Cognitive Planning for Object Goal Navigation using Generative AI Models

Arjun P S
IIT ISM Dhanbad, India

Andrew Melnik
Bielefeld University, Germany

Gora Chand Nandi
IIIT Allahabad, India

Abstract

Recent advancements in Generative AI, particularly in Large Language Models (LLMs) and Large Vision-Language Models (LVLMs), offer new possibilities for integrating cognitive planning into robotic systems. In this work, we present a novel framework for solving the object goal navigation problem that generates efficient exploration strategies. Our approach enables a robot to navigate unfamiliar environments by leveraging LLMs and LVLMs to understand the semantic structure of the scene. To address the challenge of representing complex environments without overwhelming the system, we propose a 3D modular scene representation, enriched with semantic descriptions. This representation is dynamically pruned using an LLM-based mechanism, which filters irrelevant information and focuses on task-specific data. By combining these elements, our system generates high-level sub-goals that guide the robot's exploration toward the target object. We validate our approach in simulated environments, demonstrating its ability to enhance object search efficiency while maintaining scalability in complex settings.
Visualization : `https://youtu.be/pvr1uaObL9M`

1 INTRODUCTION

Navigation in an unfamiliar environment to search for an object described in natural language is one of the most challenging problem in robotics. Even though these kind of task comes as second nature for us humans, the underlying process is really complex as it involves cognitive processing, using long term memory and experiences, and integrating the current sensory information with these processes. Achieving this with robots involve designing a language conditioned high-level planner that understands the semantic priors of this world and is able to devise intelligent sequential plans to efficiently explore the scene in search of the object. This is the core of an object goal navigation task. The agent, similar to a human being, must also have a cognitive high level-planner that can understand the semantic cues of our world, a low-level planner that can leverage on long term memory and experiences to act on the environment and an episodic memory to save critical information related to the current task.

Large Language Models (LLMs) and Large Vision Language Models (LVLMs) have shown the capability to comprehend the semantic priors of the world and reason about them from this understanding. This makes them an ideal candidate for a high level planner, that can take an action by reasoning about the scene. To plan efficiently in an environment by taking into account the underlying details of the scene, the agent should be provided with an efficient representation of the scene which is rich in semantic information but yet not overwhelmed with a mass of indistinguishable stimuli. Human beings, when exploring an unknown environment to find an object without a comprehensive map of the environment, follows a goal oriented approach in storing data. This process involves a combination of Perception which process the incoming information, Attention which directs our focus to specific cues in the scene and memory which selectively encodes information that is perceived as important for achieving a specific goal. This aids in efficient use of cognitive resources and effective retrieval of this information when needed. This goal oriented approach enhances the chances of us

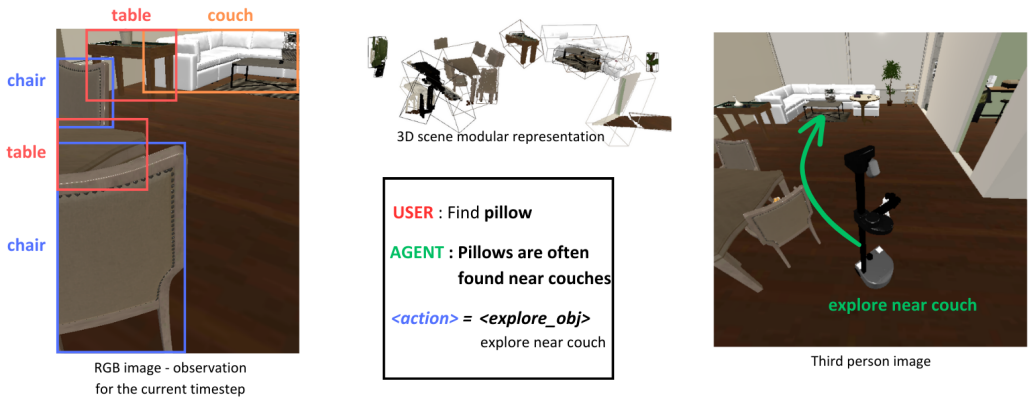


Figure 1: Overview of our framework with Homerobot Strech in Habitat simulation environment. The Robot in this episode is tasked to find a pillow. The agent, after considering all the objects in the scene (3D scene modular representation), decides to explore near the couch to find the pillow.

finding the target object. We mimic this goal oriented approach with robots by using a combination of an Large language model for attending to relevant aspects, an large vision language model and an open-vocabulary image segmentation module for perceiving and processing the incoming information, and a 3D scene modular representation and a short term memory module for saving the processed information to memory.

The agent when exploring the scene, procedurally builds a modular 3D scene representation, by using the processed information from the perception module. The 3D scene representation has nodes corresponding to objects of relevance in the scene, the relevancy of which is decided by a Large Language Model. The 3D scene representation generated is generally sparse, but becomes denser in the vicinity of the detected target object. This sparse to dense structure in encoding information mirrors the human cognition in strategically encoding information in a goal oriented task. The perception module involves an open vocabulary segmentation module which is responsible for identifying and segmenting all the objects in the scene and a Large Vision Language model used for describing the identified objects in natural language. This natural language description have details like the appearance of the object, the objects near it and the possible room this object is in. This information helps the high level planner to plan more effectively. The short term memory module temporarily holds and manipulate information needed for generating inferences about the target object.

The main contributions of our work involve:

- A framework inspired from human cognition to attend, perceive and store information, to find an object in an unknown environment, that leverages the reasoning capability of Large Language Models (LLMs) and Large Vision Language models (LVLMs), to deduce efficient plans by procedurally building a 3D scene modular representation.
- In-context learning based approach with LLMs to find the pertinent objects in the scene, to build a goal oriented, streamlined and scalable 3D scene representation on the fly, which aids in effective reasoning and efficient exploration in an unfamiliar environment.
- An inference strategy that uses an LVLM to identify whether a detected segment is the target object or not, on all the frames in the short term memory, in which this detected segment was visible.

