ChatNav: Leveraging LLM to Zero-shot Semantic Reasoning in Object Navigation

Yong Zhu, Zhenyu Wen, *Senior Member IEEE*, Xiong Li, Xiufang Shi, *Member IEEE*, Xiang Wu, Hui Dong, Jiming Chen, *Fellow IEEE*

***Abstract*—In object goal navigation tasks, the robot's understanding of semantic relationships in the environment is a key factor in its ability to localize target objects. Previously, learning-based methods trained robots using 3D scene datasets to learn semantic relationships. However, these approaches perform poorly in new environments with unfamiliar semantic contexts. In this paper, we propose ChatNav which leverages the powerful knowledge summarizing and reasoning capabilities of a Large Language Model (LLM) for zero-shot inference of explicit semantic relationships. These relationships are further integrated into the navigation system for efficient localization of target objects. ChatNav employs a spatial object clustering algorithm to collect semantic clues and designs common-sense-based prompts for interacting with LLM. It then uses a gravity-repulsion model to convert inference results into heuristic factors for robust navigation decision-making. Our approach requires no additional training and can consistently obtain accurate semantic relationships from LLM, making it well-suited for navigating unknown environments. Experimental results demonstrate the outstanding navigation performance of our proposed method on the Gibson and HM3D datasets, surpassing the current state-of-the-art object goal navigation methods.**

***Index Terms*—Object goal navigation, LLM, Object clustering, prompt, gravity-repulsion model.**

I. Introduction

In the task of visual object-goal navigation [1–5], the robot is required to navigate swiftly and accurately in an unknown environment to locate the target object effectively. In this context, the dynamic modeling and understanding of semantic relationships in the environment are crucial factors influencing the decision-making capabilities of the robot [6–8].

If the robot could perceive and reason about semantic information from the objective world, akin to human-like reasoning, it would greatly enhance its navigation performance. Approaches such as end-to-end reinforcement learning (RL) [9–12], modular RL [7, 13, 14], and supervised learning [15] have successfully integrated semantic relationships into navigation systems, thereby augmenting navigation performance.

This work was supported by the National Nature Science Foundation of China under Grant 62472387, Grant U23A20326, Grant 62372414; the China Postdoctoral Science Foundation under Grant 2023M743403; the Zhejiang Provincial Natural Science Foundation of Major Program (Youth Original Project) under Grant LDQ24F020001 *(Corresponding author: Zhenyu Wen)*

Yong Zhu, Zhenyu Wen, Xiong Li, Xiufang Shi, Xiang Wu, and Hui Dong are with the Institute of Cyberspace Security and College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China. (E-mail: yohuoyong@gmail.com, zhenyuwen@zjut.edu.cn, lx.3958@gmail.com, xiufangshi@zjut.edu.cn, xiangwu@zjut.edu.cn, hdong@zjut.edu.cn)

Jiming Chen is with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China, and Hangzhou Dianzi University, Hangzhou, China. (e-mail: jmchen@ieee.org).

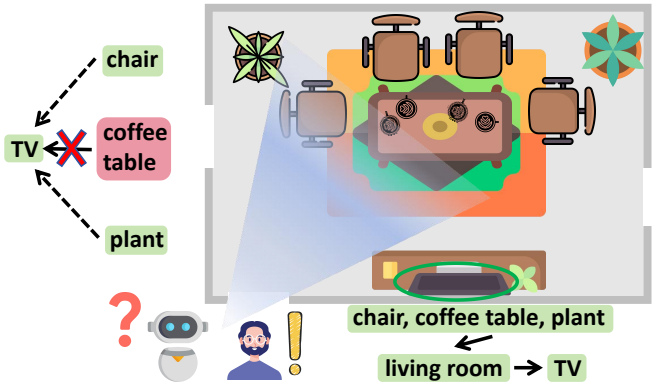


Fig. 1. The differences between humans and robots in searching for target items in unknown environments. When searching for a TV in an unknown setting, the robot's lack of knowledge about the coffee table-TV relationship forces it to speculate based on a chair and plant, often resulting in navigation errors. Humans, with a more nuanced understanding of semantic connections, would more accurately infer the presence of a TV in a living room setting.

However, these learning-based methods are limited by the training data and can only learn the semantic relationships provided by the training data.

In unfamiliar settings where novel semantic relationships are present, robots that learn these relationships through machine learning methods still fall short in exploration skills compared to humans. Take the scenario depicted in Fig. 1 for example: the robot, not having previously learned the semantic relationship between a coffee table and a TV, struggles to ascertain the presence of a TV in a room. In contrast, humans have a rich understanding of diverse semantic relationships, enabling them to engage in more advanced semantic reasoning. For instance, a human might recognize a space as a living room and, based on this recognition, infer the likely presence of a TV. To further enhance the robot's ability to leverage semantic relationships in unknown environments, we explore utilizing the powerful knowledge summarization and reasoning capabilities of large language models (LLMs) for the zero-shot acquisition of explicit spatial layout relationships between objects.

Challenges. Ensuring the stable and sensible inference of explicit object layout relationships by generative LLM and their effective integration into the navigation system still faces several challenges. 1) Providing accurate information to the LLM: In unknown environments, the lack of scene priors makes it challenging to integrate semantic information into accurate and sufficient semantic cues, which are crucial

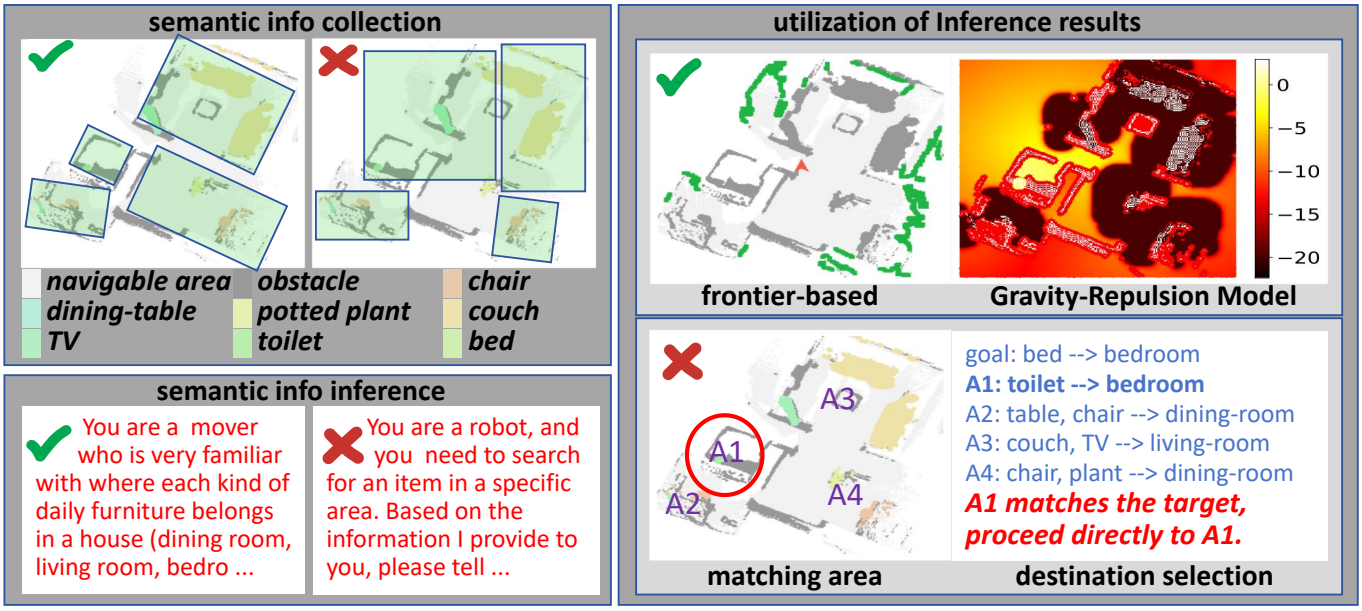


Fig. 2. Integrating LLM into Object Goal Navigation Systems. Accurate and ample semantic cues are vital for LLM to infer valid semantic relationships (up left). We propose a real-time spatial clustering algorithm for incomplete semantic maps, ensuring the collection of precise cues for LLM. The stable inference of reasonable semantic relationships by LLM is crucial for the effectiveness of the pipeline. We transform the context of the problem into a domain where LLM excels, designing prompts to guide LLM in making continuous and logical inferences of semantic relationships (bottom left). The adaptability of explicit semantic relationships to the navigation system affects navigation performance. We introduce a gravitational-repulsive model to transform semantic relationships into factors influencing frontier selection, rather than directly using inference results for navigation decisions.

prerequisites for LLM to infer ideal semantic relationships (See Fig. 2 up left). 2) Prompting the LLM for stable and rational reasoning: In a single navigation task, the robot must repeatedly extract decision information from the LLM, with each decision determining the exploration direction. However, guiding robot navigation proves to be a novel and challenging task for LLM, potentially leading to irrational reasoning outcomes. As a consequence, designing a prompt being capable of consistently steering LLM towards rational inferences poses a significant challenge (See Fig. 2 bottom left). 3) Integrating explicit semantic relationships with the navigation system: The explicit semantic layout relationships cannot be directly used for navigation decisions, and their unreasonable integration into the navigation system hinders the effectiveness of semantic guidance (See Fig. 2 right).

