

CogNav: Cognitive Process Modeling for Object Goal Navigation with LLMs

Supplementary Material

This document offers a detailed explanation of our approach, along with additional experimental results, presented in the following structure:

- Technical details (§A)
- More Experiments (§B)

A. Technical details

A.1. Notation in paper

We provide a lookup table of notations and their description mentioned in this paper in Table 1 for reference.

A.2. Cognitive Map in paper Sec.3.1.

Details of Scene Graph. The scene graph \mathcal{S}_t comprises instance nodes \mathcal{N}_t and relationship edges \mathcal{E}_t . The construction of node \mathcal{N}_t is based on an online open-vocabulary segmentation method [10] to obtain the 3D instances. Unlike [10], we employ a more accurate and stable detection and segmentation framework, OpenSEED [17] as our foundation model and replace point cloud object representation with a voxel grid for clearer object boundaries and better alignment with the 2D occupancy map \mathcal{M} . At each time t , we detect and segment objects from I_t^{rgb} and reconstruct a voxel grid in global coordinates by I_t^{depth} and θ_t synchronously. Each node \mathcal{N}_t contains the 3D voxel grid coordinates, the detected category, the semantic features of the instance, and the semantic features corresponding to the detected category. The fusion strategy in [10], which simply matches and fuses newly detected objects at time t with the object nodes \mathcal{N}_{t-1} from previous frames, may suffer from under-segmentation. This limitation can lead to under-detection of small objects, posing challenges for navigation tasks. To address over-segmentation and under-segmentation issues, we utilize Vision-Language Models (VLMs). To uniquely represent each instance $n \in \mathcal{N}_t$ in the scene graph, we utilize SoM [14], which masks and marks the instance with a node number at the center of its mapping in the image. For ambiguous cases where it is unclear whether two instances should be merged, we use SoM to generate an image containing both instances, as illustrated in Figure 1(a).

Objects that do not match any existing instances from previous frames are treated as new instances and are added as separate nodes in \mathcal{N}_t .

After constructing \mathcal{N}_t , we leverage VLMs with spatial context of objects to update edges \mathcal{E}_t . We only update edges between object fused or created at time t with surrounding objects considering the time cost. Edges that can be reasoned from spatial context will be directly computed like 'on top of' or 'under', edges that cannot be reasoned from spatial context are obtained by querying the VLMs. We use SoM [14] to mask and mark the two instances whose relationship needs to be queried on an image. The VLM is then prompted with this image to determine the relationship between the two objects, considering observations from two different viewpoints. The VLM selects one of the candidate set { 'next to', 'on top of', 'inside of', 'under', 'hang on' }, with an additional 'none' indicating that two nodes have no relationship. We show the edge generation by VLM in Figure 1(b).

Details of Landmark Graph. At time t , we construct a landmark graph \mathcal{L}_t using a semantic occupancy map \mathcal{M}_t to guide agent navigation. The landmark graph \mathcal{L}_t is derived from a processed Reduced Voronoi Diagram(RVD) extracted from the occupancy map \mathcal{M}_t . We first construct a top-down semantic occupancy map \mathcal{M}_t using I_t , similar to [7]. It is represented as a $M \times M \times 3$ tensor with $M \times M$ as map size and 3 channels including an obstacle and an explored channels from I_t^{depth} , with an semantic map, where each mask is aligned with the voxel grid of nodes \mathcal{N}_t and each channel stores the unique label of the aligned node. The semantic occupancy map is updated in real-time with each frame during navigation.

After constructing the semantic occupancy map, we refer [16] to cluster frontiers and refer [13] to extract a Generalized Voronoi Diagram(GVD) by skeletonizing the traversable areas. We remove redundant nodes and retain only the intersections and all leaf nodes in GVD to generate a Reduced Voronoi Diagram(RVD), different from the reduction strategy in [13], which only keeps the frontier end points. We propose a landmark-instance mapping \mathcal{D}_t such that each leaf node in RVD corresponds to the node label or

