sato, M. Murata, T. Ishihara, A. Kosugi, H. Takagi, S. Morishima, and C. Asakawa, "How users, facility managers, and bystanders perceive and accept a navigation robot for visually impaired people in public buildings," in *RO-MAN*, 2022.

[5] M. Kuribayashi, T. Ishihara, D. Sato, J. Vongkulbhisal, K. Ram, S. Kayukawa, H. Takagi, S. Morishima, and C. Asakawa, "Pathfinder: Designing a map-less navigation system for blind people in unfamiliar buildings," in *CHI*, 2023.

[6] K. Müller, C. Engel, C. Loitsch, R. Stiefelhagen, and G. Weber, "Traveling more independently: A study on the diverse needs and challenges of people with visual or mobility impairments in unfamiliar indoor environments," *TACCESS*, 2022.

[7] M. Kuribayashi, K. Uehara, A. Wang, S. Morishima, and C. Asakawa, "Wanderguide: Indoor map-less robotic guide for exploration by blind people," in *CHI*, 2025.

[8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *CoRL*, 2017.

[9] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *CVPR*, 2019.

[10] W. Wu, T. Chang, X. Li, Q. Yin, and Y. Hu, "Vision-language navigation: A survey and taxonomy," *NCAA*, 2023.

[11] G. Zhou, Y. Hong, and Q. Wu, "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," in *AAAI*, 2024.

[12] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *RSS*, 2024.

[13] S. Cai, A. Ram, Z. Gou, M. A. W. Shaikh, Y.-A. Chen, Y. Wan, K. Hara, S. Zhao, and D. Hsu, "Navigating real-world challenges: A quadruped robot guiding system for visually impaired people in diverse environments," in *CHI*, 2024.

[14] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *CoRL*, 2020.

[15] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. v. d. Hengel, "Reverie: Remote embodied visual referring expression in real indoor environments," in *CVPR*, 2020.

[16] H. De Vries, K. Shuster, D. Batra, D. Parikh, J. Weston, and D. Kiela, "Talk the walk: Navigating new york city through grounded dialogue," *arXiv*, 2018.

[17] Z. Huang, Z. Shangguan, J. Zhang, G. Bar, M. Boyd, and E. Ohn-Bar, "Assister: Assistive navigation via conditional instruction generation," in *ECCV*, 2022.

[18] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *ECCV*, 2020.
[19] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," in *CoRL*, 2021.
[20] D. Shah, B. Osiński, S. Levine, *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *CoRL*, 2023.
[21] R. Schumann, W. Zhu, W. Feng, T.-J. Fu, S. Riezler, and W. Y. Wang, "Velma: Verbalization embodiment of llm agents for vision and language navigation in street view," in *AAAI*, 2024.
[22] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *ICML*, 2022.
[23] P. Chen, X. Sun, H. Zhi, R. Zeng, T. H. Li, G. Liu, M. Tan, and C. Gan, "a^2 nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models," *6th Robot Learning Workshop: Pretraining, Fine-Tuning, and Generalization with Large Scale Models, NeurIPS*, 2023.
[24] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *ICRA*, 2023.
[25] H. Biggie, A. N. Mopidevi, D. Woods, and C. Heckman, "Tell me where to go: A composable framework for context-aware embodied robot navigation," in *CoRL*, 2023.
[26] D. Surís, S. Menon, and C. Vondrick, "Vipergpt: Visual inference via python execution for reasoning," in *ICCV*, 2023.
[27] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets, "Waypoint models for instruction-guided navigation in continuous environments," in *CVPR*, 2021.
[28] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IROS*, 2004.
[29] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, *et al.*, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv*, 2017.
[30] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *3DV*, 2017.
[31] Autodesk, "Autodesk fusion: More than cad, it's the future of design and manufacturing," Retrieved in February, 2024 from https://www.autodesk.com/products/fusion-360/overview, 2024.
[32] 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," in *ECCV*, 2024.
[33] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *EMNLP*, 2020.
[34] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. Wang, "Vision-and-language navigation: A survey of tasks, methods, and future directions," in *ACL*, 2022.
[35] V. Jain, G. Magalhaes, A. Ku, A. Vaswani, E. Ie, and J. Baldridge, "Stay on the path: Instruction fidelity in vision-and-language navigation," in *ACL*, 2019.
[36] G. Ilharco, V. Jain, A. Ku, E. Ie, and J. Baldridge, "General evaluation for instruction conditioned navigation using dynamic time warping," *arXiv*, 2019.