2 Related works

2.1 LLM as a planner

Large language models (LLMs) with billions of parameters, trained on massive scale datasets have shown impressive generative capabilities with a generalized semantic understanding of the world.

Many works like (14; 8; 30; 17; 13; 1) focused on grounding LLMs for high-level task planning with natural language task representation. (13) worked on grounding high-level tasks expressed in natural language and decomposing them into low level plans and admissible actions. (1) focused on grounding an LLM that provides high-level procedures for task completion, with value functions associated with these tasks. Many approaches like (29) and (12) used LLMs to follow natural language navigation instructions.

2.2 Open Vocabulary Image Segmentation

Open vocabulary image segmentation refers to the recognition and delineation of an open category of objects in a frame. The flexibility of such a system to identify and segment objects forms a critical part in building an efficient representation of the scene. With the advent of powerful object detection and segmentation models like (15; 32; 11; 40; 39; 26; 20), the agent can detect and segment the objects in the scene. Many open vocabulary segmentation modules like (40; 26; 15) require the labels of the objects to perform instance segmentation. Image tagging models like (37) can recognize the objects in a scene and return their labels, which can be used as text prompt to these models.

2.3 Representating the environment for task planning

Prior works like (10; 6; 12; 5; 36; 7) focused on building efficient representations of the environment to facilitate downstream task planning. The representations generated can be used for understanding the semantics of the scene, thus allowing the agent to query the environment for an object using natural language (10; 6) and generate task level plans (25; 6) or low-level actions (7). Even though many scene representations like (5; 7) have created a scalable and efficient representation that can be used for navigation, it lacks contextual information about the scene, which is invaluable when exploring a scene. Scene representations like (10; 6) have a really dense representation, which is often not required when exploring an unfamiliar environment in search of a target object G_o. Our present research is inspired by the human way of cognition with an affordance based memory that attends and stores only those information which are relevant to the task at hand. We leverage on the ability of In-context learning in LLMs to prune out irrelevant information and then using an LVLM to reason about the relevant pruned list of objects, thus creating a sparse as well as information rich representation of the environment. The scene representation near the target object G_o is made denser so as to facilitate further downstream task planning like manipulation of the object.

2.4 Object goal navigation

Object goal navigation task involves finding a target object G_o in an unfamiliar environment. Efficient exploration through the environment requires deep understanding of semantic priors of the world. Prior works like (5; 2; 22; 4; 16; 34) tried to learn these semantic cues from egocentric RGB and depth images (2; 22), semantic map (5) and even from YouTube videos (4). These learned semantic cues might not generalize well to new unseen environments. (2) focused on using a modular transfer learning model to generalize the learned policy for a particular task to multiple tasks and (33) used meta learning to generalize to unfamiliar environments. LLMs on the other hand exhibit extraordinary contextual awareness and ability to understand the semantic cues of our world. We leverage on this capability to efficiently explore unseen environments for the target object G_o.

Recent works like (24; 9; 28) have leveraged the capability of LLMs to understand the semantic priors for object goal navigation tasks. (24) leveraged the planning capabilities of LLMs to devise a sequential plan that includes point goals and target state information for executing multiple object goal navigation tasks in parallel. These point goals generated by an LLM is fulfilled through a low level(execution level) controller. (28) focused on using the semantic predictions from LLMs as a heuristic for a frontier based exploration strategy to find the target object G_o. (9) focused on solving language-driven zero-shot object goal navigation problem (21) by using an LLM to navigate to the target object G_o, given a natural language description of it.

Most of these approaches either focuses on building a semantically rich map by exploring the environment first, and then utilizes a policy to navigate on top of this representation or have a procedurally built semantic map which lacks rich semantic information but is built on the fly and a policy trained on top of this, which learns the semantic priors of the world. Such policies, especially those trained on simulators lack generalization capabilities. We utilize the generalization capabilities

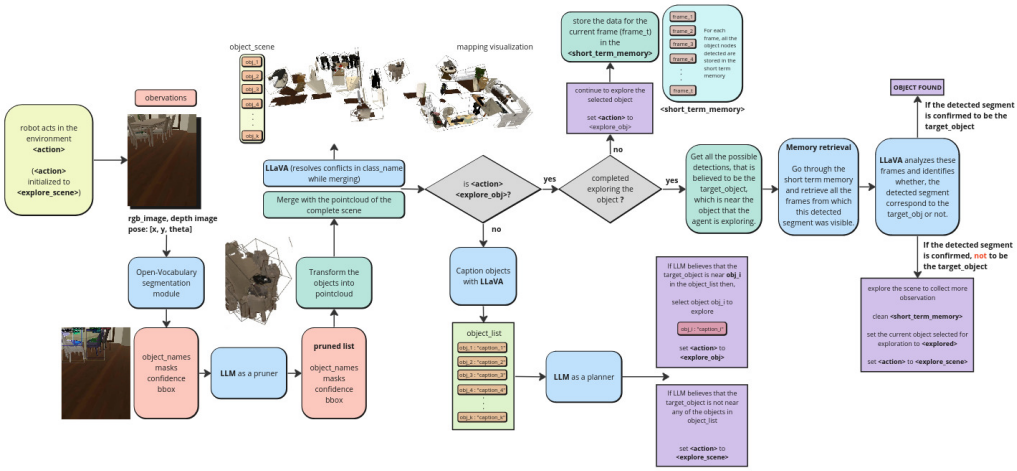


Figure 2: Architecture of the proposed pipeline.The agent explores the environment and collects observations (RGBD image and Pose). An open-vocabulary segmentation module is used to identify the objects in the current frame. Pruner takes in these detected segments and prunes out unwanted segments. The pruned segments are either initialized as a new node in the 3D scene representation or merged with an existing one, based on a similarity criteria. All new nodes are captioned with LLaVA, to provide semantic information to the LLM based planner. The agent then chooses a node to explore closer, to find the target object G_o. While doing so, the agent stores frame wise information in the short term memory module. If the agent decides that no objects in the 3D scene representation has a good chance of finding the target object G_o closer to it, it continues to explore the scene and build the 3D scene representation.

of foundational models to understand and reason about the context through natural language, to procedurally build a 3D modular representation which encodes rich semantic data, and generate plans for an object goal navigation task.