We introduce "ChatNav", a novel pipeline designed for object goal navigation, integrating a large LLM. This innovative approach facilitates stable and logical zero-shot inference of object semantic relationships, seamlessly integrating these insights into the navigation system. As a result, ChatNav achieves both efficient and precise navigation, significantly enhancing its operational capabilities.

Firstly, we design an object spatial clustering algorithm suitable for incomplete semantic maps[1] to collect accurate and comprehensive semantic cues for LLM (See Fig. 2, top left). Next, we design a new framework to transform the robot navigation commands to the common scene dialogue as the prompts for LLMs (See Fig. 2 bottom left). This is because the performance of LLMs is more accurate and stable in reasoning tasks that involve common-sense knowledge rather than unfamiliar new tasks such as robot navigation. Finally, we incorporate a frontier exploration algorithm [16] as a foundational component of our navigation system in ChatNav. This algorithm converts the inference outcomes into influential factors for selecting frontiers, utilizing a gravity-repulsion model as illustrated in Fig. 2 (right panel). This integration not only enhances the robustness of the navigation system but also ensures its stability. Collectively, these three pivotal and interdependent components constitute the foundational framework of the ChatNav pipeline, with each component serving an indispensable role in enhancing its overall functionality and operational efficiency.

Our contributions can be summarized as follows:

- We propose an approach leveraging a generative LLM for stable and rational zero-shot inference of object semantic relationships. Such inference utilizes an object spatial clustering algorithm suitable for incomplete maps to collect semantic cues for LLM and employs a common-sense task inference prompt template for interaction with LLM.
- To integrate the explicit reasoning results into the navigation system, we design a frontier exploration algorithm based on the gravity-repulsion model to transform these inference results into factors influencing the frontier selection, which ensures both flexibility and robustness in robot decision-making.
- We apply the proposed ChatNav pipeline to the Habitat simulation platform [17] and test it on the datasets of Gibson [18] and HM3D [19]. The experimental results indicate that, compared to the state-of-the-art (SOTA)

[1]The semantic map is a 2D top-down map storing observed object locations.

object goal navigation methods, ChatNav achieves comprehensive superiority in accuracy and efficiency without requiring additional training.

II. RELATED WORK

A. Large language model applications

In recent years, various large language models such as BERT [20], LLAMA [21], and ChatGPT [22, 23] have rapidly advanced, demonstrating zero-shot reasoning capabilities and the ability to perform various complex tasks. Particularly, since the emergence of ChatGPT 3.5, its powerful knowledge summarization, reasoning capabilities, contextual understanding, and flexible prompt interaction have garnered widespread attention from experts and scholars across various fields, who have begun applying it in their respective research domains. Dave et al. [24] conducted an analysis and summary of ChatGPT's applications in the field of medicine, highlighting its potential in this domain. Ren et al. [25] utilized ChatGPT to empower CLIP-based image classification tasks, aiming to further improve image classification accuracy. Jin et al. [26] applied the multimodal understanding capability of the GPT-4 model to the field of artistic creation, achieving promising results. Lu et al. [27] augmented ChatGPT with plug-and-play modules to accomplish various compositional reasoning tasks. Inspired by these works, we consider embedding ChatGPT into navigation systems to leverage its zero-shot inference capabilities for object spatial layout distribution reasoning, thereby assisting in navigation decision-making.

B. Learning-based object goal navigation methods

There are primarily three categories of learning-based object target navigation methods: end-to-end RL [5, 9–11], modular RL [7, 13, 14], and supervised learning [15]. In the end-to-end RL approaches, robots are trained to utilize semantic information from observed images to select navigation actions. Modular RL methods typically involve constructing a semantic map to store historical semantic information [13], and RL is used to train robots to select waypoints from the map, while path planning and action selection are executed by low-level planning algorithms. Supervised learning methods pre-train a neural network to store semantic relationships between objects and utilize these relationships for decision-making during navigation. Although these methods have successfully integrated semantic information into navigation systems and enhanced the decision-making capabilities of robots, the limited semantic relationships learned through data-driven approaches severely restrict their generalization to new scenarios. Our proposed pipeline leverages a comprehensive understanding of semantic relationships through LLM, enabling the derivation of semantic relationships for any object without the need for additional training.

C. LLM-based object goal navigation method

Recently, there has been a growing interest in utilizing low-lever language models to infer spatial semantic relationships for assisting in navigation. L3MVN [28] constructs query sentences using semantic information near map boundaries. By employing the RoBERTa LLM [20], it obtains query probabilities for each sentence as boundary scores and selects the boundary with the highest score for exploration. Esc [29] further develops two types of query sentences, namely room-level and object-level queries. Through the Deberta v3 language model [30], it obtains object query probabilities and room query probabilities. Ultimately, it makes a comprehensive decision to select the exploration boundary. Both L3MVN and Esc require obtaining query probabilities from LLM. However, current large-scale generative language models (such as ChatGPT series large language models [22, 23]) are only accessible through API calls, making it challenging to obtain query probabilities. Moreover, the large models they use only employ simple query sentences as prompts, lacking the ability to design structured prompts for interaction with LLM. This structured prompt approach allows for a more precise description of object layouts to adapt to navigation scenarios [31–33].

Dhruv et al. [34] initiated research on creating structured prompts to work with the ChatGPT 3.5 model. However, they did not thoroughly explore the adequacy and precision of the inferential clues given to the model, nor did they deeply integrate the model's inference outcomes with the navigation system. These factors are vital for improving navigation using large language models. The model needs accurate and ample inferential clues to correctly deduce semantic relationships, which is a key reason for using large language models. Additionally, how well the inferred semantic relationships mesh with the navigation system is critical for effectively applying semantic information. We design three stages of semantic relationship inference by LLM in navigation systems, achieving high-quality performance with a short prompt.

III. OVERVIEW OF CHATNAV

Object Navigation. Given a robot equipped with RGB-D observations $O := (I, D)$ and odometry measurements $\delta_p := (\delta_x, \delta_y, \delta_\theta)$, the goal is to enable it to navigate quickly to a specified target object G in an unknown environment. At each time step t, the robot executes an action $a_t \in \{turn\,left, turn\,right, move\,forward, stop\}$ based on the current observations $O_t := (I_t, D_t)$ and pose $p_t := p_0 + \delta_{p_0} + \delta_{p_1} + \delta_{p_2} + ... + \delta_{p_{t-1}}$. Navigation is considered complete when the robot reaches within a range of d_s meters from the target object G, and then the robot executes the stop action.

Global and Local Policy. Following previous works [13, 15, 28], we construct the navigation system using global and local policies. The global navigation policy sets a global goal point P_i for the robot, while the local policy controls the robot to execute a series of actions $(a_t, a_{t+1}, ...)$ to move towards P_i. The robot is guided to purposefully explore unknown regions by repeatedly setting global goal points, enabling it to locate target objects quickly.

To fully utilize semantic relationships in navigation decision-making, we designed the ChatNav pipeline. After each robot action, ChatNav collects observation data. At fixed

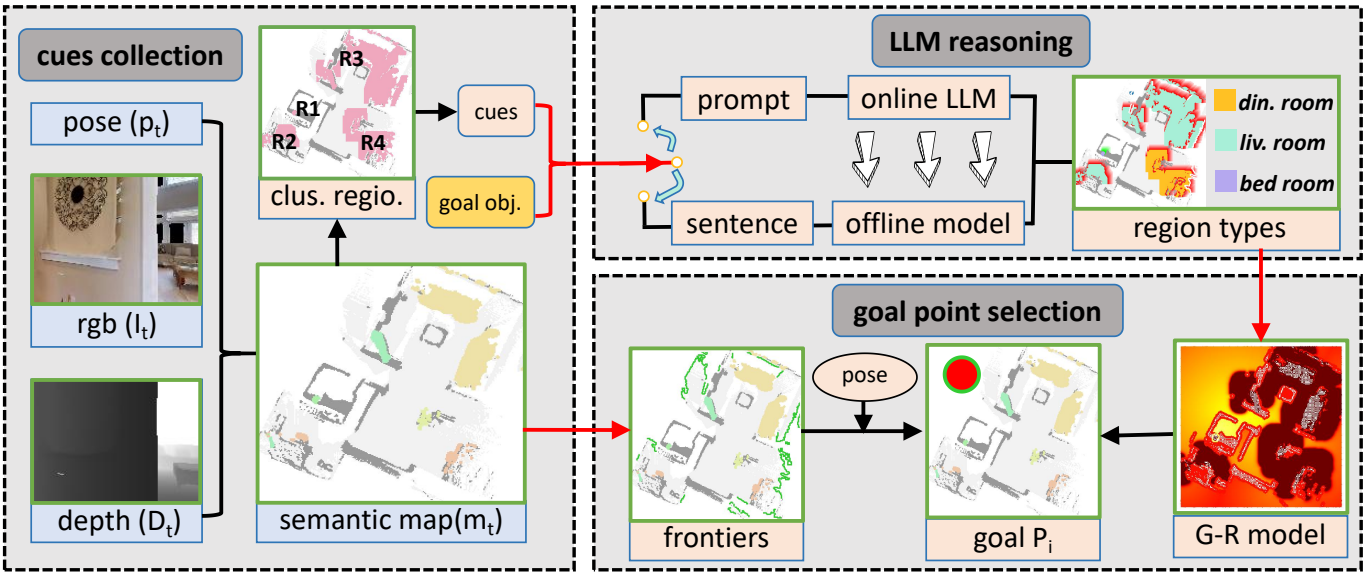


Fig. 3. The framework of ChatNav. The sensors collect real-time pose and RGB-D data. The ChatNav pipeline utilizes this data to construct a semantic map and formulate global target points based on the map. Specifically, we first employ a clustering algorithm on the incomplete map to obtain accurate and comprehensive semantic cues. Then, we combine the semantic cues with common-sense knowledge to form a prompt for interaction with the LLM (or a locally distilled model from online LLM), resulting in reasoning outcomes. Finally, these reasoning results are transformed into selection factors for exploration frontiers through a gravity-repulsion model, ultimately leading to the acquisition of global target points.

intervals T, it uses an LLM to infer semantic relationships and set global goal points P_i. For the local policy, we apply the same deterministic method as in SemExp [13] to plan the path and guide the robot to the global goal point.