Notation	Type/Unit	Description
t	number	a time step
c	string	the category of the target object
\mathcal{I}_t^{rgb}	Matrix ($640 \times 480 \times 3$)	a RGB image by agent observation at t
\mathcal{I}_t^{depth}	Matrix (640×480)	depth image by agent observation at t
p_t	$\langle x_t, y_t, \theta_t \rangle$	current pose of agent, including the planar coordinates and rotation angle at t
\mathcal{I}_t	$\langle \mathcal{I}_t^{rgb}, \mathcal{I}_t^{depth}, p_t \rangle$	posed RGB-D image at t
a_t	an action	an action that agent executes in scene
n_t	dict	an instance dict in scene, including its semantic and spatial context at t
e_t	a spatial relationship between two n_t	a relationship in ['next to', 'on top of', 'inside of', 'under', 'hang on']
\mathcal{N}_t	$\{n_t\}$	the set of instance n_t in scene at t
\mathcal{E}_t	$\{e_t\}$	the set of relationship e_t between two nodes in \mathcal{N}_t at t
\mathcal{S}_t	$\langle \mathcal{N}_t, \mathcal{E}_t \rangle$	the scene graph composed of \mathcal{N}_t and \mathcal{E}_t at t
M	number	The map size with a grid resolution of 5cm
\mathcal{M}_t	Matrix($M \times M \times 4$)	the top-down occupancy map at t
$l_{t,i}$	$\langle x_{t,i}, y_{t,i} \rangle$	a 2D location coordinate extracted from a Voronoi node graph at t
\mathcal{L}_t	$\{l_{t,i}\}$	a landmark graph extracted from the Voronoi node graph at t
\mathcal{C}_t	$\langle \mathcal{S}_t, \mathcal{M}_t, \mathcal{L}_t \rangle$	a coordinate-aligned heterogeneous cognitive map at t
\mathcal{D}_t	dict	a dict mapping the leaf node to n_t or a frontier area
$\gamma_{t,i}$	string	the room type l_i belongs to at t
$\delta_{t,i}$	bool	the frontier property of l_i at t
$l_{t,now}$	$\langle x_{t,now}, y_{t,now} \rangle$	a 2D coordinate which the agent locates at
n_c	dict	a node in \mathcal{S}_t whose object category is c
$\mathcal{L}_{t,frontier}$	set	The subset of \mathcal{L}_t whose $\delta_{t,i}$ is True
$\mathcal{L}_{t,relative}$	set	The subset of \mathcal{L}_t who are relative to the target object
$\mathcal{L}_{t,target}$	set	The subset of \mathcal{L}_t who are directed to n_c
$\mathcal{L}_{t,verify}$	set	The subset of \mathcal{L}_t who are directed to n_c and not explored before

Table 1. Summary of Mathematical Notation

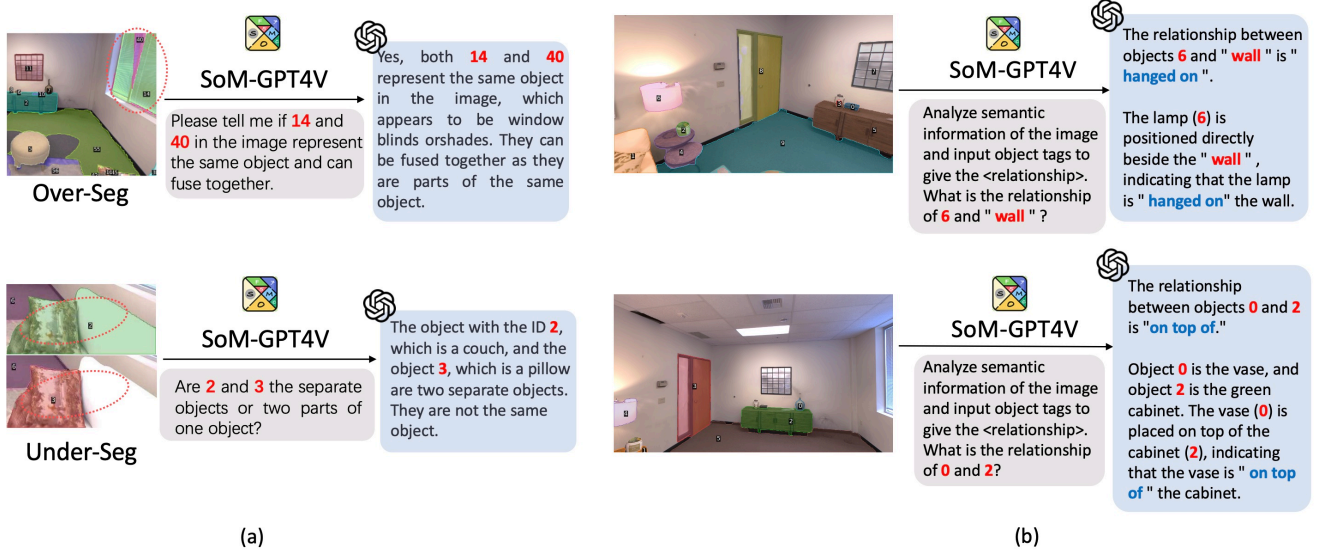


Figure 1. **Fusion and Relationship Query in Cognitive Map.** We use GPT-4v prompted by SoM image to solve the problem of fusion and relational edges in Cognitive Map. Column (a) shows how to confirm whether two partitions are the same instance. Column (b) shows the relationship generation between two adjacent instances.

frontier area in semantic map pointed to by the edge connecting this leaf node in the graph. We fuse leaf nodes that point to the same object or frontier and are close to each other on RVD, and ensure that at least two leaf nodes point to the same object, which helps in the candidate verification process in Sec.3.3. The landmark graph \mathcal{L}_t at time t is generated by fusing leaf nodes and subsequent reduction strategy on RVD, where each node $l \in \mathcal{L}_t$ is a reachable landmark that agent can navigate to as a long-term goal. Our GVD has a residual situation due to the fact that it only collects information about the scene from a single view, and therefore needs to update with the navigation process, unlike [13] that uses panorama views to be able to build a complete GVD in the area centered on the agent.

A.3. Cognitive Map Prompting in paper Sec.3.2.

