MapGPT: Map-Guided Prompting with Adaptive Path Planning for Vision-and-Language Navigation

Jiaqi Chen[1] **Bingqian Lin**[2] **Ran Xu**[3] **Zhenhua Chai**[3]
Xiaodan Liang[2] **Kwan-Yee K. Wong**[1]
[1]The University of Hong Kong
[2]Shenzhen Campus of Sun Yat-sen University [3]Meituan
Project: https://chen-judge.github.io/MapGPT/

Abstract

Embodied agents equipped with GPT as their brains have exhibited extraordinary decision-making and generalization abilities across various tasks. However, existing zero-shot agents for vision-and-language navigation (VLN) only prompt GPT-4 to select potential locations within localized environments, without constructing an effective "global-view" for the agent to understand the overall environment. In this work, we present a novel **map**-guided **GPT**-based agent, dubbed **MapGPT**, which introduces an online linguistic-formed map to encourage global exploration. Specifically, we build an online map and incorporate it into the prompts that include node information and topological relationships, to help GPT understand the spatial environment. Benefiting from this design, we further propose an adaptive planning mechanism to assist the agent in performing multi-step path planning based on a map, systematically exploring multiple candidate nodes or sub-goals step by step. Extensive experiments demonstrate that our MapGPT is applicable to both GPT-4 and GPT-4V, achieving state-of-the-art zero-shot performance on R2R and REVERIE simultaneously (~10% and ~12% improvements in SR), and showcasing the newly emergent global thinking and path planning abilities of the GPT.

1 Introduction

Large language models (LLMs) (Touvron et al., 2023a,b; Chowdhery et al., 2022; Anil et al., 2023) have demonstrated strong performance in various domains. As the most powerful LLMs, the GPT series models (Brown et al., 2020; OpenAI, 2023a,b,c) can even serve as the brains of embodied agents (Huang et al., 2023; Mu et al., 2023; Ahn et al., 2022), enabling them to engage in explicit thinking and decision-making process. Moreover, these GPT-based agents are typically zero-shot or few-shot, eliminating the burdensome tasks of data annotation and model training, and they also demonstrate remarkable sim-to-real abilities.

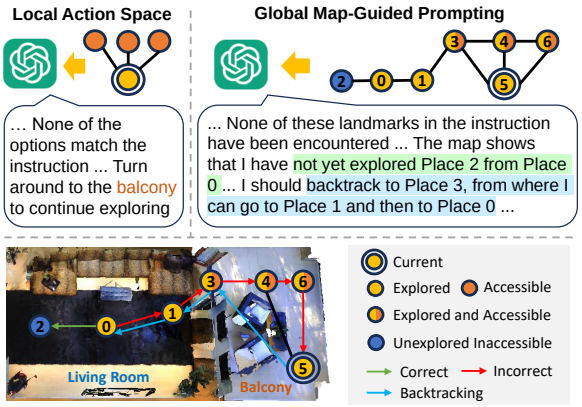


Figure 1: A comparison of the thinking process of the GPT agent without and with topological maps. Given only a local action space, the agent may explore aimlessly, especially when navigation errors have already occurred. Incorporating topological maps enables the agent to understand spatial structures and engage in global exploration and path planning.

Recently, LLMs have also been adopted in vision-and-language navigation (VLN), where agents are given human instructions that require them to visually perceive and navigate an indoor environment. Previous learning-based VLN methods (Anderson et al., 2018b; Fried et al., 2018; Qi et al., 2020a; Chen et al., 2022; Qiao et al., 2022; Wang et al., 2023) relied on the training on large-scale domain-specific data with expert instruction annotations to execute navigation tasks. To address their reliance on training data and potential sim-to-real gap, some GPT-based zero-shot agents (Zhou et al., 2023; Long et al., 2023) with explainable decision-making abilities have been proposed. These methods translated visual observations into textual prompts and required GPT-4 to act as an agent to select the correct position or direction. Besides, they also employed multiple additional experts to handle various subtasks in text form, such as summarizing the history and fusing repeated expert predictions.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9796–9810

However, such zero-shot VLN agents face some challenges. Firstly, these methods are developed for a multi-expert pure language system based on GPT-4, necessitating the conversion of visual observations into text and multiple rounds of text summarizations, which inevitably lead to information loss and hinder the application to multimodal LLMs (e.g., GPT-4V). More importantly, these agents make decisions based solely on the observations of the local environment. As shown in Figure 1 (left), given only the local action space, when an agent realizes that it has engaged in an erroneous exploration, it can only continue to explore surrounding environment aimlessly.

In this paper, we propose MapGPT which contains a topological map in the linguistic form to assist in global exploration and adaptive path planning. We first develop a simple yet efficient prompt system with only one navigation expert that can be applied to both GPT-4 and multimodal GPT-4V flexibly. To encourage global exploration, we propose a map-guided prompting method for the GPT model, to build a "global-view" for the agent. Specifically, for an online constructed map, we have discarded the precise GPS coordinates that are difficult for GPT to understand, while preserving the topological relationships between nodes and incorporating them into prompts to assist in understanding the navigation environment, as shown in Figure 1 (right). Given this tailored map, we further propose an adaptive planning mechanism to activate GPT's multi-step path planning ability. Instead of documenting the thinking process of each step as in previous works, MapGPT generates a multi-step planning similar to a human work plan, and updates it iteratively to achieve strategic exploration of potential objectives. Benefiting from this, the agent can adaptively perform path planning based on the map, systematically explore multiple candidate nodes or sub-goals step by step, and backtrack to a specific node for re-exploration when necessary.

We conduct experiments on two popular VLN benchmarks, namely R2R (Anderson et al., 2018b) and REVERIE (Qi et al., 2020b), which contain step-by-step and high-level instructions respectively. Experimental results show the superiority of MapGPT over existing zero-shot VLN agents. Especially in REVERIE, MapGPT exhibits enhanced competitiveness (31.6% SR), surpassing even some learning-based methods trained on REVERIE. Extensive ablation studies reveal the advantage of our introduced map and adaptive path planning mechanism in encouraging systematic exploration to improve navigation.

Our contributions can be summarized as follows.

- We propose a novel map-guided prompting method, which introduces an online linguistic-formed map including node information and topological relationships to encourage GPT's global exploration.
- An adaptive planning mechanism is utilized to activate GPT's multi-step path-planning ability, enabling systematic exploration of multiple potential objectives.
- MapGPT can be applied to both GPT-4 and GPT-4V and is more unified as it can adapt to varying instruction styles effortlessly, achieving state-of-the-art zero-shot performance on both the R2R and REVERIE datasets.