3 METHOD

Our work aims to design a planner for object goal navigation task in an unfamiliar environment by building a semantic 3D scene representation of the environment and utilizing an LLM to generate plans on top of this representation. The goal of the agent is to find a target object G_o in the environment and to create a rich representation of the scene around the object for downstream planning. We leverage the reasoning capability of Large Language Models(LLMs) to understand the semantic priors of the environment and thus to guide the robot to those regions which have a higher chance of finding the target object G_o. The agent constructs a semantic 3D scene modular representation which is a sparse version of (10), pruned with LLMs using the in-context learning paradigm (3; 38). The immediate relevant environment of the target object G_o will have a dense and rich representation while the rest will have a sparse representation. The overview of our approach is given in figure 2.

The agent receives RGB image, depth image, base and camera pose as inputs in every timestep. An open vocabulary semantic segmentation module identifies objects in the RGB image and generates mask for the same. LLM acting as a pruner identifies the most important objects in the scene in regards to their utility in understanding the semantic priors of the environment and prunes the remaining detections. These detected segments are then converted into a 3D scene representation, where each object is signified by a node. An object node stores information regarding the position of the object, pointcloud, frame in which this object was detected, mask and label of this object in that frame, and a semantic description of the object in the scene. LLM, grounded for an object goal navigation task, uses the 3D scene modular representation to decide whether to continue exploring the environment or to move closer to an already detected object in the scene that has a high probability of finding the target object G_o. The agent triggers its short term memory module and stores information for every frame if it decides to explore closer to an object. Upon reaching the object, the agent utilize the detections from the short term memory, to construct hypotheses regarding the target object.

3.1 Open vocabulary image segmentation

The open vocabulary segmentation module segments the RGB frame I_t at the current timestep t, given the natural language description of the object. Given an RGB image I_t of the scene, we use a combination of Recognize anything model(RAM) (37), Grounding Dino and FastSAM (26) for open vocabulary semantic segmentation. RAM is a strong foundational model used for image tagging, it is tasked with finding the object tags c_t^i in the scene. The set of object tags $C_t = (c_t^1, c_t^2, ..., c_t^m)$ is then passed as a text prompt T_t $(= C_t)$ to Grounding-Dino (20), which detects the presence of classes C_t in the scene. Grounding-Dino returns the class labels $(l_t^1, l_t^2, ..., l_t^m)$ of objects in the scene, confidence of the predictions $(p_t^1, p_t^2, ..., p_t^m)$ and their bounding boxes $(b_t^1, b_t^2, ..., b_t^m)$. The detected classes in the scene $(l_t^1, l_t^2, ..., l_t^m)$ are then passed as box prompt $(b_t^1, b_t^2, ..., b_t^m)$ through FastSAM (39) to get their corresponding masks $(m_t^1, m_t^2, ..., m_t^m)$.

3.2 LLM as a pruner

We leverage the In-context learning abilities of an LLM to prune the detected class labels $(l_t^1, l_t^2, ..., l_t^m)$ generated by the open vocabulary segmentation module for the timestep t. In-context learning is a paradigm that empowers pre-trained LLMs to ground to a particular task without fine-tuning the model, by providing the LLM with task level demonstrations along with the prompt. The language model here is provided with task-specific examples containing a pair of input P_i and output P_o object sets. The input set P_i contains class labels $(l^1, l^2, ..., l^m)$ and the output set $P_o(\subseteq P_i)$, contains those objects that have a higher utility when it comes to understanding the semantic priors of the scene. The text prompt provided to the LLM to achieve this behaviour is given in fig **??**.

3.3 3D scene modular representation

The pruned semantic masks $(m_t^1, m_t^2, ..., m_t^m)$ and labels $(l_t^1, l_t^2, ..., l_t^m)$ from the open vocabulary segmentation module, along with the depth image I_t^D and camera pose P_t in the current time step t are used to generate 3D object nodes $(N_{c1}, N_{c2}, ..., N_{ck})$ for each object O_i in that frame (I_t, I_t^D) using a modified concept graphs pipeline (10). In concept graphs, the 3D object nodes N_{ci} in the scene representation are fused from multiple views considering their spatial overlap, using an object association strategy that compares the semantic and visual similarity of these objects. Merging objects from different frames often creates conflicts in class labels, as the object segment can be associated with different class labels in different frames. We propose to use a Large Vision Language Model (LVLM) to resolve this conflict. The LVLM is tasked to find the class label that best suits the object detected, by giving it a cropped image of the detected object and the conflicting class labels.

Considering an object goal navigation task, where the agent is tasked to find a target object G_o in the scene, further pruning of the concept graphs representation considering the utility of objects, in the regions not in the immediate relevant environment of the target object G_o, so as to generate a really sparse representation will improve scalability, which enables the system to handle larger and more complex environments without a proportional increases in computational or memory requirements. But having a sparse representation around the target object, constrains the agent in generating downstream plans for manipulating this object. To resolve this, the agent is conditioned to generate a really dense and semantically rich representation around the target object G_o which will enable the agent to execute downstream planning and manipulation tasks.