Map-to-landmark prompts. For each landmark $l_{t,i} \in \mathcal{L}_t$, we integrate the scene information to understand the environment of landmark $l_{t,i}$. The information includes surrounding objects along with their relationships, the room type $\gamma_{t,i}$ and the frontier property $\delta_{t,i}$. We select object nodes $\mathcal{N}_{t,l_i} \subseteq \mathcal{N}_t$ within a certain distance threshold from landmark $l_{t,i}$ in the scene graph \mathcal{S}_t as the surrounding objects. Each node is prompted not only with its own information, including category name and semantic information but also with the information of its adjacent nodes within \mathcal{S}_t , linked by edges that define the relative relationship. This prompting approach leverages the structural information of scene graph to provide a deeper understanding of surrounding environment. The room type $\gamma_{t,i}$ is determined by querying the Vision-Language Model(VLM), which selects the most suitable option from a set of indoor room candidates based on the current image and surrounding objects. This result is then combined with the room types of nearby landmarks that have already been queried. The frontier property $\delta_{t,i}$ is a boolean variable obtained by querying the landmarks which point to the frontier in the landmark-instance mapping \mathcal{D}_t .

Agent-to-landmark prompts. The navigable message is encoded relative to the landmark $l_{t,now}$ where the agent is currently located. We calculate the distance and path from $l_{t,now}$ to $l_{t,i}$ on the landmark graph \mathcal{L}_t by means of Dijkstra Algorithm. The distance, combined with the direction, forms the navigable message of landmark $l_{t,i}$, e.g. as "Location: Direction: Up, Path: $2 \rightarrow 0 \rightarrow 1 \rightarrow 3$, Distance: 3.2m". For navigation history, we set up a boolean variable explored property $\varepsilon_{t,i}$ representing whether the agent has explored this landmark before, avoiding repetitive exploration.

The detail description of our state transition prompt and landmark selection prompt are shown in Figure. 4 and Figure. 5, respectively.

LLM	VLM	HM3D		
		SR (%) \uparrow	SPL (%) \uparrow	DTG (m) \downarrow
LLaMa3	CogVLM2	62.6	17.2	2.357
GPT-3	CogVLM2	64.1	18.3	1.882
LLaMA3	GPT-4v	69.4	22.2	1.694
QwQ-32B	GPT-4v	70.9	23.6	1.583
GPT-3	GPT-4v	72.5	26.2	1.255

Table 2. Comparison of various large language models on HM3D [12]. We replace GPT models [4, 5] in CogNav with Llama3.1-8B-Instruct [8] and Cogvlm2-llama3-chat-19B [11].

A.4. State-related landmarks in paper Sec.3.3.

- **Broad Search (BS):** The landmark candidate in state BS are landmarks where the frontier property is True: $\mathcal{L}_{t,frontier} = \{l_{t,i} \in \mathcal{L}_t \mid \delta_{t,i} = \text{True}\}$.
- **Contextual Search (CS):** The landmark candidate is changed to the landmarks which surrounding the relative instance node n_r or the room type γ_r which may contain the target object: $\mathcal{L}_{t,relative} = \{l_{t,i} \in \mathcal{L}_t \mid \mathcal{D}_t(l_{t,i}) = n_r \vee \gamma_{t,i} = \gamma_r\}$.
- **Observe Target (OT):** In state OT, landmarks in candidate are whose value of instance node which category is the target one c in landmark-instance mapping \mathcal{D}_t : $\mathcal{L}_{t,target} = \{l_{t,i} \in \mathcal{L}_t \mid \mathcal{D}_t(l_{t,i}) = n_c\}$.
- **Candidate Verification (CV):** When state CV is transformed, the landmark candidate selects landmarks in state OT however excludes landmarks explored before: $\mathcal{L}_{t,verify} = \{l_{t,i} \in \mathcal{L}_t \mid \mathcal{D}_t(l_{t,i}) = n_c \wedge \varepsilon_{t,i} = \text{False}\}$.
- **Target Confirmation (TC):** While the instance n_c is confirmed to be the target, the landmark closest to n_c is chosen as the next term goal: $l_{t,target} = \arg \min_{l_{t,i} \in \mathcal{L}_{t,target}} d(l_{t,i}, n_c)$, where $d(l_{t,i}, n_c)$ is the ground distance between the object n_c and the landmark $l_{t,i}$. The navigation ends while the agent achieves this landmark.

B. More Experiments

B.1. Results on Different LLMs of HM3D dataset

We further replace the LLM and VLM to evaluate the reasoning and decision-making abilities of different large models in navigation. As shown in Table 2, GPT (row 1) demonstrates superior analysis and decision-making capabilities in navigation tasks. Note that, after replacing it with two open-source large models (rows 1-3), our method still outperforms others on HM3D.

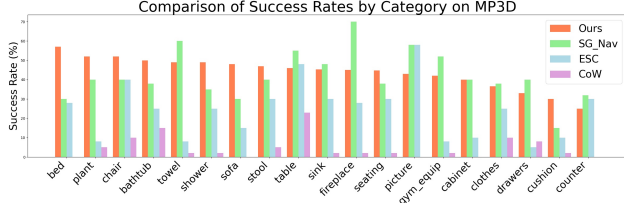


Figure 2. The comparison of each category’s success rate on MP3D [6] with other three methods.