2 Related Work

Vision-and-Language Navigation (VLN) As a representative multi-modal embodied AI task, VLN requires an agent to combine human instructions and visual observations to navigate and locate targets in real-world scenes. Previous learning-based approaches (Wang et al., 2019; Ma et al., 2019; Deng et al., 2020; Qi et al., 2020a) proposed various model architectures and trained their models on domain-specific datasets. Besides, pretrained models (Hong et al., 2021; Chen et al., 2021b, 2022; Qiao et al., 2022; Guhur et al., 2021; An et al., 2022; Lin et al., 2022; Qiao et al., 2023; Wang et al., 2023; Pan et al., 2023) have been widely applied to produce better multi-modal representations. Recently, to address the reliance on domain-specific data and the possible sim-to-real gap, some zero-shot agents based on GPT (Zhou et al., 2023; Long et al., 2023) have been proposed. However, they suffer from several limitations. For example, NavGPT (Zhou et al., 2023) has limited performance and relies on a two-stage language-only system. DiscussNav (Long et al., 2023) introduces a sequential multi-experts system to discuss and summarize various information and fuses five repeated predictions to improve performance. Some of their designs limit the agent's capability to only address step-by-step instructions in the R2R dataset (Anderson et al., 2018b), and have not been validated on other styles of instructions (e.g., REVERIE (Qi et al., 2020b)). Besides, these agents are limited to local exploration as they can only reason and

make decisions within adjacent navigable points. In this paper, we propose a map-guided prompting method with adaptive path planning for global exploration, achieving impressive performance on both R2R and REVERIE.

Large Language Models (LLMs) LLMs (OpenAI, 2023a,b,c; Chiang et al., 2023; Touvron et al., 2023a,b; Anil et al., 2023) have demonstrated remarkable capabilities in multiple domains. Recently, LLM-based agents (Huang et al., 2023; Mu et al., 2023; Ahn et al., 2022; Pan et al., 2023; Brohan et al., 2023; Schumann et al., 2023; Hu et al., 2023; Lin et al., 2024) have also attracted significant interest of the community. For example, VoxPoser (Huang et al., 2023) utilizes LLM and vision-language models to extract affordances and constraints, which enables motion planners to generate trajectories for manipulation. LangNav (Pan et al., 2023) employs LLMs for navigation, but it merely utilizes GPT-4 (OpenAI, 2023a) to synthesize some data and performs fine-tuning using Llama2 (Touvron et al., 2023b) as the backbone, rather than directly employing LLM as a zero-shot agent. In fact, the application of LLM-based zero-shot agents in navigation tasks is still limited, and how to prompt LLMs (including multimodal GPT-4V) to activate global thinking and planning abilities required by navigation task have yet to be explored (Yang et al., 2023).

Maps for Navigation Maps used for navigation tasks can be primarily categorized into two types, i.e., metric maps and topological maps. Employing SLAM (Fuentes-Pacheco et al., 2015) for constructing metric maps (Chaplot et al., 2020; Thrun, 1998) is widely used in navigation. However, this type of approach requires a trade-off between map size and computational efficiency, which affects navigation performance. To address this limitation, graph-based topological maps (Chen et al., 2021a, 2022; An et al., 2022) have been proposed for pre-exploring environment or enabling global exploration, such as backtracking to previously visited nodes. However, these methods are all designed for model learning. It remains unexplored how to build a map with prompts and leverage the powerful capabilities of LLMs for zero-shot reasoning and planning based on the map.

3 Method

In this section, we introduce our newly designed prompt system (Sec. 3.1), the details of our map-guided prompting method to help the agent understand global environment (Sec. 3.2), and the novel adaptive mechanism that encourages the agent to make multi-step path planning (Sec. 3.3).

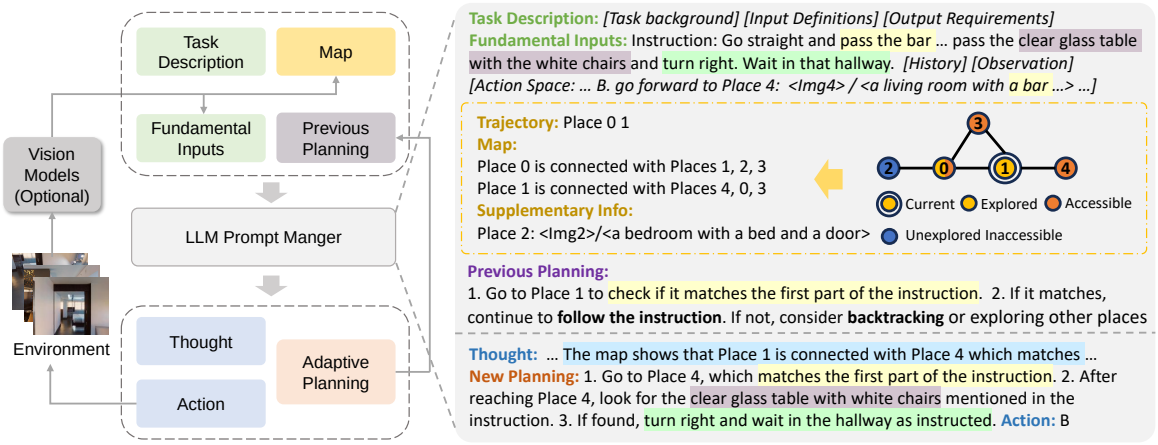
3.1 Single Expert Prompt System

Previous works such as NavGPT (Zhou et al., 2023) and DiscussNav (Long et al., 2023) are two-stage systems. They first gather visual observations from all the views and translate them into textual descriptions which are then fed into language-only GPT-4 (OpenAI, 2023a) for decision-making. Besides, they rely on complex multi-expert designs, where GPT plays different roles to achieve various functions, such as instruction parsing, summarizing textual descriptions and history, etc. However, these intricate designs are only geared towards text processing, which limits their research value. Powerful multimodal large models, such as GPT-4V (OpenAI, 2023b,c), can directly serve as the agent's brain to process multimodal information and make decisions (Yang et al., 2023).

Compared with previous works, our proposed single expert prompt system has several unique features. (1) We eliminate the need for a separate design of an additional historical summary expert or instruction decomposing expert based on GPT models, which makes it convenient to incorporate both visual/textual inputs and additional information, such as maps. (2) Our navigation expert can utilize GPT-4V to make decisions directly based on visual observations in one stage. It can also take text descriptions as inputs and flexibly apply them in the two-stage GPT-4 system. (3) Simple yet efficient. In the R2R dataset, our two-stage system requires an average of 672 input tokens and 115 output tokens per step. In comparison, NavGPT utilizes three GPT experts, resulting in an average cost of 2,465 input tokens and 317 output tokens per step.

As shown in Figure 2, we collect various fundamental inputs for the agent, including instruction I, history H_t, observation O_t, and action space A_t. The meaning of these inputs, as well as the requirements for output, are clearly pre-defined in the task description D. We utilize a prompt manager PM to organize these prompt inputs which are then fed into the large language model LLM to generate the current thought T_t and select a specific action $a_{t,i} \in A_t$. The pipeline can be formulated as

$$T_t, a_{t,i} = LLM(PM(D, I, H_t, O_t, A_t)). \quad (1)$$



(a) Prompt System (b) Map-Guided Prompting with Adaptive Path Planning

Figure 2: Our basic system consists of two types of prompts, namely task description and fundamental inputs. We introduce a map-guided prompting method that builds an online-constructed topological map into prompts, activating the agent's global exploration. We further propose an adaptive mechanism to perform multi-step path planning on this map, systematically exploring candidate nodes or sub-goals. Note that vision models are optional, and viewpoint information can be represented using either the image or textual description of the observations.