The object nodes generated are captioned using an LVLM in order to extract more information regarding the semantics of the object. Node captioning system in concept graphs is done while generating the scene graph, after construction of the entire 3D scene representation. This restricts the robots in fully utilizing the information while generating plans to explore the environment. But generating captions on the fly utilizes a lot of compute power, thus slowing down the robot by a considerable factor. To resolve this, the agent is tasked to caption nodes only when the action at the current timestep is $< explore_scene >$, i.e. the agent captions the node in every timestep, until it selects an object to explore further. In the phase where the agent explores towards a selected object ($action =< explore_obj >$), it saves all the relevant information regarding objects. The agent processes this information and generates captions for the object nodes at the end of this phase which is indicated when the robot reaches the selected object node.

You are a robotic agent, exploring a common household. Output in the following JSON string format:
{"output" : "output list for the current input, "reason" : "reason behind the choice"}
Be concise. Output only the JSON string without any explanation."

You are tasked to select those objects from the input list, that is relevant when it comes to storing information regarding the scene. A few examples are given below

Input: [black, buffet, bureau, cabinet, table, drawer, dresser, flower, glass_vase, plant, vase]
Output: [bureau, cabinet, table, drawer, dresser]
Input: [armchair, chair, cloth, curtain, floor, ladder, pillow, linen, room, stool, towel, towel_bar, white, window, wood_floor]
Output: [armchair, chair, ladder, stool]
Input: [bath, bathroom, bathroom_accessory, bed, bedroom, cabinet, carpet, closet, dresser, floor, image, pillow, room, sink, stool, tub, white]
Output: [bed, cabinet, dresser, sink, stool, tub]
Input: [baseball_glove, bat, cloth, clothesline, curtain, hang, hanger, image, shower_curtain, white]
Output: [ ]
Input: [carpet, floor, hardwood_floor, mat, plank, curtain, room, stool, wood_floor]
Output: [stool]
find the output for the given set of inputs
Input: [<*new_inputs*>]

(a) Prompt to LLM for pruning the detections

You are a robotic agent, exploring a common household. Output in the following JSON string format:
{"index" : "index of object the robot needs to explore, set this to 0 if the needs to explore elsewhere", "reason" : "why does the robot want to explore here"}.
Be concise. Output only the JSON string without any explanation.

Given below are a list of objects in the scene and their description. Choose one object from below to explore, such that there is a good chance of finding {<*target_object*>} near it.
If there is very low chance of finding the {<*target_object*>} with the given set of object list, then set the index as 0.

1. *object_1_name* : <*caption*>
2. *object_2_name* : <*caption*>
3. *object_3_name* : <*caption*>
⋮
m. *object_m_name* : <*caption*>

(b) Prompt to LLM for planning.

3.4 LLM as a planner

LLM, grounded in the current task using task level description of the scene is used to generate a high level plan for the agent. Given the current objects in the environment (objects in the constructed 3D scene representation ($N_{c1}, N_{c2}, ..., N_{ck}$)) and the generated descriptions for these object nodes, the LLM is tasked to identify whether to continue exploring the environment randomly and build up the 3D scene representation, or to explore the regions closer to an object node N_{cj}. If the LLM decides to explore the scene more, then $action$ is set to $< explore_scene >$ and if it believes that there is a high chance of finding the target object G_o near any of the detected objects, then $action$ is set to $< explore_obj >$. This ability of an LLM to extract semantic relationships between the target object G_o and objects in the scene ($N_{c1}, N_{c2}, ..., N_{ck}$) helps the agent to explore efficiently in the environment. The text prompt provided to the LLM to achieve this behaviour is given in fig **??**.

3.5 Short term memory(STM) and reasoning

Short term memory(STM) is a temporary storage module for holding and manipulating information amidst task execution. This working memory module is triggered when the $action$ is set to $< explore_obj >$. The agent stores processed information (inference by the open vocabulary segmentation module and segment in that frame which contributed to object nodes in the 3D scene representation) for all the frames and holds it, until it has completely explored near an object. Having a short term memory (temporary) oven an long term or permanent memory reduces memory constraints and increases the scalability of the architecture.

Upon reaching the target object, the agent pans around its axis to collect information regarding the scene, which is processed by the segmentation module primed with the target object. This generates all the possible segments that can be regarded as the target object. The agent then retrieves all the frames in the short term memory, from which these segments are visible, thus generating different views for the same detected segment. This is done by finding the spatial overlap of pointclouds between the segment of interest and the objects from frames stored in the short term memory.

The agent then tasks an LVLM (LLaVA (19; 18)) to verify whether the detected segment in all the views is the target object or not. The determination of whether the detected segment represents the target object G_o relies on the percentage of frames (different views) in which the LVLM (LLaVA) confirms the presence of the target object, serving as the decisive criterion for the final decision. If none of the detected segments are regarded as the target object then the current object selected for exploring is set to explored, the short term memory is cleared and the agent's $action$ is set to $< explore_scene >$.

3.6 Execution level planner

We make use of a goal oriented and episodic semantic mapping module (5), to build an episodic 2D scene obstacle map, frontier map and a goal map. In the phase where $action$ is $< explore_scene >$, the goal map is same as that of the frontier map and when the $action$ is $< explore_obj >$, the goal map is constructed in such a way that the robot can navigate to a region close to the object selected by the LLM. The goal map and the obstacle map is then parsed to a fast marching method planner, which plans execution level actionable sub-goals to reach the goal.

4 EXPERIMENT AND RESULTS

We evaluate our stack to explore for a target object by building a 3D scene representation of the environment on the HomeRobot: Open Vocabulary Mobile Manipulation (35) simulation benchmark. This simulation benchmark have multi-room realistic environments, with a diverse and complex set of receptacles (eg: chair, table, couch, bed, toilet etc.) and articles (eg: pen, book, cellphone, tray, apple etc.) that can be manipulated.The environments, designed in Habitat (23; 31; 27) provide a cluttered and navigable scene having multiple rooms with manipulable objects placed on top of receptacles, to test various benchmarks related to navigation and manipulation of objects in unfamiliar environments. We chose this dataset over conventional object-nav datasets and benchmarks, because of the availability of a wide range of small objects, that are manipulable. The main focus of our work is to explore for these manipulable objects. However, the dataset provided by this benchmark does not consider the general semantics of the world while spawning objects on top of receptacles. For example, there is an episode in this dataset that spawns in an apple on top of a toilet. We went through the dataset and curated custom episodes that follow these semantics to test the stack.