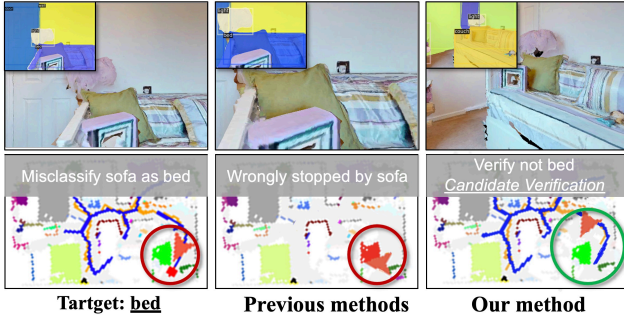


Fig 3. Target sofa is misclassified to bed (Col.1), previous methods totally believe the result and stop when approach (Col.2). While, our method verify it in another view and correct the error (Col.3).

B.2. More Results of MP3D dataset

We calculate the success rate (SR) of each category on the MP3D dataset [6] and compare with three methods Cow [9], ESC [18] and SG_Nav [15]. The results are demonstrated in Figure 2. Our success rate of category *fireplace* and *picture* are lower than SG_Nav [15] because of the unrecognizable of these two categories in some viewpoints. In other categories, our framework have the highest success rate, especially in category *bed*, *plant*, *bathtub* and *sofa*, our success rate metric is far superior to other three methods. More qualitative results can be found in the first lines of Figure 6.

We also provide an example of navigational reasoning in Fig. 3, where our method demonstrates error correction capability (with candidate verification) compared to previous approaches.

For fail case, we analyze in HM3D (Val) with 277/1008, among which 35.7% are due to persistent failures in object detection, 64.3% because of target inaccessibility (*e.g.* mesh artifacts, stairs).

B.3. Time Analysis of Our Framework

We measured the time overhead of different components in our framework, calculating the average time per execution step and the average frequency per episode for each component. The results, summarized in Table 3, represent averages across all episodes on HM3D. Note that our time overhead computation includes failure cases where the step

limit of 500 is reached, leading to an increased number of LLM queries. While the GPT-4V query process incurs a higher time overhead per query, its lower frequency minimizes its overall impact. The scene graph updating process accounts for the majority of the time overhead, primarily due to the high frequency of segmentation and detection operations performed by OpenSEED [17]. Despite these factors, our framework ensures effective real-time navigation in unknown environments.

Components	Avg. Time (s)	Avg. Frequency
Scene graph Updating	2.1	230
Landmark Graph Updating	0.85	26
Room-type Query with GPT-4v	1.8	23
Cognitive Map Prompting	0.6	15
Cognitive Process Modeling with GPT-4v	3.4	15

Table 3. The average time and frequency cost in each part of all episodes on HM3D.

B.4. Real-World Experiment Details

Real-World Setup. To deploy CogNav in the real world, we build a custom robot comprising an automated guided vehicle, robotic arm, an RGB-D camera and a master control computer. For the automated guided vehicle, we utilized Water II [3], which features a single-wire lidar for implementing indoor navigation algorithms and localization. The robotic arm selected is the E05 [1], providing flexibility and convenience for adjusting camera poses. The RGB-D camera used is the Microsoft Azure Kinect DK [2]. The master control computer are equipped with an Intel® Core™ i7-10700K @ 2.9GHz × 8. We run the memory building phase on the control workstation with an Nvidia GeForce RTX 3090Ti GPU with 24GB of memory, paired with an Intel® Core™ i9-12900K @ 3.9GHz × 16 and 32GB of RAM. The system operates on ROS Noetic Ninjemys as the software platform. In an object navigation episode, the master control computer transmits the RGB-D data stream and the 2D pose from the automated guided vehicle to the control workstation via ROS topics. The workstation processes this information as input for CogNav. Upon receiving the target landmark’s 2D pose from CogNav, the workstation directs the robot to move to the specified location. These operations repeat iteratively until the target object is successfully located. We also validate our pipeline on a Unitree quadruped, the navigation processes are demonstrated in the demo video.

Qualitative Results. To simulate a home environment, we organized a room of approximately 60m² into distinct functional areas, including a work area, a rest area, and a dining area, separated by partitions. To validate the feasibility of CogNav on robots, we carefully selected target objects of varying sizes, categories, and properties within the scene. Additional qualitative results from the real-world experiments are shown in the last line of Figure 6.

Cognitive Map Prompting for State Transition

Goal: Sofa

Current State: Broad Search

Agent Now: Landmark 11

The Landmarks:

Landmark 7:

Frontier Landmark: False

Surrounding objects: a desk which next to a chair, under a flower and next to the wall .

Room: living room.

Landmark 9:

Frontier Landmark: True

Surrounding objects: a vase which next to the wall.

Room: living room.