The Pipeline of ChatNav. As shown in Fig. 3, the pipeline consists of three parts: cues collection, LLM reasoning, and goal point selection. Cues collection: At each time step, ChatNav converts observation information into the semantic map. When a new global goal point needs to be set, it conducts semantic clustering on this map to identify various candidate regions. Each region offers specific semantic cues, determined by the types of objects present in them, as detailed in §IV. LLM reasoning: ChatNav utilizes the semantic cues of the candidate regions to interact with the LLM, thereby obtaining the spatial relationship between these regions and the target object. This process is further explained in §V. Goal point selection: ChatNav constructs a gravity-repulsion model based on the spatial relationships inferred by the LLM. This model is then combined with a frontier exploration algorithm to select global goal points, as detailed in §VI.

IV. Clustering-Based Regional Semantic Cue Generation

In environments lacking prior map information, robots struggle to integrate observed semantic information into effective cues, a prerequisite for LLM to infer reasonable semantic relationships. We build a semantic map to store the semantic information observed by the robot at each time step. When setting a global goal point, we use a clustering algorithm to transform the historical semantic information in the map into semantic cues.

Semantic Map. The semantic map, denoted as m_t, is an $M \times M$ 2D matrix, containing accumulated observations from time step 0 to t. Different numbers in $m_t(i,j)$ correspond to different meanings: 1 and 2 correspond to explored areas and obstacles, and other numbers correspond to different categories of objects. At $t = 0$, the semantic map is initialized as a zero matrix, with the robot placed at the center of the map, oriented towards the x-axis. At each subsequent time step, the robot updates the semantic map with the RGB-D observations (I_t, D_t) obtained from its pose p_t.

This update process is illustrated in Fig. 4. Firstly, the robot uses a semantic segmentation network to label each pixel of RGB image I_t as either a specific object category or background [35]. Then, the robot converts each pixel in the depth map d_t into a corresponding 3D point cloud and assigns the category obtained from the semantic segmentation, forming a semantic point cloud, $PC(x, y, z, category)$. Finally, the robot transforms the semantic point clouds into the semantic map coordinate system, projects them downward, and updates the semantic map m_t.

To achieve the conversion mentioned above from the depth map to the point clouds, we employ a classic geometric transformation method, as described by Equation (1).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = d(x', y') \begin{bmatrix} \frac{1}{f_x} & 0 & 0 \\ 0 & \frac{1}{f_y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' - \frac{w}{2} \\ y' - \frac{h}{2} \\ 1 \end{bmatrix} \tag{1}$$

where (x', y') are the pixel coordinates, (w, h) are the dimensions of the depth map, (x, y, z) are the corresponding point cloud coordinates, $d(x', y')$ is the depth at (x', y'). f_x and f_y are the known focal lengths of the camera. Similarly, we use classical coordinate transformation and projection methods

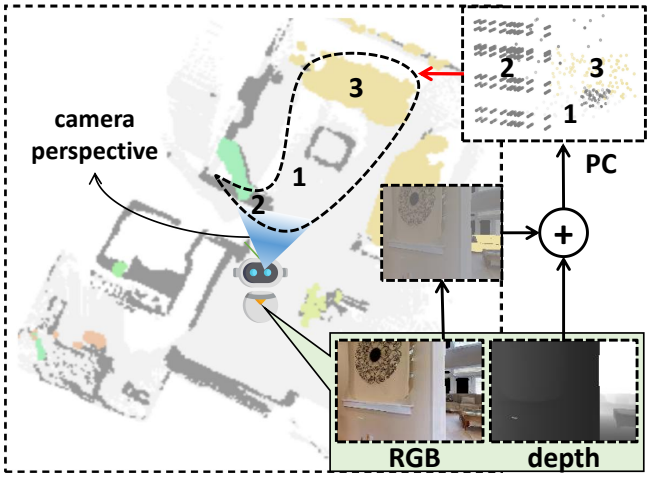


Fig. 4. Projection of Robot Observations onto the Semantic Map. At time t, the robot captures observations with an RGB-D camera and integrates them into the semantic map. First, a semantic segmentation network identifies a couch in the RGB image. Next, the depth map is converted into 3D point clouds with attached semantic information, forming semantic point clouds. Finally, the point clouds are projected onto the semantic map using differential projection. In the semantic map, the dashed area represents new content from the current RGB-D observation, with 1 indicating the explored area, 2 representing the obstacle, and 3 denoting the couch.

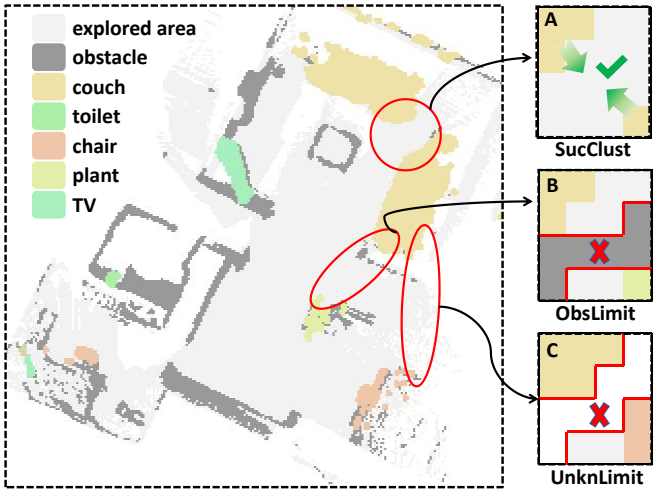


Fig. 5. A spatial object clustering algorithm suitable for incomplete maps. In this incomplete semantic map, we use obstacle and explored regions as clustering limitations to maximize object clustering under correct assumptions. In the case of A, two nearby couches cluster successfully without these constraints. In case B, obstacles between a couch and a plant prevent clustering, treating them as separate regions. In case C, a large unexplored area between a couch and a chair also leads to their separate regions.

[13] to map the semantic point clouds onto the semantic map.

$$\begin{bmatrix} X \\ Y \end{bmatrix} = Proj. \left(\begin{bmatrix} \mathbf{R^{3\times 3}} & \mathbf{t^{3\times 1}} \\ \mathbf{1^{1\times 3}} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \right) \tag{2}$$

where $[X, Y]$ represents the 2D coordinates on the map. $\mathbf{R^{3\times 3}}$ denotes the rotation matrix and $\mathbf{t^{3\times 1}}$ represents the translation vector between the camera and map coordinate systems. $\mathbf{R}$ and $\mathbf{t}$ are determined by factors such as the camera's pitch angle, height, and the robot's pose.

Clustering Algorithm for Incomplete Map. Understanding some objects in a room allows for better guessing of other room characteristics. The more types of objects identified in the room, the more precise these guesses are. Consequently, we have created an algorithm for object spatial clustering. This algorithm groups objects in the same region, giving the LLM more precise and complete cues for semantic reasoning. Different objects are considered to be in the same region based on three basic conditions: 1) The objects are spatially close to each other, such as the 2 couches in Fig. 5. A; 2) no obstacles are separating the objects, unlike the couch and plant in Fig. 5. B; 3) There is a known connecting passage (explored area) between the objects, unlike the couch and chair in Fig. 5. C. To ensure the clustering of objects consistently satisfies these three constraints, we design a clustering strategy based on object dilation. To satisfy condition 1), we perform dilation on objects in the map, connecting nearby objects into a single region to achieve clustering. To meet conditions 2) and 3), we perform the dilation in multiple small steps, incrementally expanding the objects and blocking any trends that cross obstacles and unexplored regions after each step.

Specifically, we first decompose the semantic map into three distinct components: the explored map, the obstacle map, and the object map. Then, we perform N iterations of convolutional dilation using a small $c \times c$ kernel on the object map. After each dilation operation, we apply the obstacle and explored maps as masks to the dilated object map. These masking operations can eliminate abnormal dilation clustering that crosses obstacles and unknown regions. Finally, we obtain multiple candidate regions that accurately encompass objects, with these objects serving as the semantic cues for their respective regions. Taking Fig. 5 as an example, there exists semantic information {table, chair, TV, couch, toilet, table, plant}. After the clustering operation, we obtain region-cue pairs: $R_1 = [toilet], R_2 = [table, chair], R_3 = [couch, TV], R_4 = [table, plant]$.