We test our stack for exploration by spawning the robot in a random location in the scene and asking it to find the target object within 500 steps. We evaluate the agents performance in navigating to the target object by considering success rate (SR) and success rate weighted by path length (SPL) which indicates the efficiency of the planner. The focus of our approach is in creating a planning pipeline that mimics human cognition in the way we perceive, store, express and regulate information. To test the efficacy of the approach compared to human level cognition, we tasked a set of random human volunteers to find the target object, for the same set of episodes that were given to the LLM based planner and measured the success rate (SR) and the success rate weighted by path length (SPL). The results of the experiments involving human and LLM agent is given in table 1.

Table 1: Experimental results

AGENT	SR	SPL
Human	0.9375	0.7588813781
GPT-4	0.4375	0.2721472463
GPT-3.5 Turbo	0.0	0.0

Table 2: Ablation analysis

AGENT	SR	SPL
GPT-4 with STM and pruner	0.4375	0.2721472463
GPT-4 without STM	0.125	0.089943
GPT-4 withot pruner	0.182	0.1131
GPT-4 withot object captions	0.36363	0.2254

5 DISCUSSION

5.1 Comparison with human cognition

Experimental results from table 1, shows the comparison between human agent and LLM agent in completing an object goal navigation task. Although GPT-4 based agent's performance is far from human level performance, it has exhibited a similar thought process in exploring near certain objects in the scene, for finding the target object G_o. GPT-3.5 on the other hand failed to reason in most of the episodes, leading to an inefficient and meaningless exploration of the scene. Most of the failure scenarios with GPT-3.5 was when it failed to find the target object within the 500 step limit.

5.2 Ablation studies

To signify the importance of having a short term memory based inference module, object captioning module and a detection pruner, we performed ablation analysis with these, the results of which are presented in table 2.

GPT-4 with the short term memory(STM) based inference module, more effectively identified the target object G_o, when compared to the agent without this module. The strategy of finding the same detected segment from multiple views, already encountered while exploring and constructing a hypothesis from these views, improved the agents performance in identifying the target object. Comparing with an agent with STM, Most of the failure scenario for the agent without STM was due to false positive detection of target object G_o, leading to premature termination.

LLM based pruner is tasked to prune out smaller objects in an environment like pen, book, pillow etc, and keep object like chair, kitchen table, bed etc, which are important in understanding the semantics of the scene. By not doing so, the agent is flooded with a lot of redundant information, which can lead to inefficient plans. Table 2 shows the performance of the agent without the pruner. The agent without a pruning module engages in a lot of inefficient exploration, which can be seen with the reduced SPL, when compared to an agent with pruner.

Captioning nodes can capture a lot of semantic information, which can increase the efficiency of the planner. GPT-4 based agent, provided with captions performed better at reasoning about the target object G_o being near certain objects in the scene, leading to a more optimal path when compared to the case when the agent was only provided with the object names. Object captions can capture the semantic differences between objects of the same label. For example, the chance of finding an apple is more near a table in the kitchen than in the bedroom. Compared to the agent that had access to object captions, most of the failure scenarios transpired when the agent was not able to find the object within the 500 step limit. This can be attributed to the inefficient exploration due a lack of semantic understanding of the scene.

5.3 Limitations

Regardless of its capabilities, this framework possesses certain limitations, with object detection and segmentation being the most critical among them, creating false positive detection of objects. Another limitation was with the object captioning system, which often lead to exploration of unwanted objects. This can be attributed to the current limitations of LVLMs like LLaVA. Using a proprietary LLM like GPT-4 has a lot of challenges associated with it like the economic cost and slower control loop as its a cloud hosted model.

6 CONCLUSION

In this paper we introduced a framework to tackle the object goal navigation problem, by leveraging the generalization and semantic reasoning capabilities of Large Language Models and Large Vision Language Models to generate intelligent plans, by utilizing an efficient 3D scene representation of the environment. The agent embodied with this framework, was able to reason about its surrounding environment in a similar way to that of humans. Ablations studies shows the importance of having a short term memory (STM) based inference module, a pruner and semantic captions, which is in line with our architectural design inspired from human cognition. Future works may also explore the use of a more efficient representation, which the agent can use to reason about the environment.

References

[1] AHN, M., BROHAN, A., BROWN, N., CHEBOTAR, Y., CORTES, O., DAVID, B., FINN, C., FU, C., GOPALAKRISHNAN, K., HAUSMAN, K., HERZOG, A., HO, D., HSU, J., IBARZ, J., ICHTER, B., IRPAN, A., JANG, E., RUANO, R. J., JEFFREY, K., JESMONTH, S., JOSHI, N., JULIAN, R., KALASHNIKOV, D., KUANG, Y., LEE, K.-H., LEVINE, S., LU, Y., LUU, L., PARADA, C., PASTOR, P., QUIAMBAO, J., RAO, K., RETTINGHOUSE, J., REYES, D., SERMANET, P., SIEVERS, N., TAN, C., TOSHEV, A., VANHOUCKE, V., XIA, F., XIAO, T., XU, P., XU, S., YAN, M., AND ZENG, A. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691* (2022).

[2] AL-HALAH, Z., RAMAKRISHNAN, S. K., AND GRAUMAN, K. Zero experience required: Plug and play modular transfer learning for semantic visual navigation, 2022.