Landmark 11:

Frontier Landmark: False

Surrounding objects: a cabinet which next to a vase, next to a wall.

Room: living room.

Landmark 19:

Frontier Landmark: False

Surrounding objects: a lamp which is next to a window.

Room: living room.

Landmark 28:

Frontier Landmark: False

Surrounding objects: a chair which is next to a chair and next to a desk.

Room: living room.

Landmark 37:

Frontier Landmark: False

Surrounding objects: a vase which is under a flower, next to a bed and next to a wall.

Room: bedroom.

Landmark 46:

Frontier Landmark: False

Surrounding objects: a couch which is next to a chair, next to a lamp under the desk.

Room: living room.

Landmark 55:

Frontier Landmark: False

Surrounding objects: a picture which is hanged on the wall, under the light.

Room: hallway.

Cognitive Map Prompting for State Transition

```
{  
  "Result": Transition to state 'Contextual Search'.  
}
```

Fig 4. **State Transition Prompting of CogNav and LLM Results.** We provide state transition prompting and state transition by LLM in an episode finding a sofa. The result by LLM is transited to state 'Contextual Search'.

Cognitive Map Prompting for Landmark Selection

Goal: Sofa

State: Contextual Search

Agent Now: Landmark 11

The Landmarks candidate:

Landmark 7:

Frontier Landmark: False

Location—Direction to 11: Up-right; Path from 11 to 7: 11 ->2 ->3 ->10 ->1; Distance from 11 to 7: 27.03

Surrounding objects: a desk which next to a chair, under a flower and next to the wall .

Room: living room. Explored: False

Landmark 9:

Frontier Landmark: True

Location—Direction to 11: Up-right; Path from 11 to 9: 11 ->2 ->10 ->9; Distance from 11 to 9: 15.16

Surrounding objects: a vase which next to the wall.

Room: living room. Explored: False

Landmark 19:

Frontier Landmark: False

Location—Direction to 11: Down-left;

Surrounding objects: a lamp which is next to a window.

Room: living room. Explored: False

Landmark 28:

Frontier Landmark: False

Location—Direction to 11: Up-left; Path from 11 to 28: 11 ->2 ->33 ->16 ->17 ->28; Distance from 11 to 28: 48.06

Surrounding objects: a chair which is next to a chair and next to a desk.

Room: living room. Explored: False

Landmark 46:

Frontier Landmark: False

Location—Direction to 11: Down-left

Surrounding objects: a couch which is next to a chair, next to a lamp under the desk.

Room: living room. Explored: False

LLM Reasoning of the Next Landmark

```
{  
  "Result": Landmark 46,  
  "Prompting Reason": "Landmark 46 is chosen due to descriptions indicating the presence of  
    a couch, a potential match to the target object—sofa. The proximity of the couch strengthens  
    the likelihood of a nearby sofa. Furthermore, Landmark 46 lies within the living room and is  
    unexplored, increasing the chances of discovering the target object in a relevant setting.  
    This Landmark, despite lacking the 'Frontier' status, serves well in alignment with the search  
    goal due to the promising object association and location context."  
}
```

Fig 5. **Cognitive Map Prompting of CogNav and LLM Results.** We provide landmark selection prompting and landmark selection by LLM in an episode finding a sofa. The current state is contextual search, we prompt the landmarks $\mathcal{L}_{t, relevant}$ as candidates for next landmark selection.