In the dilation clustering process, the dilation steps N and kernel size $c \times c$ are two critical parameters. The kernel size should be small enough to avoid crossing obstacles or unexplored regions, which could lead to objects from different regions being incorrectly clustered. After selecting an appropriate kernel size, the dilation steps N will determine the extent of the dilation clustering. If N is too small, objects in the same area cannot be adequately clustered. Conversely, if N is too large, objects from different regions may be incorrectly clustered. To select an appropriate N, we design two metrics to evaluate the clustering performance, i.e., the average clustering rate ACR and the average mis-clustering rate $AMCR$. We experiment with the complete semantic maps of Gibson training set scenes [18], testing the clustering performance for different values of N. Before clustering, each object is treated as an individual chunk. After dilation, these chunks merge into larger and fewer chunks. The ACR measures the reduction in the number of chunks before and after N iterations of dilation clustering, while the $AMCR$

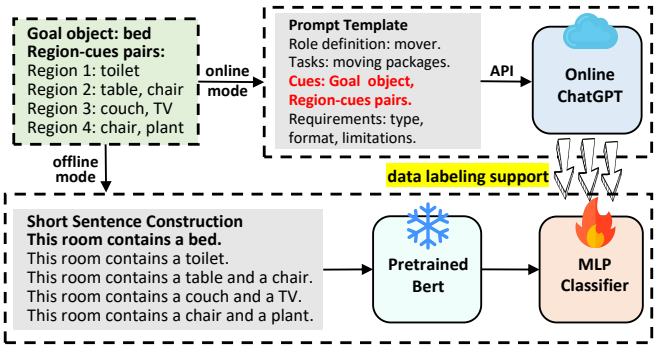


Fig. 6. The usage modes of ChatNav in online and offline settings. In the online mode, we integrate clustering results into common-sense prompts for the ChatGPT model, enabling the LLM to derive reasoning outcomes. In the offline mode, we transform the clustering outcomes into simple sentences, extract sentence embeddings using a pre-trained RoBERTa-large model, and classify them with an MLP classifier. The MLP is trained based on data collected in the online mode.

represents the ratio of the abnormal chunk dilation area (such as the overlapping area created when chunks extend from the living room to the bedroom) to the total map area. Our goal is to select a dilation step N that maximizes ACR and minimizes $AMCR$. The formulas for calculating ACR and $AMCR$ are as follows:

$$ACR = \frac{1}{S}\sum_{i=1}^{S}\sum_{j=1}^{R(i)}\frac{1}{R(i)}\frac{C_{orig.}(i,j) - C_{clus.}(i,j)}{C_{orig.}(i,j)} \tag{3}$$

Where S is the number of scenes, $R(i)$ is the number of regions in the i_{th} scene, $C_{clus.}(i,j)$ is the number of chunks in the j_{th} region of the i_{th} scene after dilation clustering, and $C_{orig.}(i,j)$ is the number of chunks in the corresponding region before clustering.

$$AMCR = \frac{1}{S}\sum_{i=1}^{S}\frac{A_{mis}(i)}{H_{map}(i) \times W_{map}(i)} \tag{4}$$

Where $A_{mis}(i)$ is the mis-clustering area in the i_{th} scene, and $(H_{map}(i), W_{map}(i))$ are the dimensions of the complete scene map.

V. Semantic Reasoning with LLM

After obtaining the semantic cues of the candidate regions, we construct prompts based on these cues and interact with the ChatGPT model by calling the API online, as shown in the online mode in Fig. 6, to determine the semantic relationship between the target object and each region. Additionally, we offer an offline reasoning solution for situations where there is no network connection, as depicted in the offline mode in Fig. 6.

Online Mode. In this mode, the design of the prompt is crucial, directly determining the accuracy and stability of ChatGPT's semantic relationship reasoning during navigation. Our prompt design leverages two key features of the ChatGPT.

First, ChatGPT is a general-purpose model trained on common interactions and articles, excelling at solving common-sense problems but prone to hallucinations in specialized domains [22, 23]. To address this, we switch from a professional robot navigation task to a common-sense knowledge task. Instead of asking the LLM to navigate as a robot, we ask it to act as a mover, placing packages into rooms. Second, ChatGPT generates content autoregressively based on the prompt, and the completeness of the prompt description helps improve the quality of the generated content. To this end, we organize the prompt in a structured way, including the role, tasks, cues, and requirements, to form a coherent narrative.

Specifically, as shown in the prompt template in Fig. 6 for the online mode, we first create a commonsense context of a mover distributing packages, where the packages represent candidate or target regions, and their contents correspond to semantic cues. Next, within this context, we direct the LLM to predict the type of room in which each package should be placed, thereby obtaining room-type labels of candidate regions and target region. Finally, we add inference constraints for LLM to improve the validity of LLM inference information and develop a specific output format to facilitate the extraction of valid information.

We pass this complete prompt to the ChatGPT model through the API for inference, obtaining the types of candidates and target regions. By comparing the candidate region types with the target region type, we categorize the candidate regions as target-related or target-unrelated. In Fig. 7, we present practical examples of ChatGPT reasoning under different types of prompts. In the commonsense context, the model can accurately link the target object to each region based on room types. However, under the professional prompt, ChatGPT provides incorrect navigation guidance due to its lack of expertise in robotic navigation tasks.

Additionally, to maintain system stability, we develop two special handling prompts: abnormal handling prompt and multi-target disambiguation prompt, as shown in Figure 8. When ChatGPT's response contains content or format anomalies, we pass the abnormal handling prompt to ChatGPT, instructing it to respond strictly according to the requirements. When ChatGPT identifies multiple target-related regions, we use the multi-target disambiguation prompt to further interact with ChatGPT, asking it to determine the single target-related region.

Offline Mode. When network conditions are poor or rapid decision-making is required, incorporating spatial layout knowledge into a local model becomes crucial. To achieve this, we apply a distillation technique that combines the local model with fine-tuning from ChatGPT, transferring ChatGPT's knowledge of object spatial layouts into the local model. Specifically, we first place the robot randomly in two training datasets—Gibson (25 scenes) and HM3D (146 scenes)—and assign random destinations for it to navigate, collecting extensive object clustering data. This data collection simulates real-world robot exploration. For each set of clustering data (e.g., *clus1, clus2, clus3...*), we infer the room type of each cluster (e.g., C_1, C_2, C_3 ...) following the same method as in the online mode shown in Fig. 6, thus generating a large number of labeled clusters. Using the labeled data, we then train the local model. As shown in the offline mode of Fig. 6, we follow this process: construct simple sentences for each

A. Common sense prompt: You are an experienced mover who is familiar with where each piece of furniture belongs in a house. One day, you are asked to deliver a batch of packages containing household items to a house that consists of a dining room, a living room, and a bedroom. You need to select a storage room for each package based on the layout of the furniture in the home (multiple packages can be stored in one room). In these packages, Please provide separate answers for each package's storage room. Package 1 contains [toilet], package 2 contains [chair, table], package 3 contains [couch, TV], package 4 contains [table, plant], and package 5 contains [bed]. Please respond in the following JSON file format: {"package no": num, "package items": [], "target room": room type}, In the JSON format mentioned above, num is the package number, fill in the items with the corresponding package number in []; room type is the room type (Choose one type from the dining room, living room, or bedroom, and do not include additional room types such as bathroom or kitchen. You are only allowed to place the package in the three rooms mentioned earlier.)
Response of A: [{"package no": 1, "package items": ["toilet"], "target room": "bedroom"}, {"package no": 2, "package items": ["chair", "table"], "target room": "living room"}, {"package no": 3, "package items": ["couch", "TV"], "target room": "living room"}, {"package no": 4, "package items": ["table", "plant"], "target room": "dining room"}, {"package no": 5, "package items": ["bed"], "target room": "bedroom"}]

B. Professional prompt: You are a robot who is required to quickly explore an unknown environment to find a target item. One day, you are asked to find an item in an unknown environment with a household layout consisting of a dining room, living room, and bedroom. I will tell you what you see, and you should answer which region to go to find the item. Please first explain your reasoning, then provide a response in JSON format. The JSON format is as follows: {"target area no": num, "area items": []}. In the JSON format mentioned above, num is the package number, fill in the items with the corresponding package number in []; if there is no target area, reply [no] and return an empty JSON response. Below are a few examples: Example 1: {Information: You now see two areas. Area 1 has a couch; Area 2 has a bed; You need to find a toilet. Which area should you go to? Answer: I should go to Area [2], because there is a bed, likely near the bedroom, and toilets are usually found around bedrooms. Therefore, I will go to Area [2]. {"target area no": 2, "area items": [bed]}}. Example 2: {Information: You see three areas. Area 1 has a couch, a TV; Area 2 has a chair and a table; Area 3 has a toilet; You need to find a plant. Which area should you go to? Answer: I should go to Area [2], because Area 2 has a chair and a table, which is likely near the dining room, and plants are usually found near dining rooms. Therefore, I will go to Area [2]. {"target area no": 2, "area items": [chair, table]}}. Example 3: {Information: You see two areas. Area 1 has a bed; Area 2 has a toilet; You need to find a potted plant. Which area should you go to? Answer: [no], because the given Area 1 is clearly near the bedroom, and Area 2 is likely near the bedroom as well. Potted plants are usually placed in the dining room. Therefore, I will explore a new area. {}}. When answering the reason, the type of room can only be living room, bedroom, or dining room. You are not allowed to infer any other types, such as bathroom or kitchen. Information: You see 4 areas. Area 1 has a toilet; Area 2 has a chair and a table; Area 3 has a couch and a TV; Area 4 has a plant and a chair. You need to find a bed. Which area should you go to?
Response of B: I should go to Area [3], because Area 3 has a couch and a TV, which are commonly found in the living room. Beds are typically located in the bedroom, and the living room is often adjacent to or near the bedroom in household layouts. Therefore, I will go to Area [3]. {"target area no": 3, "area items": [couch, TV]}