[3] BROWN, T., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J. D., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A., AGARWAL, S., HERBERT-VOSS, A., KRUEGER, G., HENIGHAN, T., CHILD, R., RAMESH, A., ZIEGLER, D., WU, J., WINTER, C., HESSE, C., CHEN, M., SIGLER, E., LITWIN, M., GRAY, S., CHESS, B., CLARK, J., BERNER, C., MCCANDLISH, S., RADFORD, A., SUTSKEVER, I., AND AMODEI, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., pp. 1877–1901.

[4] CHANG, M., GUPTA, A., AND GUPTA, S. Semantic visual navigation by watching youtube videos. In *NeurIPS* (2020).

[5] CHAPLOT, D. S., GANDHI, D., GUPTA, A., AND SALAKHUTDINOV, R. Object goal navigation using goal-oriented semantic exploration. In *In Neural Information Processing Systems* (2020).

[6] CHEN, B., XIA, F., ICHTER, B., RAO, K., GOPALAKRISHNAN, K., RYOO, M. S., STONE, A., AND KAPPLER, D. Open-vocabulary queryable scene representations for real world planning. In *arXiv preprint arXiv:2209.09874* (2022).

[7] CHEN, S., CHABAL, T., LAPTEV, I., AND SCHMID, C. Object goal navigation with recursive implicit maps, 2023.

[8] DING, Y., ZHANG, X., PAXTON, C., AND ZHANG, S. Task and motion planning with large language models for object rearrangement, 2023.

[9] DORBALA, V. S., AU2, J. F. M. J., AND MANOCHA, D. Can an embodied agent find your "cat-shaped mug"? llm-guided exploration for zero-shot object navigation, 2023.

[10] GU, Q., KUWAJERWALA, A., MORIN, S., JATAVALLABHULA, K., SEN, B., AGARWAL, A., RIVERA, C., PAUL, W., ELLIS, K., CHELLAPPA, R., GAN, C., DE MELO, C., TENENBAUM, J., TORRALBA, A., SHKURTI, F., AND PAULL, L. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv* (2023).

[11] HE, K., GKIOXARI, G., DOLLÁR, P., AND GIRSHICK, R. Mask r-cnn, 2018.

[12] HUANG, C., MEES, O., ZENG, A., AND BURGARD, W. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (London, UK, 2023).

[13] HUANG, W., ABBEEL, P., PATHAK, D., AND MORDATCH, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207* (2022).

[14] HUANG, W., XIA, F., XIAO, T., CHAN, H., LIANG, J., FLORENCE, P., ZENG, A., TOMPSON, J., MORDATCH, I., CHEBOTAR, Y., SERMANET, P., BROWN, N., JACKSON, T., LUU, L., LEVINE, S., HAUSMAN, K., AND ICHTER, B. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608* (2022).

[15] KIRILLOV, A., MINTUN, E., RAVI, N., MAO, H., ROLLAND, C., GUSTAFSON, L., XIAO, T., WHITEHEAD, S., BERG, A. C., LO, W.-Y., DOLLÁR, P., AND GIRSHICK, R. Segment anything, 2023.

[16] LIANG, Y., CHEN, B., AND SONG, S. Sscnav: Confidence-aware semantic scene completion for visual semantic navigation, 2021.

[17] LIN, K., AGIA, C., MIGIMATSU, T., PAVONE, M., AND BOHG, J. Text2motion: from natural language instructions to feasible plans. *Autonomous Robots* (Nov 2023).

[18] LIU, H., LI, C., LI, Y., AND LEE, Y. J. Improved baselines with visual instruction tuning, 2023.

[19] LIU, H., LI, C., WU, Q., AND LEE, Y. J. Visual instruction tuning. In *NeurIPS* (2023).

[20] LIU, S., ZENG, Z., REN, T., LI, F., ZHANG, H., YANG, J., LI, C., YANG, J., SU, H., ZHU, J., AND ZHANG, L. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023.

[21] MAJUMDAR, A., AGGARWAL, G., DEVNANI, B., HOFFMAN, J., AND BATRA, D. Zson: Zero-shot object-goal navigation using multimodal goal embeddings, 2023.

[22] MOUSAVIAN, A., TOSHEV, A., FISER, M., KOSECKA, J., WAHID, A., AND DAVIDSON, J. Visual representations for semantic target driven navigation, 2019.

[23] PUIG, X., UNDERSANDER, E., SZOT, A., COTE, M. D., PARTSEY, R., YANG, J., DESAI, R., CLEGG, A. W., HLAVAC, M., MIN, T., GERVET, T., VONDRUS, V., BERGES, V.-P., TURNER, J., MAKSYMETS, O., KIRA, Z., KALAKRISHNAN, M., MALIK, J., CHAPLOT, D. S., JAIN, U., BATRA, D., RAI, A., AND MOTTAGHI, R. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023.

[24] RAJVANSHI, A., SIKKA, K., LIN, X., LEE, B., CHIU, H.-P., AND VELASQUEZ, A. Saynav: Grounding large language models for dynamic planning to navigation in new environments, 2023.

[25] RANA, K., HAVILAND, J., GARG, S., ABOU-CHAKRA, J., REID, I., AND SUENDERHAUF, N. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. In *7th Annual Conference on Robot Learning* (2023).

[26] REN, T., LIU, S., ZENG, A., LIN, J., LI, K., CAO, H., CHEN, J., HUANG, X., CHEN, Y., YAN, F., ZENG, Z., ZHANG, H., LI, F., YANG, J., LI, H., JIANG, Q., AND ZHANG, L. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[27] SAVVA, M., KADIAN, A., MAKSYMETS, O., ZHAO, Y., WIJMANS, E., JAIN, B., STRAUB, J., LIU, J., KOLTUN, V., MALIK, J., PARIKH, D., AND BATRA, D. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019).