3.1.1 Task Description

As shown in Figure 2, our prompts for task description D consist of three main parts. We set the task background for GPT, provide some definitions regarding the inputs at each step, and propose some basic requirements regarding how the agent should accomplish this task.

3.1.2 Fundamental Inputs

Instruction I Since we aim to adapt both fine-grained and high-level instructions, we directly feed the raw instructions to GPT without any initial analysis or decomposition.

Visual Observation O_t Our agent is equipped with an RGB camera to capture M images of the environment. Unlike NavGPT (Zhou et al., 2023) that gathers excessive environmental information, we prioritize the observations of navigable points, which can be formulated as $O_t = \{o_{t,i}\}_{i=1}^{N}$, where N is the number of navigable viewpoints at step t. Each $o_{t,i}$ represents the observation towards a specific navigable point. These observations are original images for a one-stage system based on GPT-4V and can also be replaced with caption and detection results in a two-stage system, where we follow NavGPT (Zhou et al., 2023) and utilize off-the-shelf vision models BLIP-2 (Li et al., 2023a) and Faster R-CNN (Ren et al., 2015; Anderson et al., 2018a) to acquire scene description and object detection respectively. NavGPT also utilizes the bounding boxes provided by the REVERIE dataset to extract an additional object list of all the views, which is then used for the R2R experiment. We also utilize this object list and name it the "Surroundings". However, we do not utilize this additional information in R2R. We only utilize it to enable the agent to determine whether to stop in the REVERIE dataset since we do not specify the directions of these surrounding objects.

Action Space A_t NavGPT allows the agent to directly select a viewpoint and DiscussNav enables the model to predict a direction. In this paper, we incorporate both directional phrases (e.g., turn left to) and the corresponding observations of each direction into the action space for GPT to select from, making the decision-making process more intuitive.

Specifically, we take as an input the action space $A_t = \{a_{t,i}\}_{i=0}^{N}$ for GPT model. N is also the number of navigable points and we additionally define $a_{t,0}$ as *"A. stop"* so that the agent has $N+1$ options in total. Each $a_{t,i}$ in the remaining N options is formulated using the template:

"{label} {direction} {$o_{t,i}$ }".

At each step t, the agent only needs to choose one option $a_{t,i} \in A_t$. For the output format, we require the agent to simply provide a single option label, such as *"Action: B"*.

History H_t We record all previous actions $a_0 \sim a_{t-1}$ for history. The following prompt template is utilized for appending the actions into H_t:

"step 0: {a_0^*}, ..., step t-1: {a_{t-1}^*}",

in which $t \geq 1$ and a^* denotes the selected action a but with the option label removed. The initial history is defined as H_0 = *"The navigation has just begun, with no history"*.

3.2 Map-Guided Prompting

For the VLN task, previous work (Zhu et al., 2021; Chen et al., 2022; An et al., 2023) has demonstrated the effectiveness of online constructed maps for global navigation. However, how to construct the maps and transform them into a certain form to prompt LLMs has not been investigated in this domain. Additionally, we observe that GPT-4V struggles slightly in its attempts to understand navigation environments based on multiple precise coordinates. Therefore, we propose a novel map-guided prompting approach that converts topological relationships of a map into textual prompts, as shown in Figure 2(b).

Topological Mapping In the VLN task, the agent has never explored the entire environment and must construct a map based on its own observations online. We store the map as a dynamically updated graph, following the graph-based method DUET (Chen et al., 2022). At each step t, we record all observed nodes along the navigation trajectory and their connectivity into the graph $G_t = \{V_t, E_t\}$, where $V_t = \{v_{t,i}\}_{i=1}^{K}$ are a series of K observed nodes marked with the index i in the order of observations. All the edges between these observed nodes are recorded in E_t. At any step t, given the current location, the simulator will provide several neighboring nodes that are currently navigable. These new nodes and edges will be utilized for updating the graph from G_{t-1} to G_t.

3.2.1 Constructing Maps with Prompts

After we have obtained a topological connectivity graph representing the structure of the environment, the next step is to transform it into an appropriate prompt and add map annotations to form a complete map-guided prompt to help the agent understand the navigation environment. For each step t, we categorize all observed nodes in the environment into three types, namely (1) explored nodes $\{en_j\}_{j=0}^{t}$ (including starting node en_0 and current node en_t), (2) accessible nodes $\{an_t^0, an_t^1, ...\}$, and (3) unexplored inaccessible nodes $\{un_0, un_1, ...\}$.

Trajectory As we have already marked each location during the navigation process, it follows logically that we can create a simplified trajectory prompt to help the agent understand its navigation path in the map and avoid repeated exploration as far as possible. For the explored nodes, we formulate our trajectory prompt using the template:

"Trajectory: Place {en_0} ... {en_t}",

where each en_j corresponds to a node $v_{t,i} \in V_t$ stored in the order of observation. Thus, we consider the index i as the ID of the node and fill it into the template to denote en_j.

Map Connectivity Unlike DUET, which also converts precise GPS coordinates into embeddings for graph learning, we only retain the topological relationships of the map nodes since we discover that it seems challenging for GPT to understand precise coordinate data. These topological relationships are transformed into textual prompts, making it easier to comprehend spatial structures. Since the connectivity can only be observed at explored nodes, we always start with *"Place {en_t} is connected with ..."*. All IDs corresponding to the neighboring accessible nodes of these exploded nodes will be listed using the following template:

"Map:
Place {en_0} is connected with Places {an_0^0}, ...
Place {en_1} is connected with Places {an_1^0}, ...
...
Place {en_t} is connected with Places {an_t^0}, ...",

where all the nodes should be filled with their node IDs. Note that this map connectivity does not need to be updated if the agent decides to backtrack and revisit some previously explored nodes.

Map Annotations The final step involves adding an annotation to each node of this topological map, enabling the agent to refer to them for path planning. As we have already provided currently accessible nodes in the action space, and the selected actions are also included in the history to form explored nodes, there is no need to repeat them. It is sufficient to simply add the node IDs in the action space at each step. Specifically, each $a_{t,i}$ in the action space A_t is reformulated as

"{label} {direction} Place {an_t^i}: {$o_{t,i}$ }".

The agent can therefore find the corresponding explored nodes and accessible nodes in history H_t and action space A_t respectively. However, we still have some unexplored and currently inaccessible nodes that are important, especially when the agent encounters obstacles in exploration and needs to revisit previous nodes for re-exploration. These inaccessible nodes are considered as supplementary information to assist the agent in backtracking to the most suitable node. In *"Supplementary Info"*

prompts, we record their raw images for the one-stage system, whereas in the two-stage system, it can be replaced with the corresponding scene descriptions.