Fig. 7. Common sense prompt and professional prompt: In the common sense prompt, the LLM is asked to play the role of a mover, where the cues for the candidate regions are represented as packages, and the LLM is required to infer the type of room each package should be moved to. In the professional prompt, the LLM is asked to play the role of a robot and directly infer the most likely region for the target object based on the region-cue pairs. In the above prompts, the text in red is replaced based on the actual target and regional clues. For example, with the target object being a bed and the region-cue pairs being $R_1 = [toilet]$, $R_2 = [chair, table]$, $R_3 = [couch, TV]$, and $R_4 = [plant, table]$, we construct prompts accordingly. The two prompts interact with the ChatGPT model separately. From the responses, it is evident that, under common-sense conditions, the correct area classification can be obtained. However, under the professional prompt, ChatGPT failed to infer the relationship between the toilet and the bedroom, leading to an incorrect answer.

cluster, obtain embeddings for these sentences using the pre-trained RoBERTa-large model, and feed these embeddings into a classifier to predict the local model's labels (e.g., c_1, c_2, c_3 ...). We compare these predicted labels with those from ChatGPT to compute the loss (e.g., l_1, l_2, l_3 ...), where $l_i = \textit{Cross-entropy}(\textit{one-hot}(C_i) - c_i)$, and update the classifier network accordingly. Ultimately, we obtain a local object to region type classifier.

VI. Utilization of Semantic Relationships

Since LLM's inference of candidate regions is not based on the robot's actual observations, directly using the inferred target-related region as the navigation target area can result in system crashes. Instead, these inferences should serve as guidance for navigation decision-making. To achieve this, we combine the inferred semantic relationships with exploration boundaries for comprehensive decision-making. Specifically, we first establish a gravity-repulsion model to compute the semantic utility values at every position on the map. Then, we utilize the traditional frontier exploration algorithm [16] and cost-utility method [36] to obtain all frontiers and their corresponding geometric utility values. Finally, we combine semantic utility and geometric utility to identify semantic frontiers, selecting the optimal semantic frontier as the next exploration target.

Gravity-Repulsion Model. Target-unrelated regions exert the repulsive force on the robot, preventing it from further exploration in these regions, and encouraging the robot to explore new, unknown areas instead. Meanwhile, the target-related region exerts an attractive force on the robot, drawing it deeper into exploring that region. However, the impact of target-related and target-unrelated regions on the robot should exhibit distinct characteristics. The repulsion should inhibit the robot from exploring target-irrelevant areas and their immediate vicinity, without affecting other exploration processes. However, attraction should manifest in target-relevant areas and their broader vicinity. Therefore, we design the repulsion function and gravity function in the form of (5) and (6).

$$V_{repu}(P, A_i) = a \cdot e^{-\alpha \cdot (dist(P, A_i) + \beta)} \tag{5}$$

$$V_{grav}(P, G) = k \cdot dist(P, G) + b \tag{6}$$

The (5) computes the repulsion force generated by target-unrelated region A_i at point P, and $dist(P, A_i)$ represents the length of the shortest path from point P to region A_i. We calculate this length using the classic Fast Marching Method (FMM) [37]. β represents the truncation distance, a is the magnitude parameter, and α is the distance scaling parameter. The (6) calculates the attraction force generated by the target-related region G at point P. $dist(P, G)$ represents the length of the shortest path from point P to the target-related region,

A. Abnormal Handling Promp: Based on your response, we cannot extract the effective relationship between rooms and packages. Please adhere strictly to the format we provided earlier to ensure we can extract valid information.
B. Multiple Target Disambiguation Prompt: You are an experienced mover who is familiar with where each piece of furniture belongs in a house. One day, you are asked to deliver a batch of packages containing household items to a house that consists of a dining room, a living room, and a bedroom. You need to select a storage room for each package based on the layout of the furniture in the home (multiple packages can be stored in one room). In these packages, Please provide separate answers for each package's storage room. Package 1 contains [chair, TV], and package 2 contains [couch, TV]. Please respond in the following JSON file format: {"package no": num, "package items": [], "target room": room type}, In the JSON format mentioned above, num is the package number, fill in the items with the corresponding package number in [], and room type presents the type of room. Note: One of these two packages goes in the living room, the other in the different room.

Fig. 8. Abnormal handling prompt and multiple target disambiguation prompt. When ChatNav detects an abnormal response, it sends the abnormal handling prompt to the LLM, explaining the error and the inability to extract useful information. If multiple target-related regions appear, ChatNav extracts the region-cue pairs corresponding to all target-related areas and constructs the multiple target disambiguation prompt to interact with the LLM, identifying the unique target-related region. For example, if the target object is in the living room and two target-related regions are inferred: R1 = chair, TV and R2 = couch, TV, we will construct the prompt shown in the figure above and further interact with the LLM.

k represents the correlation decay coefficient, and b serves as the offset. To calculate the shortest path distance from a point to a region, we first set up the regions and obstacles for the FMM algorithm. The FMM algorithm spreads outwards from the region's boundary, changing direction when encountering obstacles, until it covers the entire map. The arrival time at each point estimates the shortest path distance.

Finally, we combine the gravity and repulsion forces to calculate the field force, as shown in formula (7). N_A represents the number of target-irrelevant regions, and *flag* is a 0-1 variable, only becoming 1 when a target-relevant region is present.

$$V(P) = \sum_{i=1}^{N_A} V_{repu}(P, A_i) + flag \cdot V_{grav}(P, G) \tag{7}$$

Semantic Frontier. The frontier refers to regions on the boundary between open space and unexplored space. By moving to new frontiers, the robot can extend its map into new territory until the entire environment has been explored [16]. Traditional frontier-based exploration algorithms utilize geometric information in the map to select frontiers, thereby achieving rapid exploration of the entire environment. However, in object goal navigation tasks, the purpose of exploration is to locate target objects, and the selection of frontiers should be more targeted. Therefore, we combine the semantic relationships inferred by LLM with the frontiers to form semantic frontiers and select the center of the optimal semantic frontier as the global target point. Specifically, it consists of the following three steps:

1) Utilize obstacle maps and exploration maps to obtain frontiers. Initially, we extract the boundaries of the exploration map and perform dilation to connect adjacent boundaries.

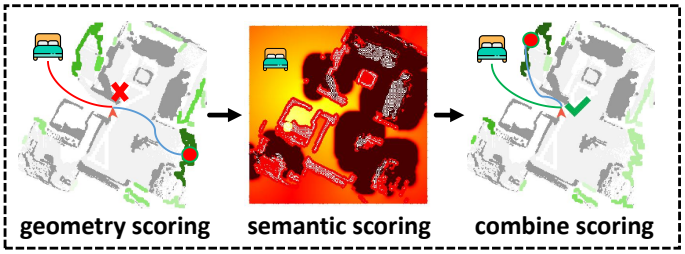


Fig. 9. Gravity-Repulsion Field Affecting Frontier Utility: The left image shows frontiers scored by geometric utility, with the robot selecting the highest-scoring frontier at the red dot, opposite the target object (bed). We use a gravity-repulsion field to obtain semantic scores for each map position (middle). The right image combines these scores to select the frontier center with the highest composite score as the global goal point, marked by the red dot, located near the target object (bed).

Subsequently, we use the obstacle map to eliminate invalid portions. Finally, we further filter out boundaries that are too small for the robot to navigate through, resulting in the exploration frontiers denoted as $F = \{f_1, f_2, f_3...\}$.

2) Combine the gravity-repulsion field formed by semantic relationships with the frontiers, resulting in semantic frontiers. Each semantic frontier has three attributes: $f_i = [Dist(f_i, pose), N(f_i), V(f_i)]$, where $Dist(f_i, pose)$ is the relative shortest path length from the frontier center to the current pose, calculated with FMM, $N(f_i)$ is the pixel count of the frontier, and $V(f_i)$ is the field force value exerted by the gravity-repulsion field at the center of this frontier.

3) Reference the cost-utility method outlined in paper [36], and augment it with semantic components. The utility value for each frontier is calculated as shown in (8).

$$U(f_i) = \frac{N(f_i)}{C_N} + \lambda_1 \frac{Dist(f_i, pose)}{C_D} + \lambda_2 \frac{V(f_i)}{C_V} \tag{8}$$

where C_N, C_D, C_F are scaling factors for pixel count, path length, and field force values respectively, λ_1 and λ_2 are used to adjust the relative importance of geometric distance and semantic field force. Finally, the frontier with the highest utility value is chosen as the target frontier, and its center is chosen as the global target point P, as shown in the formula (9).

$$f_P = \text{argmax}_i\{U(f_1), U(f_2)...U(f_i)...\} \tag{9}$$

In Fig. 9, we illustrate the change in the robot's frontiers scoring before and after the influence of semantic components. The semantic components assist the robot in avoiding unnecessary exploration, enabling a swift and accurate localization of the target object.

VII. Discussion

Computational Complexity Analysis. The ChatNav pipeline consists of various components: traditional methods (e.g., object space clustering), neural networks (e.g., semantic segmentation), and online API calls to LLM (e.g., interactions with ChatGPT). Thus, we separately compute the computational complexity of ChatNav. Specifically, we use algorithm complexity to evaluate traditional methods, average inference time for neural networks, and average response time for

TABLE I
COMPUTATIONAL COMPLEXITY ANALYSIS.