[28] SHAH, D., EQUI, M., OSINSKI, B., XIA, F., ICHTER, B., AND LEVINE, S. Navigation with large language models: Semantic guesswork as a heuristic for planning, 2023.

[29] SHAH, D., OSINSKI, B., ICHTER, B., AND LEVINE, S. LM-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *6th Annual Conference on Robot Learning* (2022).

[30] SONG, C. H., WU, J., WASHINGTON, C., SADLER, B. M., CHAO, W.-L., AND SU, Y. Llm-planner: Few-shot grounded planning for embodied agents with large language models, 2023.

[31] SZOT, A., CLEGG, A., UNDERSANDER, E., WIJMANS, E., ZHAO, Y., TURNER, J., MAESTRE, N., MUKADAM, M., CHAPLOT, D., MAKSYMETS, O., GOKASLAN, A., VONDRUS, V., DHARUR, S., MEIER, F., GALUBA, W., CHANG, A., KIRA, Z., KOLTUN, V., MALIK, J., SAVVA, M., AND BATRA, D. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

[32] WANG, C.-Y., BOCHKOVSKIY, A., AND LIAO, H.-Y. M. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022.

[33] WORTSMAN, M., EHSANI, K., RASTEGARI, M., FARHADI, A., AND MOTTAGHI, R. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning, 2019.

[34] WU, Y., WU, Y., TAMAR, A., RUSSELL, S., GKIOXARI, G., AND TIAN, Y. Learning and planning with a semantic model, 2018.

[35] YENAMANDRA, S., RAMACHANDRAN, A., YADAV, K., WANG, A., KHANNA, M., GERVET, T., YANG, T.-Y., JAIN, V., CLEGG, A. W., TURNER, J., KIRA, Z., SAVVA, M., CHANG, A., CHAPLOT, D. S., BATRA, D., MOTTAGHI, R., BISK, Y., AND PAXTON, C. Homerobot: Open vocab mobile manipulation, 2023.

[36] ZHANG, J., DAI, L., MENG, F., FAN, Q., CHEN, X., XU, K., AND WANG, H. 3d-aware object goal navigation via simultaneous exploration and identification, 2023.

[37] ZHANG, Y., HUANG, X., MA, J., LI, Z., LUO, Z., XIE, Y., QIN, Y., LUO, T., LI, Y., LIU, S., ET AL. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514* (2023).

[38] ZHAO, T. Z., WALLACE, E., FENG, S., KLEIN, D., AND SINGH, S. Calibrate before use: Improving few-shot performance of language models, 2021.

[39] ZHAO, X., DING, W., AN, Y., DU, Y., YU, T., LI, M., TANG, M., AND WANG, J. Fast segment anything, 2023.

[40] ZHOU, X., GIRDHAR, R., JOULIN, A., KRÄHENBÜHL, P., AND MISRA, I. Detecting twenty-thousand classes using image-level supervision. In *ECCV* (2022).

7 SUPPLEMENTARY MATERIAL

7.1 SHORT TERM MEMORY

When the agent chooses a node in the 3D scene representation to explore, It starts storing frame-wise data (RGBD image and processed segments from the open vocabulary object detector). If the agent detects the presence of the target object in a frame, then it finds all other views stored in the memory, from which this segment of the target object was visible. This is done by computing the spatial similarity (overlap between the pointcloud of detected object and all the segments in the memory). A LVLM is then tasked to reason the presence of the target object in all the retrieved frames. We use this reasoning to decide whether it is true or false detection. When the agent chooses a new node to explore, the current data stored in the short term memory is cleared.

Figure 4: Examples of frames retrieved from the short term memory module, in which the target object is detected. The top 8 frames corresponds to those in which the target object orange was visible and the bottom 8 frames have the target object soda can visible in them.

7.2 PRUNING

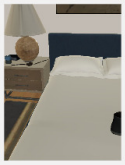
The pruner decides the abstraction level in which the planning occurs. For example, suppose we need to find an apple in an unseen environment. We can choose to represent the environment around us in different levels of abstraction, from room name to small objects like book. This representation will affect the quality of plans that we make to find apple. The task of the pruner is to provide an object level abstraction for the LLM based agent to plan a task. The pruner retains objects like computer table, chair, kitchen table, bed, cabinet and so on. To find an object like apple, the agent can choose to explore closer to one of these objects, in this case the kitchen table. Pruning is done with the help of incontext learning in LLMs. We provide a series of examples to LLM, that has a list of input objects and pruned objects and tasks LLM to do the same with a new list of input objects.

INPUT OBJECT LIST	PRUNED OBJECT LIST
['living room' 'hardwood floor' 'wall' 'table' 'picture frame' 'wall' 'armchair']	['armchair' 'table']
['living room' 'bookshelf' 'armchair' 'wall' 'hardwood floor' 'bookshelf' 'DVD' 'DVD' 'DVD' 'plant']	['bookshelf', 'armchair']
['furniture' 'living room' 'table' 'chair' 'furniture' 'hardwood' 'wall']	['table', 'chair']
'[room' 'hardwood floor']	[ ]
['round table' 'living room' 'couch' 'table' 'pillow' 'armchair' 'pillow' 'stool' 'sky' 'hassock']	['round table', 'armchair', 'stool', 'couch']
['kitchen table' 'fireplace' 'armchair' 'living room' 'armchair' 'armchair' 'living room']	['kitchen table', 'armchair']
['billiard table' 'chair' 'hardwood floor' 'wall']	['table', 'chair']
['bottle' 'bottle' 'room' 'dresser' 'bed']	['dresser', 'bed']

Figure 5: Input object list to LLM and the corresponding pruned list. The pruner here uses GPT-3.5 Turbo

7.3 LLM BASED PLANNER

The table below represents the plan executed by the agent (GPT-4 based agent) in an unseen environment to find an orange. Column 1 represents the object list along with the captions generated by LLaVA. Column two displays the frame when the agent identified a potential object (marked with a bounding box) to explore, for finding the target object. Column 3 has the frame at the end of exploring and Column 4 shows the agent's decision regarding the target object's presence.