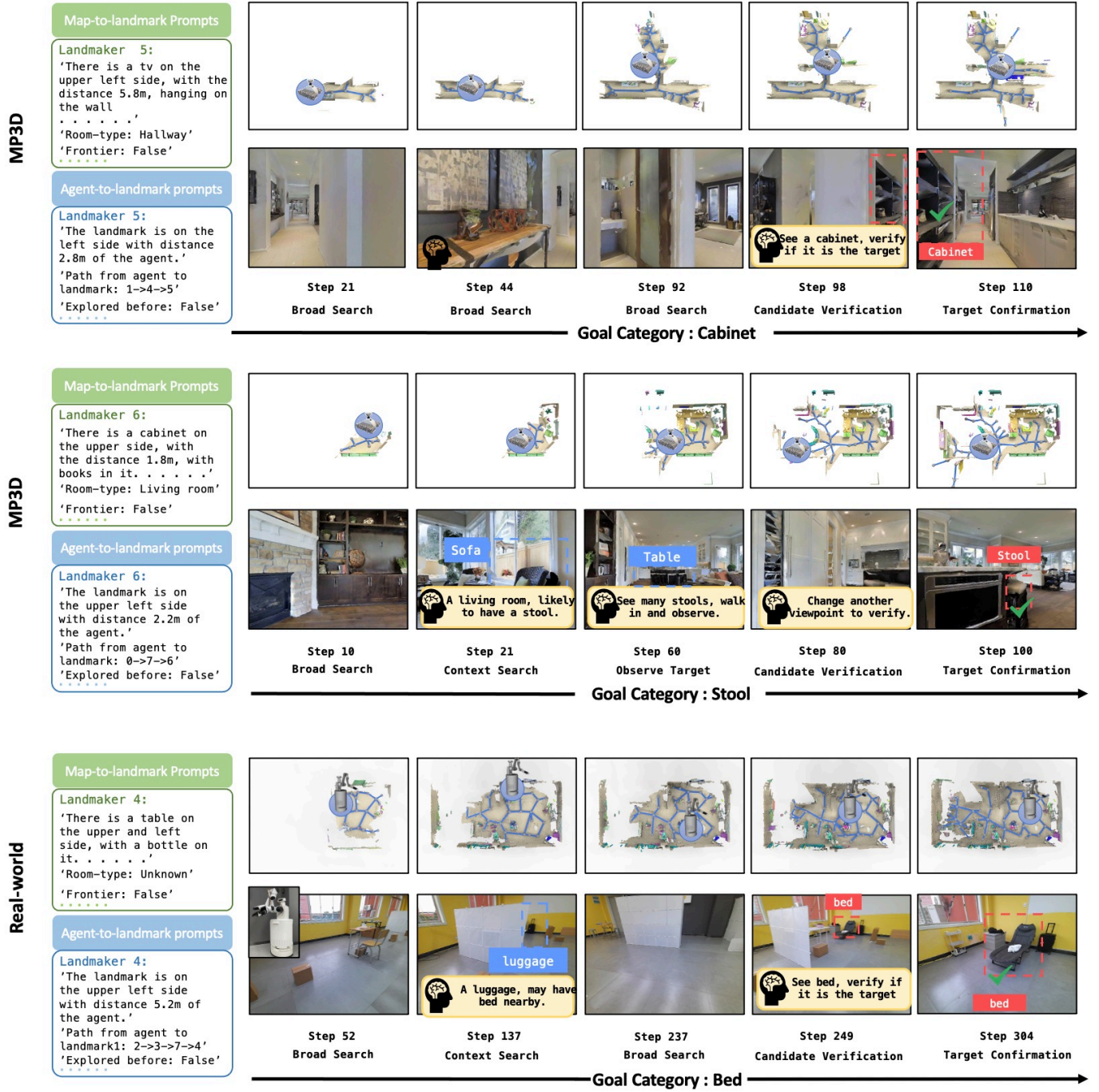
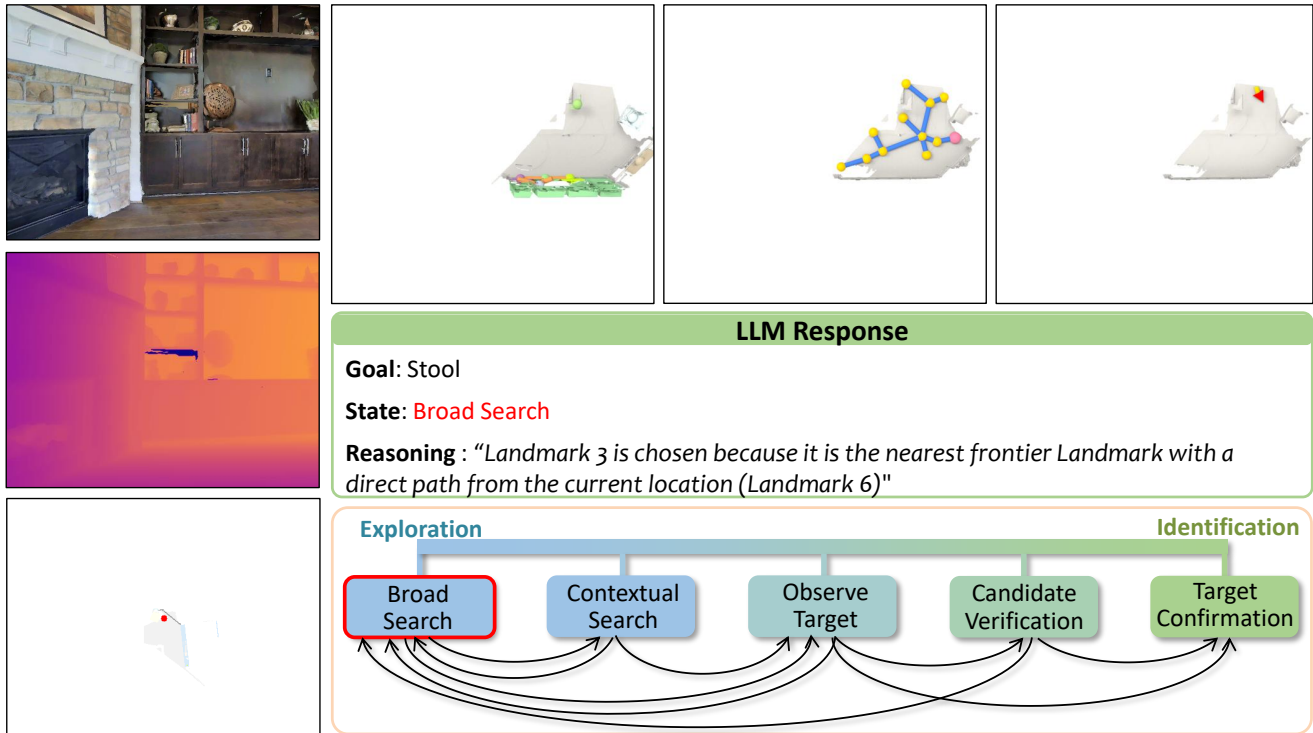