3.3 Adaptive Path Planning

NavGPT and DiscussNav record the thinking process of the agent at each step. Despite employing another GPT expert for summarization, they still involve a significant amount of redundancy. This is also not consistent with human thinking, as we usually do not document every moment of our thoughts. Instead, we tend to document a work plan and update it as necessary.

Inspired by the above insight and benefiting from the utilization of maps, we propose an adaptive planning module that demands the agent to dynamically generate and update its multi-step path planning at each step. Concretely, the agent is required to combine the thought, map, and previous planning to adaptively update a new multi-step path planning. The entire process is iterative, where the planning output of the current step serves as the input to the next step, allowing the agent to refer to previous plans. Therefore, at step t, the agent should refer to the last planning P_{t-1}, where P_0 is set as *"Navigation has just started, with no planning yet"*.

In summary, the proposed MapGPT that combines map M_t and path planning P_{t-1} can be defined as follows:

$$T_t, P_t, a_t = LLM(PM(D, I, H_t, O_t, A_t, M_t, P_{t-1})). \quad (2)$$

In addition to the widely-used thought and action on the previous agents, MapGPT outputs additional multi-step planning information. Thanks to the powerful reasoning abilities of LLMs, the agent can focus on multiple potential nodes or sub-goals during planning, instead of being limited to predicting only one optimal choice from the global action space in a probabilistic manner without interpretability, as in the previous DUET model. Furthermore, the agent can adaptively update its plan, choosing to continue exploring sub-goals or backtrack to a previous node for re-exploration, which enhances the agent's navigation performance. Some analysis of these capabilities is presented in Section 4.

Methods	LLMs	Exp	NE↓	OSR↑	SR↑	SPL↑
NavGPT (Zhou et al.)	GPT-3.5	3	8.02	26.4	16.7	13.0
MapGPT (Ours)	GPT-3.5	**1**	8.48	29.6	19.4	11.6
DiscussNav (Long et al.)	GPT-4	5	6.30	51.0	37.5	33.3
MapGPT (Ours)	GPT-4	**1**	5.80	**61.6**	41.2	25.4
MapGPT (Ours)	GPT-4V	**1**	**5.62**	57.9	**47.7**	**38.1**

Table 1: Results on 72 various scenes of the R2R dataset. "Exp" refers to the number of GPT experts.

4 Experiments

4.1 Experimental Settings

Datasets and Evaluation We choose two datasets, R2R (Anderson et al., 2018b) and REVERIE (Qi et al., 2020b), to validate our MapGPT since they have distinct instruction styles. R2R provides detailed step-by-step instructions while REVERIE only offers a high-level description of finding the target object, which usually requires more exploration in the environment. To unify the prompt system, we focus only on navigation performance, which involves finding the correct location or object to stop, while neglecting the object grounding sub-task in REVERIE. We therefore adopt several evaluation metrics for navigation, including Navigation Error (NE, the distance between agent's final location and the target location), Success Rate (SR), Oracle Success Rate (OSR, SR given Oracle stop policy), and SR penalized by Path Length (SPL).

4.2 Experimental Results

4.2.1 Results on the Room-to-Room Dataset

Various scenes. As shown in Table 1, we employ an identical sampled subset (72 scenarios, 216 trajectories) as in NavGPT's experiment to evaluate the zero-shot performance across various scenarios. In addition, some of DiscussNav's experiments fuse five repeated predictions to enhance performance. For a fair comparison, we evaluate the performance of our MapGPT, and previous NavGPT and DiscussNav under the single-prediction setting. For a two-stage system, MapGPT outperforms previous models when applied to different LLMs (including GPT-3.5 and GPT-4). When utilizing GPT-4V as a one-stage agent, combined with the proposed map-guided prompting with adaptive path planning, MapGPT achieves a success rate of 47.7%. MapGPT (GPT-4 based) has limited performance on the SPL metric, which could be attributed to the fact that map-guided prompting encourages the agent to continue global exploration when encountering insufficient textual descriptions. The

Settings	Methods	NE↓	OSR↑	SR↑	SPL↑
Train	Seq2Seq (Anderson et al., 2018b)	7.81	28	21	-
	Speaker (Fried et al., 2018)	6.62	45	35	-
	EnvDrop (Tan et al., 2019)	5.22	-	52	48
Pretrain	LangNav (Pan et al., 2023)	-	-	43	-
	PREVALENT (Hao et al., 2020)	4.71	-	58	53
	RecBERT (Hong et al., 2021)	3.93	69	63	57
	HAMT (Chen et al., 2021b)	2.29	73	66	61
	DUET (Chen et al., 2022)	3.31	81	72	60
	ScaleVLN (Wang et al., 2023)	2.09	88	81	70
ZS	NavGPT (Zhou et al., 2023)	6.46	42	34	29
	MapGPT (with GPT-4)	6.29	57.6	38.8	25.8
	MapGPT (with GPT-4V)	**5.63**	**57.6**	**43.7**	**34.8**

Table 2: Results on the validation unseen set of the R2R dataset. MapGPT outperforms two non-pretrained methods and the zero-shot NavGPT.

Settings	Methods	OSR↑	SR↑	SPL↑
Train	Seq2Seq (Anderson et al., 2018b)	8.07	4.20	2.84
	RCM (Wang et al., 2019)	14.2	9.29	6.97
	SMNA (Ma et al., 2019)	11.3	8.15	6.44
	FAST-MATTN (Qi et al., 2020b)	28.2	14.4	7.19
Pretrain	HAMT (Chen et al., 2021b)	35.4	31.6	29.6
	DUET (Chen et al., 2022)	50.0	45.8	35.3
	LAD (Li et al., 2023b)	64.0	57.0	37.9
ZS	NavGPT (Zhou et al., 2023)	28.3	19.2	14.6
	MapGPT (with GPT-4)	**42.6**	28.4	14.5
	MapGPT (with GPT-4V)	36.8	**31.6**	**20.3**

Table 3: Comparison on a randomly sampled subset from the validation unseen set of the REVERIE dataset. Note that we have retested the released HAMT and DUET on this same subset.

agent has a longer navigation path, thus impacting the SPL metric. On the other hand, MapGPT based on GPT-4V avoids the issue of information loss during vision-to-text conversion and achieves an SPL of 38.1%.

Validation unseen set. We further compare the navigation performance between the proposed MapGPT and previous NavGPT on a larger validation unseen set with 11 scenes and 783 trajectories. As shown in Table 2, due to the distribution difference, the success rate of MapGPT is slightly lower (38.8% with GPT-4 and 43.7% with GPT-4V) compared to the results on the 72 scenes. Compared to NavGPT (GPT-4 based), MapGPT (GPT-4 based) exhibits a noticeable reduction in NE, leading by 4.8% in SR, and a substantial 15.6% improvement in OSR. Obviously, our proposed map-guided prompting with adaptive planning has raised the upper limit of agent navigation ability (OSR), thus increasing final success rate. However, due to information loss in the two-stage pipeline, MapGPT (GPT-4 based) tends to continue exploring when encountering insufficient description and has a weaker SPL performance. Nevertheless, the GPT-4V-based agent does not suffer from this issue, thus achieving 34.8% in SPL.