OBJECT LIST WITH CAPTIONS	LLM PLAN & REASON	FINISHED EXPLORING	OBJECT FOUND?
1. lamp : A white lamp with a wicker base is placed on a table. **2. basket : The basket is made of wicker and has a brown appearance. It is placed on a table next to a lamp.** 0. elsewhere	 {"index": "2", "reason": "The basket might contain various items, including an orange"}		NO
1. lamp : A white lamp with a wicker base is placed on a table. **3. nightstand : A wooden nightstand with a lamp on it.** 4. pillow : The pillow is white and has a wooden frame. 5. headboard : The headboard is blue and has a lamp on it. 0. elsewhere	 {"index": "3", "reason": "nightstand may have an orange as it can be a place where people keep food items"}		NO

1. lamp : A white lamp with a wicker base is placed on a table. 4. Bed : The pillow is white and has a wooden frame. 5. headboard : The headboard is blue and has a lamp on it. 6. Bed : The mattress is white and appears to be a twin size. There is a remote control on the bed. 7. Cabinet : The dresser is made of wood and has a drawer. There is a box on top of it and a chair nearby. 8. Chair : The armchair is brown and has a wooden frame. It is placed next to a wooden dresser. 9. Bed : The bedcover is white and appears to be a simple design. There is a wooden floor under the bed and a window next to it. 10. Armchair : The armchair is brown and has a pillow on it. 11. Armchair : A chair is visible in the scene, with a wooden frame and a blue cushion. **0. elsewhere**	{"index" : "0", "reason" : "none of the objects suggest a high chance of finding an orange near them"}	-	-
1. lamp : A white lamp with a wicker base is placed on a table. 4. Bed : The pillow is white and has a wooden frame. 5. headboard : The headboard is blue and has a lamp on it. 6. Bed : The mattress is white and appears to be a twin size. There is a remote control on the bed. 7. Cabinet : The dresser is made of wood and has a drawer. There is a box on top of it and a chair nearby. 8. Chair : The armchair is brown and has a wooden frame. It is placed next to a wooden dresser. 9. Bed : The bedcover is white and appears to be a simple design. There is a wooden floor under the bed and a window next to it. 10. Armchair : The armchair is brown and has a pillow on it. 11. Armchair : A chair is visible in the scene, with a wooden frame and a blue cushion. 12. stool : A wooden stool is sitting on the floor next to a window. 13. dresser : The dresser is wooden and has a drawer. There is a lamp on top of it and a cup on the drawer. **0. elsewhere**	{"index" : "0", "reason" : "There is no specific information which implies the presence of an orange near any listed object"}	-	-
1. lamp : A white lamp with a wicker base is placed on a table. 4. Bed : The pillow is white and has a wooden frame. 5. headboard : The headboard is blue and has a lamp on it. 6. Bed : The mattress is white and appears to be a twin size. There is a remote control on the bed. 7. Cabinet : The dresser is made of wood and has a drawer. There is a box on top of it and a chair nearby. 8. Chair : The armchair is brown and has a wooden frame. It is placed next to a wooden dresser. 9. Bed : The bedcover is white and appears to be a simple design. There is a wooden floor under the bed and a window next to it. 10. Armchair : The armchair is brown and has a pillow on it. 12. stool : A wooden stool is sitting on the floor next to a window. 13. Dresser : The dresser is wooden and has a drawer. There is a lamp on top of it and a cup on the drawer. 14. lamp : The lamp in the scene is a white lamp with a wooden base. It is placed on a wooden table next to a window. 15. bed : The bed is a large, white bed with a blue blanket. There is a lamp on the table next to the bed. 16. carpet : The carpet in the scene is gray and brown. It is placed on a wooden floor. 17. pillow : The pillow is blue and white. 18. bed : The bed is blue and white. **0. elsewhere**	{"index" : "0", "reason" : "None of the objects indicate a high probability of finding an orange nearby"}	-	-
1. lamp : A white lamp with a wicker base is placed on a table. 4. Bed : The pillow is white and has a wooden frame. 5. headboard : The headboard is blue and has a lamp on it. 6. Bed : The mattress is white and appears to be a twin size. There is a remote control on the bed. 7. Cabinet : The dresser is made of wood and has a drawer. There is a box on top of it and a chair nearby. 8. Chair : The armchair is brown and has a wooden frame. It is placed next to a wooden dresser. 9. Bed : The bedcover is white and appears to be a simple design. There is a wooden floor under the bed and a window next to it. 10. Armchair : The armchair is brown and has a pillow on it. 12. stool : A wooden stool is sitting on the floor next to a window. 13. Dresser : The dresser is wooden and has a drawer. There is a lamp on top of it and a cup on the drawer. 14. lamp : The lamp in the scene is a white lamp with a wooden base. It is placed on a wooden table next to a window. 15. bed : The bed is a large, white bed with a blue blanket. There is a lamp on the table next to the bed. 16. carpet : The carpet in the scene is gray and brown. It is placed on a wooden floor. 17. pillow : The pillow is blue and white. 18. bed : The bed is blue and white. 0. elsewhere 19. armchair : A stool is sitting on a wooden floor next to a wall. 20. stool : A stool is located in the corner of the room. 21. Chair : A wooden stool is in the kitchen, next to a chair. **22. dinning table : A dining table is visible in the background, with chairs around it. The table appears to be made of wood and is surrounded by chairs, creating a cozy and inviting atmosphere.** 23. armchair : The armchair is white and located in the corner of the room. 0. elsewhere	 {"index" : "22", "reason" : "orange might be found on the dining table or with the chairs around it, as this is a common place for eating"}		**YES**