Executed at every step			Executed at fixed intervals T			
Semantic map update			Clust.	LLM reason.	Goal point selec.	
PC gene.	Coord. tra.	Sem. seg.	—	—	Frontier.	R-G mod.
$O(n^2)$	$O(n^2)$	$0.023s$	$O(n^2)$	$1.825s$	$O(n^2)$	$O(n^2 \log n)$

LLM interactions. Additionally, since certain pipeline parts are executed at every time step while others are executed at fixed interval T, we distinguish between them in our complexity analysis. The results are shown in the TABLE I.

PC generation: Converting a depth map to point clouds involves transforming each pixel in a $w \times h$ depth map into a corresponding point cloud, with a computational complexity of $O(n^2)$. The computation for a single point is as formula (1). Coordinate translation: Transforming a point cloud from the camera coordinate system to the semantic map coordinate system involves converting each of the $w \times h$ points. This process has a computational complexity of $O(n^2)$. The transformation for a single point is given by the formula (2). Semantic segmentation: Using a pre-trained Mask-RCNN on $w \times h$ RGB images, the average inference time on an NVIDIA 3090 Ti GPU was 0.0232 seconds per image (based on 500 images). Semantic clustering: The object spatial clustering algorithm involves multiple convolutions with a small kernel on a $m \times m$ semantic map, masked by obstacles and exploration areas, with a computational complexity of $O(n^2)$. LLM reasoning: For 100 interactions with ChatGPT, the average time from prompt to response is 1.825 seconds. Frontier detection: Executing edge detection on a $M \times M$ semantic map has a complexity of $O(n^2)$. Repulsion-gravity model: The core involves the classic Fast Marching Method [37] for shortest path calculation, with a complexity of $O(n^2 \log n)$.

LLM for ObjNav Future Research Directions. ChatNav is the first attempt to integrate LLM with a navigation system, allowing the robot to perform zero-shot reasoning about spatial relationships between objects, which significantly enhances navigation performance. However, this is just the initial application of LLM in navigation. By incorporating advanced technologies such as multimodal large models [38–41] and LLM agents [42], we can further improve the system's intelligence and precision. For instance, multimodal large models can analyze RGB images to gain a deeper understanding of environments, such as identifying room types or inferring that a room with a glass door is likely a bathroom, thereby increasing exploration efficiency. Meanwhile, LLM agents can centralize decision-making across modules, integrating LLM into tasks like semantic map construction and clustering, which enhances ChatNav's overall intelligence and robustness.

VIII. EXPERIMENT

In this section, we conduct comparative and ablation experiments in the high-resolution Habitat simulator [17]. We assess our pipeline's performance by comparing it with other object goal navigation methods and validate the roles of individual modules by substituting them within the pipeline. Additionally, we provide a more detailed examination of how key parameters and the type of LLM affect navigation performance.

A. Experimental Setup

Datasets. We conduct our experiments on real-world 3D reconstruction scene datasets: Gibson [18] and HM3D [19]. The Gibson dataset comprises 25 train / 5 val scenes from the Gibson tiny split, with successful navigation defined when the robot is within 0.1m of the target object. The HM3D dataset includes 145 train / 36 val scenes, with successful navigation defined when the robot is within 0.15m of the target object. We configure 1000 episodes on the validation sets of both datasets for performance evaluation, while setting 5000 episodes on the training dataset to collect clustering data for training the offline mode of semantic relationship reasoning. In both datasets, the target objects consist of six categories: chair, couch, potted plant, bed, toilet, and TV.

Configuration Details. We utilize the same robot configuration and a similar semantic map construction setup as SemExp [13]. We emphasize different configurations and parameter settings during the ChatNav execution process. The robot uses fine-tuned MASK-RCNN [35] to recognize 15 common objects. The semantic map is 480×480 in size, with each cell representing a $5cm \times 5cm$ physical area. The hyperparameters for the gravitational function are set to $[k = 5, b = 0]$, while those for the repulsion function are $[a = 0.3, \alpha = -0.5, \beta = -30]$. The relative importance parameters (λ_1, λ_2) are set to $(0.2, 0.2)$, and the scaling factors C_N, C_D, and C_V are all set to 1. Semantic relationship reasoning is conducted using the ChatGPT Turbo-3.5 model. The interval for calling the LLM to determine global goal points is set to $T = 30$. We check for new semantic information in the semantic map before each call to avoid unnecessary LLM calls. If new semantic information is detected, we query the LLM to update the semantic relationships in the map; otherwise, we use the previous LLM inference results.

For the clustering algorithm, we set the kernel size to 3×3 and test different N values on the complete semantic maps of 25 Gibson training scenarios [18]. As shown in Fig. 10, $N = 15$ achieved high ACR and low AMCR, indicating good clustering performance. Thus, we set N to 15 for further experiments.

Evaluation Metrics. In addition to the three classic object navigation metrics—Success Rate (**SR**), Success weighted by Path Length (**SPL**), and Distance to Goal (**DTG**) [13, 15, 28]—we introduce two time-based metrics, Average Time (**AT**) and Average Time Steps (**ATS**), to compare the navigation performance of ChatNav with other baselines. **SR** refers to the percentage of episodes where the target object is successfully located among the total episodes. **SPL** is an indicator focused on successful episodes and means the efficiency of the robot's actual path relative to the oracle's shortest path [43]. SPL is defined as

$$SPL = \frac{1}{N} \sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)} \tag{10}$$

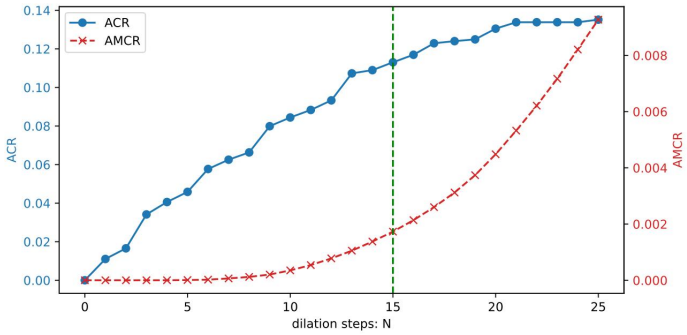


Fig. 10. Evaluation of clustering performance with different dilation steps N. When testing clustering performance with N values ranging from 0 to 25, both ACR and $AMCR$ gradually increase. At $N = 15$, ACR is already at a high value while $AMCR$ remains relatively low, indicating good clustering performance.

where N represents the number of episodes, l_i denotes the oracle shortest path length from the starting position of the robot to the target object, p_i is the actual path length expected by the robot, and S_i is a 0-1 variable indicating navigation success, where S_i equals 1 in case of successful navigation. **DTG** represents the average distance between the robot and the target object at the end of episodes. **ATS** is the average time steps per episode in the simulator, representing the number of actions the robot executes since it performs one action per time step. **AT** is the average real-time in seconds spent per navigation episode, reflecting both the system's algorithmic efficiency and the robot's time steps (ATS) to find the target.

An example of a scenario. Fig. 11 illustrates a complete robot navigation process of searching a bed. ChatNav effectively guides the robot to explore and ultimately locate the target bed.

B. Baselines

We design five baselines to compare their navigation performance with ChatNav.

- **Random Global Target (RGT):** The global target point is randomly sampled on the entire map, and a deterministic policy guides the robot to move toward the global target point.
- **Random Frontier Target (RFT):** Compared to RGT, RFT randomly selects a target frontier from all frontiers on the map, with the center of the chosen target frontier serving as the global target point.
- **SemExp [13]**: SemExp integrates semantic information into state representation and employs reinforcement learning to train the robot in selecting global target points on the training sets of the Gibson and MP3D datasets. The global target points are sampled across the entire map and are not based on frontiers. It marked the winning entry in the Habitat ObjectNav challenge during CVPR 2021.
- **PONI [15]:** PONI utilizes the Gibson and MP3D datasets along with their corresponding semantic annotations to construct a supervised learning dataset. Through supervised training, it acquires both geometric and semantic relationships. These relationships are then used for decision-making during the navigation process.
- **L3MVN [28]:** L3MVN is the first to use a Bert language model [20] for explicit reasoning of object layout relationships and selecting frontiers as global goal navigation points based on these relationships. It introduces two usage modes for the Bert model: zero-shot (L3MVN-Z) and feed-forward (L3MVN-F). L3MVN-Z constructs simple sentences by combining objects around the frontier with the target object and then employs Bert to score the sentences, thus selecting the optimal frontier. L3MVN-F utilizes the MP3D dataset to train a small-scale classifier, which predicts upcoming categories based on known categories of objects near the frontier, thus establishing a link between the target object and the frontier.