4.2.2 Results on the REVERIE Dataset

Benefiting from the flexible single-expert system and the utilization of a universal map-guided prompting and planning, we effortlessly apply MapGPT to the REVERIE dataset with different instruction styles. Considering the API costs and easier comparison for future work, we randomly sample 500 instructions from the validation unseen set for the zero-shot setting REVERIE benchmark.

Validation unseen set. As shown in Table 3, MapGPT exhibits greater competitiveness on REVERIE (31.6% SR), significantly outperforming zero-shot NavGPT and some training-only models across all metrics. Moreover, when compared to HAMT which benefits from the pretraining and fine-tuning process, MapGPT demonstrates a highly competitive performance as well.

Backtracking ratio. We analyze the trajectory of each case, and if there are repeated visits to previously visited locations, we consider it as backtracking and calculate the ratio of such occurrences in both NavGPT and MapGPT for comparison on the REVERIE dataset. MapGPT experiences at least one instance of backtracking in 49% of cases, and among them, there is an 80% probability of successfully correcting its erroneous navigation path at least once. As a comparison, NavGPT, which does not utilize any maps and planning, performs poorly on this metric. Only 18% of cases involve backtracking to visited nodes, and among these cases, 53% of instances successfully correct the previous exploration error at least once.

4.3 Ablation Study

As shown in Table 4, we explore the effectiveness of different prompt designs on 72 various scenes of the R2R dataset.

Coordinate Maps vs. Topological Maps. DUET encodes the precise GPS coordinates of each node for graph learning. We also attempt to input these coordinates into GPT for understanding and reasoning. However, the agent appears to struggle with global exploration based on the coordinates, resulting in a performance decline compared to the baseline that does not utilize the map. On the other hand, our proposed map-guided prompting method provides the topological relationship in a

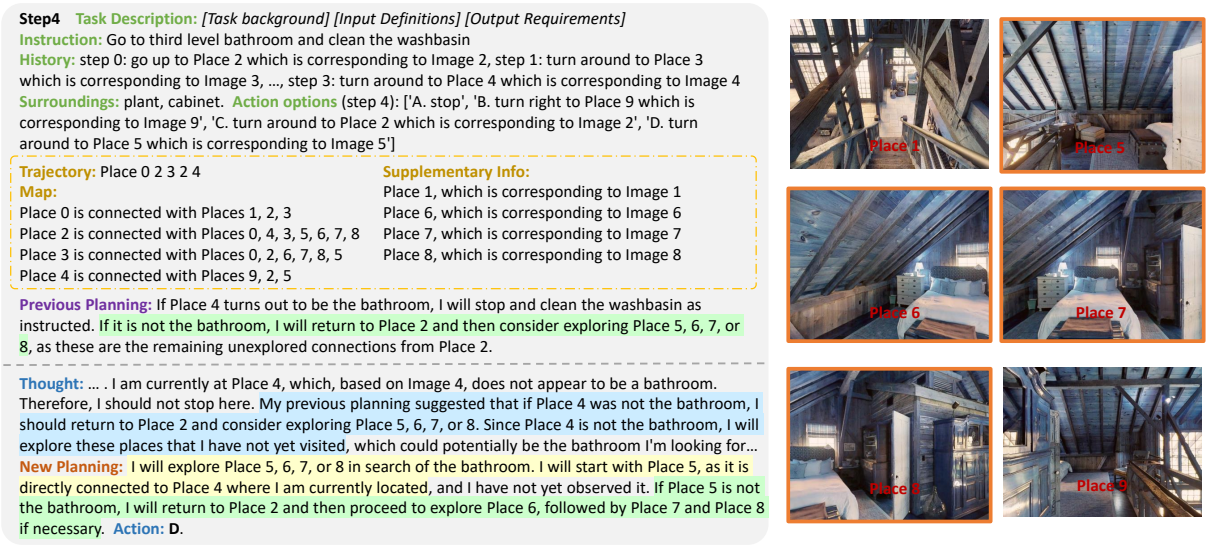


Figure 3: A successful case on REVERIE showcases MapGPT's (GPT-4V based) various abilities, including global exploration (blue), map understanding (yellow), and adaptive multi-step path planning (green). The six images on the right represent six unexplored places at step 4. Among these, MapGPT focuses on four possible places and systematically explores them until it discovers the bathroom when moving to place 8.

LLMs	Map	Planning	NE↓	OSR↑	SR↑	SPL↑
GPT-4	×	×	6.49	49.5	32.9	19.4
	Topological	×	6.40	59.7	37.5	24.8
	Topological	Adaptive	5.80	61.6	41.2	25.4
GPT-4V	×	×	5.96	58.8	42.6	34.7
	Coordinate	×	6.12	55.1	41.2	32.8
	Topological	×	5.89	56.5	44.9	36.5
	Topological	Action	5.82	58.3	45.4	35.6
	Topological	Adaptive	5.62	57.9	47.7	38.1

Table 4: Ablation of different map or planning designs on 72 various scenes of the R2R dataset.

natural language form for GPT-4 and GPT-4V to understand, leading to a significant performance improvement.

Global Action Planning vs. Adaptive Path Planning. DUET develops a global action planning, which involves selecting an optimal node from both accessible and unexplored inaccessible nodes, and teleporting the agent to that node. We also implement a similar agent that incorporates nodes from the "Supplementary Info" into the action space for action planning by GPT-4V. Experimental results indicate that this approach does not significantly improve the zero-shot agent equipped with GPT. Instead, we adopt an adaptive planning approach, where GPT explicitly outputs a segment of multi-step path planning, allowing flexible attention to multiple potential nodes or sub-goals and the ability to correct previous errors. This effectively leverages the advantages of the GPT for thinking and planning, rather than selecting a single action.

4.4 Case Study

In Figure 3, we showcase a successful example from REVERIE that demonstrates the various abilities of MapGPT (GPT-4V based). This example poses some challenges for a zero-shot agent since the places observed at place 4, namely places 5 and 9, as well as the previously observed places 1, 6, 7, and 8, do not contain the bathroom which is behind the door. Additionally, the instructions in REVERIE typically do not include information about turning or any other specific actions. Therefore, the agent needs to explore the entire environment to determine the correct direction. Benefiting from map-guided prompting with adaptive planning, MapGPT demonstrates a strong understanding of the topological relationships between nodes and adaptively performs multi-step path planning. Based on six unexplored global candidates, the agent systematically conducts global exploration by selecting the four most probable nodes, as they are situated within the bedroom and are more likely to be connected to the bathroom. Besides, the planning content mentions the possibility of backtracking to place 2 for re-exploration if necessary, and is also adaptively updated upon discovering the direct connection between places 5 and 4 in the map. After several steps, when the agent moves to place 8, it discovers the bathroom hidden behind the door and successfully reaches the destination.