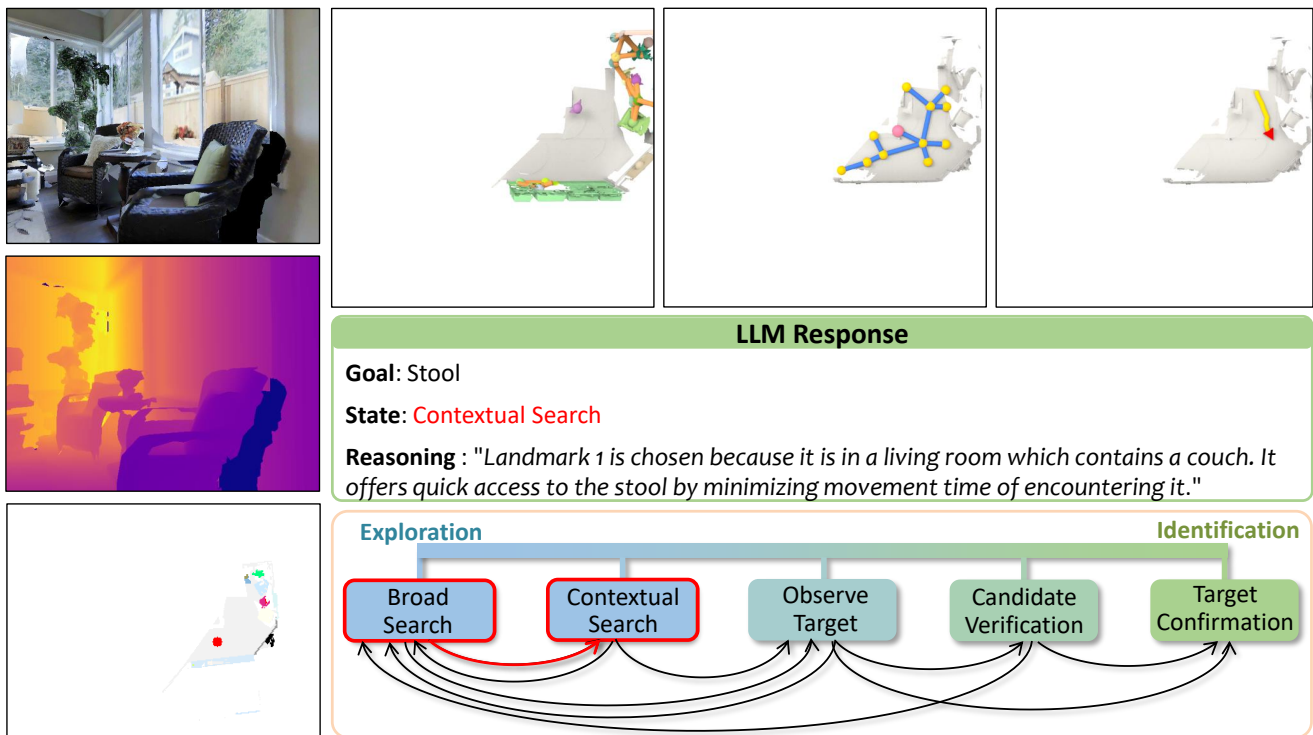