C. Result and Discussion

Classic Object Navigation Metrics Analysis. The comparative results are shown in TABLE II. The RFT method (row 2 in TABLE II) significantly improves all three metrics on both datasets compared to the RGT method (row 1 in TABLE II). This indicates that setting the global target point at the frontier is advantageous for the robot to explore unknown environments. The significant improvement in navigation performance observed with the SemExp and PONI methods (row 3 and row 4 in TABLE II) clearly illustrates the advantage of leveraging semantic relationships to enable robots to make more effective navigation decisions. L3MVN further enhances navigation performance (row 5 and row 6 in TABLE II), indicating that using language models to reason about semantic relationships and incorporating them into navigation decisions is reliable in object goal navigation. Our proposed ChatNav method outperforms 3 learning-based methods—SemExp, PONI, and L3MVN-F (row 7 in TABLE II). This is attributed to our approach not requiring training on specific datasets to acquire semantic relationships, thereby demonstrating enhanced generalization capabilities in new and unknown environments. ChatNav (row 7 in TABLE II) and L3MVN-Z (row 6 in TABLE II) acquire semantic relationships from LLM through zero-shot learning and participate in navigation decisions. However, ChatNav consistently outperforms the latter on both datasets, suggesting that ChatNav can obtain more accurate semantic relationships and incorporate them into the navigation system more reasonably. Comparing ChatNav and LocChat (row 8 in TABLE II), the offline mode requires no network and has a shorter reasoning time, yet the navigation performance is hardly affected across the three metrics.

Time-based Metrics Analysis. As shown in TABLE III, our ChatNav and LocChat methods rank first and second respectively in ATS, and third and second in AT. This indicates that the algorithmic execution efficiency of our proposed pipeline is relatively high, and the proportion of effective actions performed by the robot is enhanced. Although SemExp achieves the lowest AT, its ATS is among the highest, suggesting that while its algorithmic efficiency is high, the robot performs many redundant navigation actions.

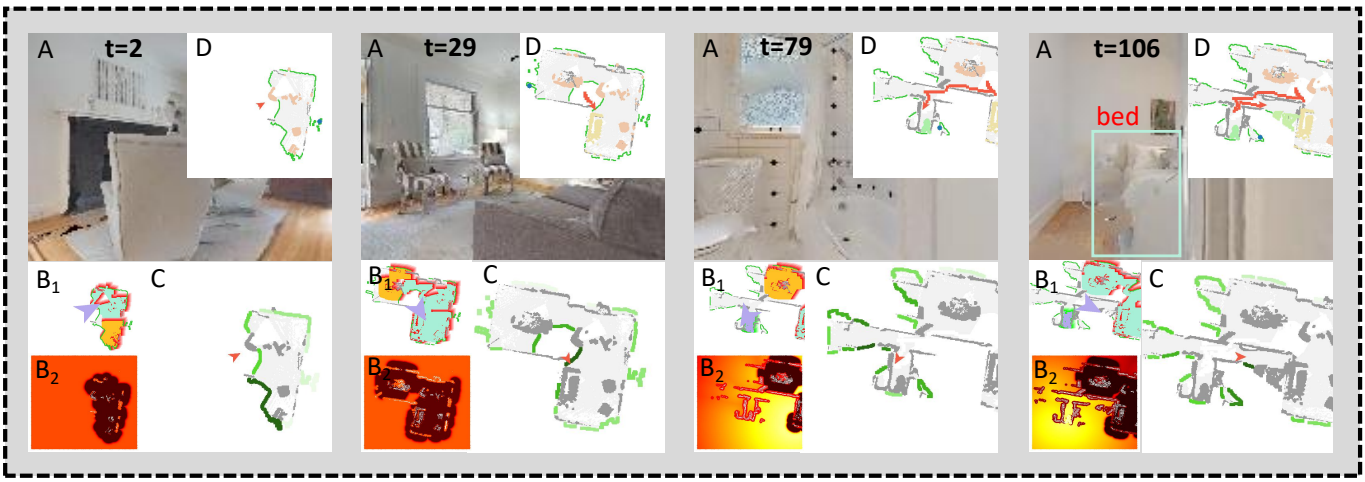


Fig. 11. The process of the robot searching for a bed. In this context, A depicts the robot's first-person view, B_1 shows the matching result between the clustered regions and the target region, while B_2 illustrates the gravity-repulsion field based on this matching. C displays the frontier's geometric score, and D shows the selection of the global goal point, integrating both geometric and semantic utility. At time step $t = 2$, the LLM identifies two regions the living room and dining room, which do not match the target region (bedroom). Both of the regions are represented as repulsive forces. At $t = 29$, the target region still hasn't appeared, and all regions continue to exhibit repulsive forces. At $t = 79$, the robot discovers the toilet and identifies its region as the target-relevant region, causing an attractive force. The robot successfully discovers the bed at $t = 106$, guided by the attractive force.

TABLE II
CHATNAV OUTPERFORMS ALL BASELINES IN NAVIGATION.

	Gibson(val)			HM3D(val)		
Method	Succ. ↑	SPL ↑	DTG ↓	Succ. ↑	SPL ↑	DTG ↓
RGT	0.661	0.325	1.650	0.207	0.088	3.142
RFT	0.709	0.395	1.218	0.385	0.170	3.057
SemExp	0.717	0.396	1.39	0.431	0.200	3.128
PONI	0.736	0.410	1.250	0.401	0.197	3.239
L3MVN-F	0.764	0.471	1.220	0.594	0.256	2.370
L3MVN-Z	0.756	0.417	1.174	0.561	0.242	2.598
ChatNav	**0.826**	**0.495**	**0.867**	**0.595**	**0.267**	**2.367**
LocChat	0.817	0.490	0.872	0.591	0.262	2.384

TABLE III
CHATNAV DEMONSTRATES THE BEST OVERALL TIME PERFORMANCE.

time	RGT	RFT	SemExp	PONI	L3-F	L3-Z	ChatNav	LocChat
AT	31.5	28.2	**16.4**	38.7	32.9	36.9	24.1	22.4
ATS	175.95	163.45	161.82	158.92	147.98	146.74	**130.15**	138.18

D. Ablation Studies

Importance of Each Module. To demonstrate the impact of each component of the proposed pipeline on navigation performance, we conducted ablation experiments on the Gibson and HM3D datasets: frontier exploration algorithm (F-only), clustering algorithm (Clus.), LLM interaction mode based on common-sense background (Comm.), and gravity-repulsion model (G-R). The analysis of the experimental results in TABLE IV is as follows:

1) The complete ChatNav structure achieves the best navigation performance (row 6 in TABLE IV). 2) Frontier exploration is essential for effective exploration in unknown environments, as removing the frontier exploration algorithm results in nearly all metrics performing the worst (row 5 in TABLE IV). 3) Properly applying semantic cues to the navigation system significantly improves navigation SR and SPL (row 1 in Table IV). 4) Removing the clustering algorithm leads to incomplete cues provided to the LLM, causing a decline in all system metrics (row 2 in TABLE IV). 5) Interacting with the LLM using common-sense prompts significantly improves navigation speed on the HM3D dataset, as LLMs excel at handling common-sense questions (row 3 in Table IV). 6) The removal of the gravity-repulsion model results in a significant drop in navigation performance, indicating that properly utilizing semantic relationships is crucial (row 4 in TABLE IV).

TABLE IV
MODULE ABLATION STUDIES PROVE THE CHATNAV'S EFFICACY.

ChatNav ablation				Gibson(val)			HM3D(val)		
F-only	clus.	comm.	G-R	Succ. ↑	SPL ↑	DTG ↓	Succ. ↑	SPL ↑	DTG ↓
✓				0.787	0.474	1.058	0.586	0.267	2.367
✓		✓	✓	0.806	0.475	0.893	0.581	0.257	2.472
✓	✓		✓	0.819	0.482	0.874	0.590	0.179	2.385
✓	✓	✓		0.803	0.46	0.962	0.574	0.262	**2.331**
	✓	✓	✓	0.762	0.452	1.295	0.549	0.246	2.520
✓	✓	✓	✓	**0.826**	**0.495**	**0.867**	**0.595**	**0.267**	2.377

Key Hyperparameters Analysis. To further enhance the interpretability of the ChatNav pipeline design, we conducted additional experimental analysis on the key parameters and model types involved in the pipeline on the Gibson dataset. Specifically, we investigated the impact of dilation iterations in clustering N, geometric importance parameters λ_1, and semantic importance parameters λ_2.

We conduct four comparative experiments with N set to 0, 5, 15, and 25. Among these settings, $N = 0$ corresponds to no clustering, $N = 5$ to insufficient clustering, N = 15 to optimal clustering performance, and $N = 25$ to severe clustering errors. As shown in TABLE V, the best navigation performance is achieved with N = 15.

We conduct three sets of experiments by adjusting the im-

TABLE V
NAVIGATION PERFORMANCE IS BETTER WHEN N IS SET TO 15.

	Gibson(val)		
N	Succ. ↑	SPL ↑	DTG ↓
0	0.806	0.475	0.893
5	0.811	0.481	0.887
15	**0.826**	**0.495**	**0.867**
25	0.791	0.476	1.032

portance parameters for semantics and geometry. The (λ_1, λ_2) values were set to 0.1 and 0.3 to prioritize semantic relationships, 0.3 and 0.1 to prioritize geometric relationships, and 0.2 and 0.2 to treat both semantic and geometric relationships with equal importance in navigation. The results, shown in TABLE VI, indicate that navigation performance is optimal when both semantic and geometric information are considered equally.

TABLE VI
NAVIGATION PERFORMANCE IS OPTIMAL WHEN BOTH SEMANTIC AND GEOMETRIC INFORMATION ARE CONSIDERED EQUALLY.