5 Conclusion

In this paper, we propose a novel zero-shot agent, named MapGPT, for the VLN task. MapGPT utilizes map-guided prompting, which builds online constructed maps using prompts that provide GPT with node information and topological relationships to activate global exploration. Additionally, we propose an adaptive planning mechanism that enables multi-step path planning based on the map, allowing the agent to systematically explore potential objectives. Through extensive experiments, we demonstrate that MapGPT achieves state-of-the-art zero-shot performance with global thinking and path planning capabilities.

Limitations

Despite the significant performance gap between MapGPT and models based on pre-training and fine-tuning, zero-shot VLN still holds significant research value. GPT's pre-training corpus contains a large amount of real-world image data, thus demonstrating great potential in terms of generalization and sim-to-real transfer. However, MapGPT is only experimented within a simulator that incorporates certain ideal assumptions. Developing LLM-based agents directly in the real world and addressing various real-world challenges would be a meaningful future direction.

Acknowledgements

This work was supported in part by CAAI-Huawei MindSpore Open Fund.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. 2022. Bevbert: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*.

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. 2023. Bevbert: Multimodal map pre-training for language-guided navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2737–2748.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018a. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018b. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE.

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. 2020. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258.

Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. 2021a. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11286.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. 2021b. History aware multimodal transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34:5834–5847.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16537–16547.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. 2020. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 33:20660–20672.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 31.

Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. 2015. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81.

Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. 2021. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1634–1643.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. 2020. 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*, pages 13137–13146.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. vlnbert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1643–1653.

Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. 2023. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. 2023. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Mingxiao Li, Zehao Wang, Tinne Tuytelaars, and Marie-Francine Moens. 2023b. Layout-aware dreamer for embodied visual referring expression grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1386–1395.

Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. 2024. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *arXiv preprint arXiv:2403.07376*.

Bingqian Lin, Yi Zhu, Zicong Chen, Xiwen Liang, Jianzhuang Liu, and Xiaodan Liang. 2022. Adapt: Vision-language navigation with modality-aligned action prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15396–15406.

Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. 2023. Discuss before moving: Visual language navigation via multi-expert discussions. *arXiv preprint arXiv:2309.11382*.

Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. 2019. Self-monitoring navigation agent via auxiliary progress estimation. *arXiv preprint arXiv:1901.03035*.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2023. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *arXiv preprint arXiv:2305.15021*.

OpenAI. 2023a. Gpt-4 technical report.

OpenAI. 2023b. Gpt-4v(ision) system card.

OpenAI. 2023c. Gpt-4v(ision) technical work and authors.

Bowen Pan, Rameswar Panda, SouYoung Jin, Rogerio Feris, Aude Oliva, Phillip Isola, and Yoon Kim. 2023. Langnav: Language as a perceptual representation for navigation. *arXiv preprint arXiv:2310.07889*.

Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. 2020a. Object-and-action aware model for visual language navigation. In *European Conference on Computer Vision*, pages 303–317. Springer.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020b. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991.

Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. 2022. Hop: History-and-order aware pre-training for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15418–15427.

Yanyuan Qiao, Yuankai Qi, Zheng Yu, Jing Liu, and Qi Wu. 2023. March in chat: Interactive prompting for remote embodied referring expression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15758–15767.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Raphael Schumann, Wanrong Zhu, Weixi Feng, Tsu-Jui Fu, Stefan Riezler, and William Yang Wang. 2023. Velma: Verbalization embodiment of llm agents for vision and language navigation in street view. *arXiv preprint arXiv:2307.06082*.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of NAACL-HLT*, pages 2610–2621.

Sebastian Thrun. 1998. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6629–6638.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. 2023. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12009–12020.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1).

Gengze Zhou, Yicong Hong, and Qi Wu. 2023. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*.

Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. 2021. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12689–12699.

Appendices

A More Details

A.1 Prompts

Task Description We provide specific task description prompts that are directly fed into the system content of GPT-4V (OpenAI, 2023b,c) API. As shown in Figure 4, our unified prompts consist of three parts, namely task background, input definitions, and output requirements.

We have achieved effortless adaptation between fine-grained R2R (Anderson et al., 2018b) instructions and high-level REVERIE (Qi et al., 2020b) instructions with only a few intuitive and necessary modifications. These modifications are primarily utilized for ignoring the extensive interactive actions with objects in REVERIE, since our unified agent is designed to focus on the navigation task. The stopping conditions also differ, as R2R only requires the agent to stop at the destination, while REVERIE demands the agent to check the target object in its surroundings before stopping.

Templates All of the environmental information collected at each step (including observations, history, maps, etc.) will be incorporated into the user messages of GPT.

For a two-stage prompt system, we formulate the observations that have been converted into text as $O_t = \{o_{t,i}\}_{i=1}^{N}$, where N is the number of navigable viewpoints at step t. Each $o_{t,i}$ represents the observation towards a specific navigable point and is formulated using the prompt template:

"<{scene}>, which also includes <{objects}>".

Besides, we have defined six directional concepts, namely *"go forward to"*, *"turn left to"*, *"turn right to"*, *"turn around to"*, *"go up to"*, and *"go down to"*, according to the directions of navigable viewpoints. Thus, a complete action a_t could be *"B. turn around to Place 1: <a room with blue walls>, which also includes <bed, curtain, picture>"*.

For a one-stage system based on GPT-4V, we directly require the agent to refer to images corresponding to various places. At the beginning of the user messages, we input the image IDs and observed images in an interleaved format, such as *"Image 0: <Img0> Image 1: <Img1> ..."*. Therefore, these images can be directly referenced in subsequent prompts. For example, when used in the action space, an action a_t could be *"C. turn around to Place 2 which is corresponding to Image 2"*. Prompts in history and maps also employ similar templates.

A.2 Implementation Details

We conduct experiments in the Matterport3D simulator (Chang et al., 2017), which provides a discrete navigation environment with predefined navigable viewpoints. At each viewpoint, the agent can obtain visual observations and some connected navigable candidate viewpoints which are incorporated into prompts for GPT. Once GPT has selected one of these candidates by predicting the corresponding label in the prompts, we can convert it into the candidate's ID which can be executed in the simulator and teleport the agent to the selected viewpoint.

In this work, we have built the first GPT-4V-based (OpenAI, 2023b,c) agent in the VLN field, directly processing multimodal inputs in one stage. For a fair comparison with previous methods specifically designed for GPT-4 (OpenAI, 2023a), we have also implemented a two-stage system, where we follow NavGPT (Zhou et al., 2023) and utilize BLIP-2 (Li et al., 2023a) to provide a caption for the observation, and employ Faster R-CNN (Ren et al., 2015) to detect existing objects. Our core contributions, namely map-guided prompting and adaptive path planning, can be applied to both of these systems. To adapt our MapGPT to the REVERIE dataset, we only make some simple yet necessary modifications, which demonstrates that our MapGPT is more unified in the VLN field.

B More Qualitative Examples

We provide additional successful and failure cases to qualitatively analyze the capabilities and limitations of our proposed MapGPT.