	Gibson(val)		
(λ_1, λ_2)	Succ. ↑	SPL ↑	DTG ↓
(0.1, 0.3)	0.812	0.485	0.877
(0.3, 0.1)	0.822	0.492	0.871
(0.2, 0.2)	**0.826**	**0.495**	**0.867**

Impact of LLM Type. We conduct four comparative experiments using ChatGPT-3.5, ChatGPT-4.0, and RoBERTa-large [20] models to evaluate ChatNav's navigation performance. ChatGPT-3.5 and 4.0 have larger parameter sizes and superior text processing capabilities than RoBERTa-large. For RoBERTa-large, we tested two modes: Bert-Z (zero-shot inference) and Bert-F (fine-tuned for downstream tasks, as in LocChat). The experimental results, as shown in TABLE VII, lead to two conclusions: first, comparing row 1, row 2, and row 3, row 4, the stronger the reasoning capability of the large model, the better the navigation performance; second, comparing row 1 and row 2, fine-tuning less capable language models improves navigation performance.

TABLE VII
THE HIGHER THE MODEL'S PERFORMANCE, THE BETTER THE NAVIGATION RESULTS.

	Gibson(val)		
LLM	Succ. ↑	SPL ↑	DTG ↓
Bert-Z	0.812	0.482	0.931
Bert-F	0.817	0.490	0.872
Chatgpt3.5	0.826	0.495	**0.867**
Chatgpt4	**0.827**	**0.497**	0.871

IX. CONCLUSION

We propose a pipeline for object goal navigation, called ChatNav, which integrates a LLM. It enables stable and reasoned zero-shot inference of object semantic relationships and effectively incorporates them into the navigation system, thereby achieving efficient and accurate navigation. We evaluated the performance of our pipeline on two high-resolution 3D datasets, Gibson and HM3D. Our pipeline demonstrated superior accuracy and speed without the need for any training, outperforming current state-of-the-art methods. In the future, we plan to further explore the local zero-shot semantic relationship acquisition to address the issue of frequent network dependency on accessing large model APIs [44].

REFERENCES

[1] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[2] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020.

[3] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "Minos: Multimodal indoor simulator for navigation in complex environments," *arXiv preprint arXiv:1712.03931*, 2017.

[4] B. Chen, J. Kang, P. Zhong, Y. Cui, S. Lu, Y. Liang, and J. Wang, "Think holistically, act down-to-earth: A semantic navigation strategy with continuous environmental representation and multi-step forward planning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[5] X. Hu, Y. Lin, S. Wang, Z. Wu, and K. Lv, "Agent-centric relation graph for object visual navigation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[6] M. Lingelbach, C. Li, M. Hwang, A. Kurenkov, A. Lou, R. Martín-Martín, R. Zhang, L. Fei-Fei, and J. Wu, "Task-driven graph attention for hierarchical relational object navigation," *arXiv preprint arXiv:2306.13760*, 2023.

[7] N. Gireesh, A. Agrawal, A. Datta, S. Banerjee, M. Sridharan, B. Bhowmick, and M. Krishna, "Sequence-agnostic multi-object navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9573–9579.

[8] S. Chen, T. Chabal, I. Laptev, and C. Schmid, "Object goal navigation with recursive implicit maps," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7089–7096.

[9] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," *arXiv preprint arXiv:1911.00357*, 2019.

[10] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu *et al.*, "Learning to navigate in complex environments," *arXiv preprint arXiv:1611.03673*, 2016.

[11] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2616–2625.

[12] Y. Bai, X. Song, W. Li, S. Zhang, and S. Jiang, "Long-short term policy for visual object navigation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9035–9042.

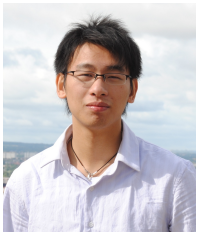
[13] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4247–4258, 2020.

[14] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *arXiv preprint arXiv:2004.05155*, 2020.

[15] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal

navigation with interaction-free learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 890–18 900.
[16] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.
[17] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9339–9347.
[18] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9068–9079.
[19] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," *arXiv preprint arXiv:2109.08238*, 2021.
[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
[21] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.
[22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback, 2022," *URL https://arxiv. org/abs/2203.02155*, vol. 13, 2022.
[23] B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.
[24] T. Dave, S. A. Athaluri, and S. Singh, "Chatgpt in medicine: an overview of its applications, advantages, limitations, future prospects, and ethical considerations," *Frontiers in Artificial Intelligence*, vol. 6, p. 1169595, 2023.
[25] Z. Ren, Y. Su, and X. Liu, "Chatgpt-powered hierarchical comparisons for image classification," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[26] C. Jin, R. Zhu, Z. Zhu, L. Yang, M. Yang, and J. Luo, "Mtartgpt: A multi-task art generation system with pre-trained transformer," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.
[27] P. Lu, B. Peng, H. Cheng, M. Galley, K.-W. Chang, Y. N. Wu, S.-C. Zhu, and J. Gao, "Chameleon: Plug-and-play compositional reasoning with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[28] B. Yu, H. Kasaei, and M. Cao, "L3mvn: Leveraging large language models for visual target navigation," *arXiv preprint arXiv:2304.05501*, 2023.
[29] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," *arXiv preprint arXiv:2301.13166*, 2023.
[30] P. He, J. Gao, and W. Chen, "Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing," *arXiv preprint arXiv:2111.09543*, 2021.
[31] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.
[32] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.
[33] S. Ekin, "Prompt engineering for chatgpt: a quick guide to techniques, tips, and best practices," *Authorea Preprints techrxiv.22683919.v2*, 2023.
[34] D. Shah, M. R. Equi, B. Osiński, F. Xia, B. Ichter, and S. Levine, "Navigation with large language models: Semantic guesswork as a heuristic for planning," in *Conference on Robot Learning*. PMLR, 2023, pp. 2683–2699.
[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
[36] M. Juliá, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Autonomous Robots*, vol. 33, pp. 427–444, 2012.
[37] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts." *proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.
[38] W. Li, H. Fan, Y. Wong, Y. Yang, and M. Kankanhalli, "Improving context understanding in multimodal large language models via multimodal composition learning," in *Forty-first International Conference on Machine Learning*, 2024.
[39] Y. Zhang, H. Fan, and Y. Yang, "Prompt-aware adapter: Towards learning adaptive visual tokens for multimodal large language models," *arXiv preprint arXiv:2405.15684*, 2024.
[40] X. Yang, F. Lv, F. Liu, and G. Lin, "Self-training vision language berts with a unified conditional model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3560–3569, 2023.
[41] G. Zhang, B. Zhong, Q. Liang, Z. Mo, N. Li, and S. Song, "One-stream stepwise decreasing for vision-language tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2024.
[42] H. Du, S. Zhang, B. Xie, G. Nan, J. Zhang, J. Xu, H. Liu, S. Leng, J. Liu, H. Fan *et al.*, "Uncovering what why and how: A comprehensive benchmark for causation understanding of video anomaly," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 793–18 803.
[43] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.
[44] L. Lu, Z. Wen, Y. Yuan, B. Dai, P. Qian, C. Lin, Q. He, Z. Liu, J. Chen, and R. Ranjan, "iquery: A trustworthy and scalable blockchain analytics platform," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4578–4592, 2023.

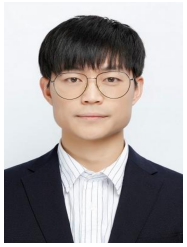
Yong Zhu received the B.S. degree from Hunan University of Science and Technology, Xiangtan, China, in 2021. He is currently pursuing the Ph.D. degree in control engineering with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His current research direction includes the application of large language models, robotic systems, and deep learning.



Zhenyu Wen (Senior Member, IEEE) is currently the Tenure-Tracked Professor with the Institute of Cyberspace Security, and College of Information Engineering, Zhejiang University of Technology. His current research interests include IoT, crowd sources, AI systems, and cloud computing. For his contributions to the area of scalable data management for the Internet of Things, he was awarded the IEEE TCSC Award for Excellence in Scalable Computing (Early Career Researchers) in 2020.

Xiong Li received the B.S. degree from Jiangxi University of Science and Technology, Ganzhou, China, in 2020. He is currently pursuing the Ph.D. degree in control theory and control engineering with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His current research interests include unmanned system security and machine learning.

Xiufang Shi (Member, IEEE) received the B.Sc. degree in automation from the East China University of Science and Technology, Shanghai, China, in 2011, and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2016. She was the joint Ph.D. student with the University of Sydney, Sydney, NSW, Australia, in 2015. From 2016 to 2018, she was the Postdoctoral Researcher with the College of Control Science and Engineering, at Zhejiang University. She is currently the Associate Professor with the College of Information Engineering, Zhejiang University of Technology, Hangzhou. Her major research interests include wireless localization, target tracking, the Internet of Things, and statistical signal processing.



Xiang Wu received a B.E. degree in automation, the M.E. degree in control engineering, and the Ph.D. degree in control theory and control engineering from the Zhejiang University of Technology, Hangzhou, China, in 2012, 2015, and 2020, respectively. His research interests include disturbance rejection, networked motion control systems, and cloud control systems.

Hui Dong received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2007. He is currently a Professor at the College of Information Engineering, Zhejiang University of Technology, Hangzhou. His current research interests include the industrial Internet of Things, intelligent manufacturing, and motion control.

Jiming Chen (Fellow, IEEE) received the Ph.D. degree in control science and engineering from the Zhejiang University, Hangzhou, China, in 2005. He is currently a Professor at the Department of Control Science and Engineering, at Zhejiang University. His research interests include IoT, networked control, and wireless networks. He serves on the editorial boards of multiple IEEE Transactions. He is an IEEE VTS Distinguished Lecturer and a fellow of CAA.