Figure 5 demonstrates a successful case on the R2R dataset. In step 6, after thoroughly exploring places 3 and 4 connected to place 1, the agent decides to backtrack to place 1 and subsequently explore currently inaccessible places 6 and 7. Ultimately, the agent successfully terminates at place 7 in step 10.

As shown in Figure 6, we further summarize two typical types of failure cases, which are also common challenges for other zero-shot VLN agents. (a) The agent may fail to follow the details in the instructions accurately. For instance, instead of walking straight into a bedroom in the eleven o'clock direction as instructed, it turns left in step 1 and enters another incorrect bedroom, and stops there. (b) The scenes are highly challenging, and the instructions may not provide sufficient clues. Thus, the agent may fail to explore the correct direction in time.

R2R Task Description

[Task background]

You are an embodied robot that navigates in the real world. You need to explore between some places marked with IDs and ultimately find the destination to stop. At each step, a series of images corresponding to the places you have explored and have observed will be provided to you.

[Input Definitions]

'Instruction' is a global, step-by-step detailed guidance, but you might have already executed some of the commands. You need to carefully discern the commands that have not been executed yet.
'History' represents the places you have explored in previous steps along with their corresponding images. It may include the correct landmarks mentioned in the 'Instruction' as well as some past erroneous explorations.
'Trajectory' represents the ID info of the places you have explored. You start navigating from Place 0.
'Map' refers to the connectivity between the places you have explored and other places you have observed.
'Supplementary Info' records some places and their corresponding images you have ever seen but have not yet visited. These places are only considered when there is a navigation error, and you decide to backtrack for further exploration.
'Previous Planning' records previous long-term multi-step planning info that you can refer to now.
'Action options' are some actions that you can take at this step.

[Output Requirements]

For each provided image of the places, you should combine the 'Instruction' and carefully examine the relevant information, such as scene descriptions, landmarks, and objects. You need to align 'Instruction' with 'History' (including corresponding images) to estimate your instruction execution progress and refer to 'Map' for path planning. Check the Place IDs in the 'History' and 'Trajectory', avoiding repeated exploration that leads to getting stuck in a loop, unless it is necessary to backtrack to a specific place. If you can already see the destination, estimate the distance between you and it. If the distance is far, continue moving and try to stop within 1 meter of the destination. Your answer must include four parts: 'Thought', 'Distance', 'New Planning', and 'Action'. You need to combine 'Instruction', 'Trajectory', 'Map', 'Supplementary Info', your past 'History', 'Previous Planning', 'Action options', and the provided images to think about what to do next and why, and complete your thinking into 'Thought'. Based on your 'Map', 'Previous Planning' and current 'Thought', you also need to update your new multi-step path planning to 'New Planning'. At the end of your output, you must provide a single capital letter in the 'Action options' that corresponds to the action you have decided to take, and place only the letter into 'Action', such as "Action: A".

REVERIE Task Description

[Task background]

You are an embodied robot that navigates in the real world. You need to explore between some places marked with IDs and ultimately find the target object to stop. At each step, a series of images corresponding to the places you have explored and have observed will be provided to you.

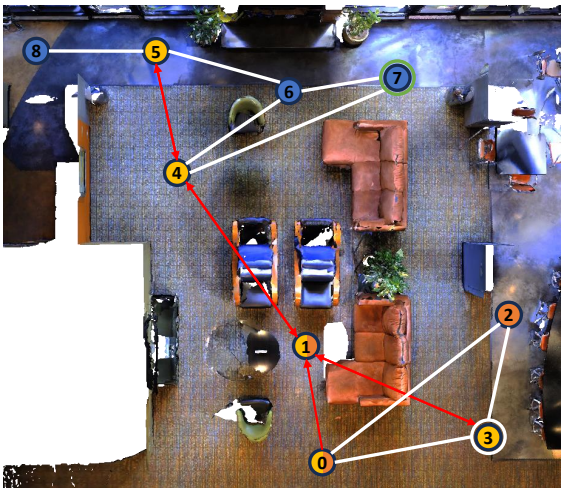
[Input Definitions]

'Instruction' is a global guidance that you should follow. You only need to find the indicated or hidden target object within it, stop, and ignore any actions mentioned in the 'Instruction' regarding the target object. You don't need to excessively adhere to the color details about landmarks and the target object in the 'Instruction', as the descriptions about colors might be incorrect.
'History' represents the places you have explored in previous steps along with their corresponding images. It may include the correct landmarks mentioned in the 'Instruction' as well as some past erroneous explorations.
'Trajectory' represents the ID info of the places you have explored. You start navigating from Place 0.
'Map' refers to the connectivity between the places you have explored and other places you have observed.
'Supplementary Info' records some places and their corresponding images you have ever seen but have not yet visited. These places are only considered when there is a navigation error, and you decide to backtrack for further exploration.
'Previous Planning' records previous long-term multi-step planning info that you can refer to now.
'Surroundings' represent some objects you can touch in your surrounding environment at this step.
'Action options' are some actions that you can take at this step.

[Output Requirements]

For each provided image of the places, you should combine the 'Instruction' and carefully examine the relevant information, such as scene descriptions, landmarks, and objects. Then, check the image of your current place and see if the touchable objects in 'Surroundings' include any possible target object that you may need to stop at. If not, you should utilize commonsense knowledge, such as potential room layouts in indoor environments, to explore possible directions and places. You need to align 'Instruction' with 'History' (including corresponding images) to estimate your instruction execution progress and refer to 'Map' for path planning. Check the Place IDs in the 'History' and 'Trajectory', avoiding repeated exploration that leads to getting stuck in a loop, unless it is necessary to backtrack to a specific place. If you can already see the target object, estimate the distance between you and it. If the distance is far, continue moving util the target is touchable. Your answer must include four parts: 'Thought', 'Distance', 'New Planning', and 'Action'. You need to combine 'Instruction', 'Trajectory', 'Map', 'Supplementary Info', your past 'History', 'Previous Planning', 'Surroundings' and 'Action options' information to think about what to do next and why, and complete your thinking into 'Thought'. Based on your 'Map', 'Previous Planning' and current 'Thought', you also need to update your new multi-step path planning to 'New Planning'.
At the end of your output, you must provide a single capital letter in the 'Action options' that corresponds to the action you have decided to take, and place only the letter into 'Action', such as "Action: A".

Figure 4: Task description prompts for the R2R and REVERIE datasets. We make some simple yet necessary modifications to transfer MapGPT from the R2R task to REVERIE. This work focuses on unified navigation, while instructions in REVERIE often require some interactive actions on objects. Therefore, we require the agent to ignore these actions.



Step6 **Task Description:** *[Task background] [Input Definitions] [Output Requirements]*
Instruction: Go forward toward the windows. Go toward the the far couch, Stop next to the couch, in front of the windows.
History: step 0: turn around to Place 1: <a lobby with blue couches and a fireplace>, which also includes <teapoy, bin, 2 window frames, shape sofa, 2 armchairs, ceiling, plant, floor, ledge>, ... step 5: turn left to Place 3: <a blue and white kitchen with a bar>, which also includes <kitchen cabinet, 3 walls, kitchen island, 2 ceilings, 2 stools, cabinet, shape sofa, pipe>
Action options (step 6): ['A. stop', 'B. turn around to Place 0: <the lobby of the savannah hotel>, which also includes <2 floors, 3 walls, 3 armchairs, picture, tv stand, teapoy, ceiling>', 'C. turn around to Place 1: <the lobby at the savoy hotel>, which also includes <2 ceilings, ledge, 2 walls, tv stand, 3 armchairs, picture, fireplace, tv>', 'D. turn around to Place 2: <a modern office with a circular bar>, which also includes <floor, 2 plants, 2 walls, 3 ceilings, ledge, 2 shape sofas, window frame, cabinet, stool>']

Trajectory: Place 0 1 4 5 4 1 3

Map:
Place 0 is connected with Places 1, 2, 3
Place 1 is connected with Places 0, 4, 3
Place 4 is connected with Places 1, 5, 6, 7
Place 5 is connected with Places 4, 8, 6
Place 3 is connected with Places 0, 1, 2

Supplementary Info:
Place 6: <the living room at the san antonio hotel>
Place 7: <the living room at the san antonio hotel>
Place 8: <a staircase leading to a building>

Previous Planning: From Place 1, I will turn left to Place 3, which is a blue and white kitchen with a bar. If I still can't find the couch in front of the windows, I will then move to Place 0, which is connected to Places 1, 2, and 3. I will continue this process until I find the couch in front of the windows.

Thought: According to the instruction, I need to find a couch in front of windows. From the history, I have visited a few places with couches and windows, but none of them match the instruction. The previous planning was to go to Place 3, which is a kitchen, and then to Place 0. However, the kitchen does not match the instruction and Place 0 is the starting point, which I have already visited. Therefore, I need to update my planning. From the map, I can see that Place 4 is connected to Place 6 and 7, both of which are living rooms at the San Antonio hotel. It's possible that one of these places matches the instruction. I should consider visiting these places.
New Planning: From Place 3, I will go back to Place 1, then to Place 4, and finally to Place 6 or 7. I will check these places to see if they match the instruction.
Action: C. turn around to Place 1: <the lobby at the savoy hotel>, ...

Step10 **Thought:** According to the instruction, I need to find a place with a couch and windows. Currently, I am at Place 7, which is described as "a living room with couches and a large window". This place seems to match the instruction ..... Therefore, I believe that I am currently at the correct place.
New Planning: Since I believe that I am currently at the correct place, I do not need to plan any further steps.
Action: A. stop

Figure 5: A successful example on the R2R dataset. We demonstrate some crucial steps that leverage map-guided global exploration and planning capabilities, ultimately resulting in successful navigation.

(a)

Task Description: *[Task background] [Input Definitions] [Output Requirements]*
Instruction: Go all the way down the stairs and then go down the hallway in the 11 o'clock position from the base of the stairs. Keep walking straight until you enter a bedroom and stop after entering.
History: step 0: go down to Place 2: <a hallway with blue stairs and a painting>, ..., step 1: turn left to Place 3: <a bedroom with a blue door>, ... step 2: turn around to Place 4: <a blue guitar and a blue door>, ... step 3: turn around to Place 3: <a bedroom with a view of the mountains>, which also includes <2 walls, bed, door, side table, 3 doorframes, 2 floors, tv>
Action options (step 4): ['A. stop', 'B. turn around to Place 4: <a blue guitar and a blue door> ...]

Trajectory: Place 0 2 3 4 3
Map:
...
Place 4 is connected with Places 2, 6, 3, 5

Supplementary Info:
Place 1: <a staircase with a painting on it>
Place 5: <a hallway with a painting of albert einstein>

Previous Planning: From Place 4, I will turn around to Place 3. From Place 3, I will go to Place 7.

Thought: The instruction indicates that I should go down the stairs, then go down a hallway, and finally enter a bedroom. From the history, I can see that I have already gone down the stairs to Place 2, and then entered a bedroom at Place 3. However, I then turned around to Place 4 and back to Place 3 again, which seems to be a mistake as it doesn't align with the instruction. The instruction doesn't mention turning around. Now, I am at Place 3, which is a bedroom. According to the instruction, I should stop after entering the bedroom. Therefore, I believe I have reached the destination.
New Planning: Since I believe I have reached the destination, there is no need for further planning. Action: A. stop

(b)

Task Description: *[Task background] [Input Definitions] [Output Requirements]*
Instruction: Wash the soap dispenser in the spa room at the end of the hall
History: step 0: go down to Place 1: <a spa room with a large tub> ... step 2: turn right to Place 13: <a room with a hot tub and a circular light>, which also includes <3 walls, ceiling, lounge chair, window>
Surroundings: vent. Action options (step 4): ['A. stop', 'B. turn around to Place 12: <a living room with a couch and a television>, ... ', 'C. turn around to Place 2: <a bathroom at a hotel>, which also includes <door, cabinet, towel, stair, 3 floors, wall otherroom, wall>',...]

Trajectory: Place 0 1 7 13
Map:
Place 0 is connected with Places 1, 2, 3, 4
Place 1 is connected with Places 5, 6, 7, 8, 2, 9, 10, 0
Place 7 is connected with Places 6, 11, 12, 13, 2, 8, 1, 5
Place 13 is connected with Places 12, 2, 10, 8, 7

Supplementary Info:
Place 3: <a hallway with tiled walls and a door>
Place 4: <a hallway with a white tiled wall>
Place 5: <a room with a massage table and chairs>
Place 6: <a room with a massage table and chairs>
Place 9: <a bathroom with a bathtub and a shower>
Place 11: <a small living room with a bar and a television>

Previous Planning: My previous plan is to first explore Place 13. If I don't find the soap dispenser there, I will then go to Place 12. If I still don't find the soap dispenser, I will continue exploring the other places.

Thought: I am currently in Place 13, which is described as a room with a hot tub and a circular light. However, I do not see a soap dispenser in my surroundings, which is the target object according to the instruction. Looking at the map, I see that Place 13 is connected to Places 12, 2, 10, 8, and 7. Among these, Place 2 is described as a bathroom at a hotel, which might contain a soap dispenser. However, the instruction specifically mentions a spa room, and none of the connected places seem to fit that description. I have already visited Places 1 and 7, both described as spa rooms, but did not find the soap dispenser there. Therefore, I should continue exploring the other places.
New Planning: My new plan is to first explore Place 12. If I don't find the soap dispenser there, I will then go to Place 2. If I still don't find the soap dispenser, I will continue exploring the other places. Action: B.

Figure 6: Two typical reasons for failure. (a) Stopping erroneously in similar locations. (b) Failure to timely backtrack and explore the correct direction if the scene is complex and the instruction does not provide much guidance (the agent has explored place 1 and plans to explore place 2, while the correct direction is 0→3).