Fig 6. **Navigation process visualization of CogNav.** We provide visual results of navigation process for one synthetic scene and one real-world one. Cognitive maps encode scene information and facilitate landmark prompting, enabling the agent to explore environments efficiently and identify target objects accurately.

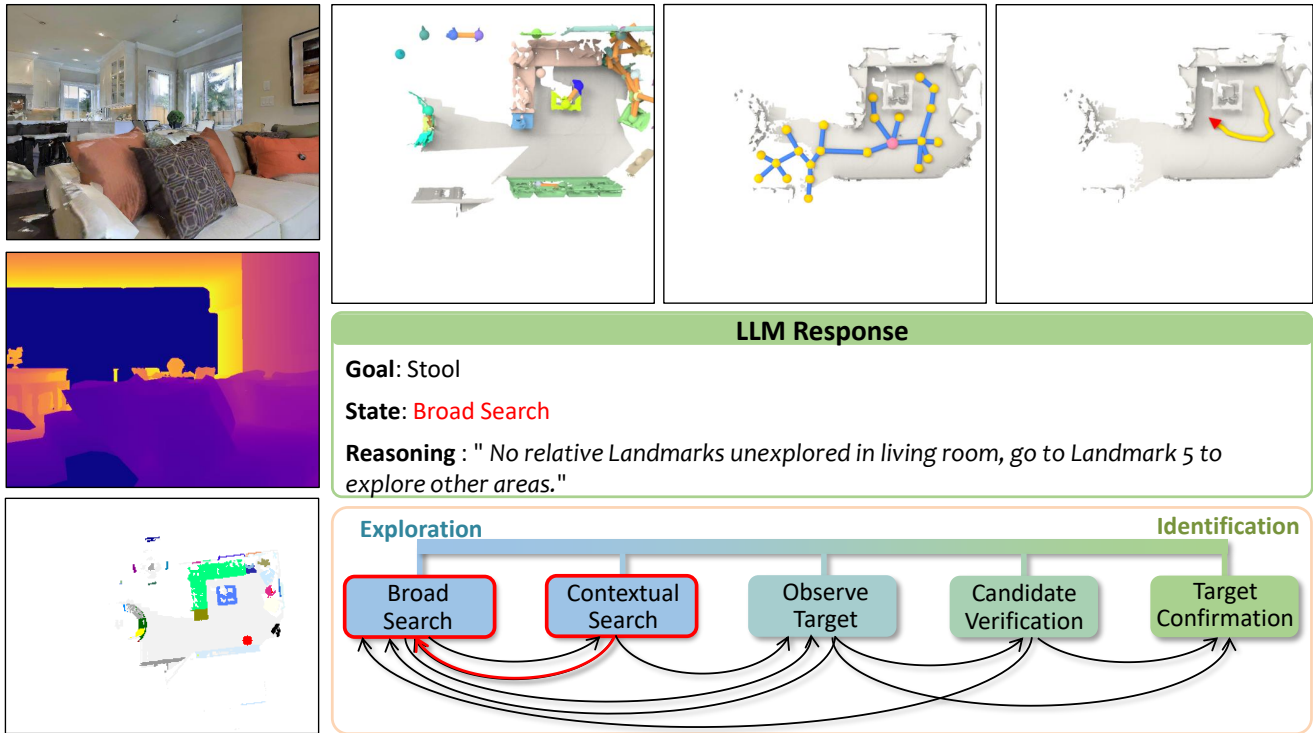


1

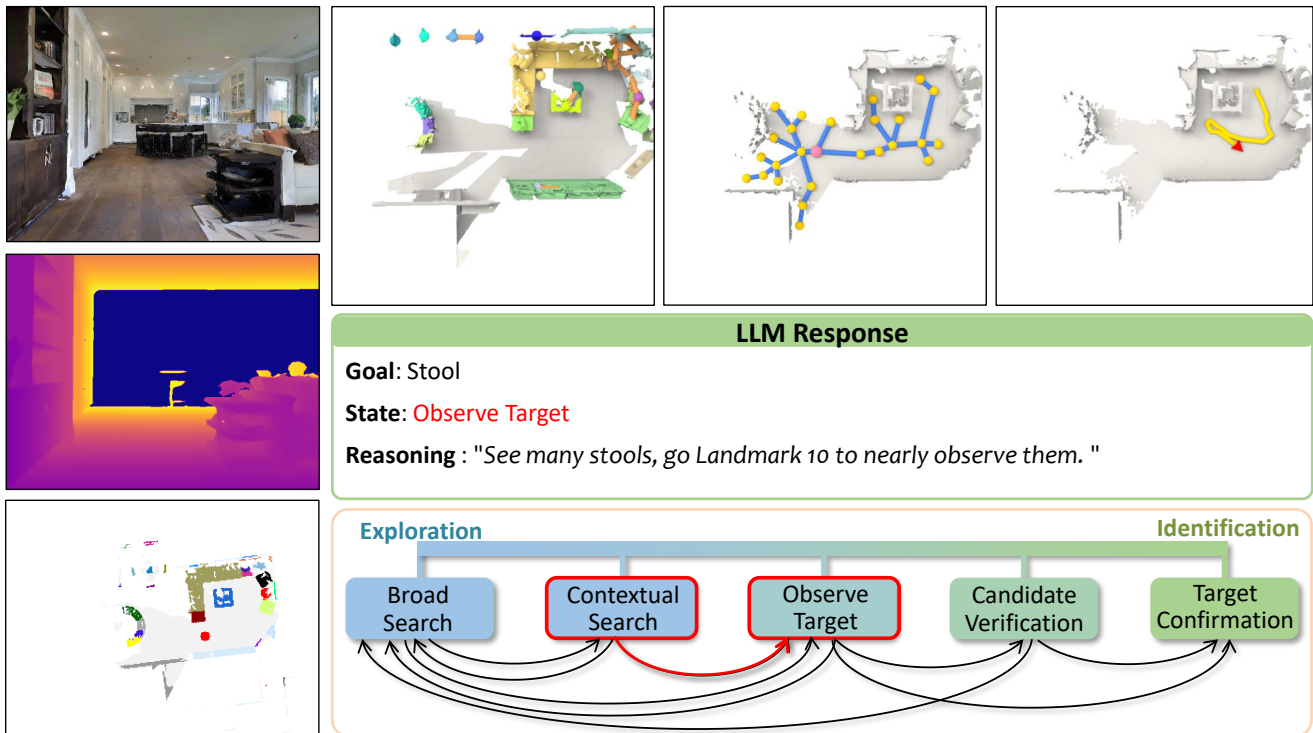


2

Fig 7. State transition in navigation process of finding a stool.

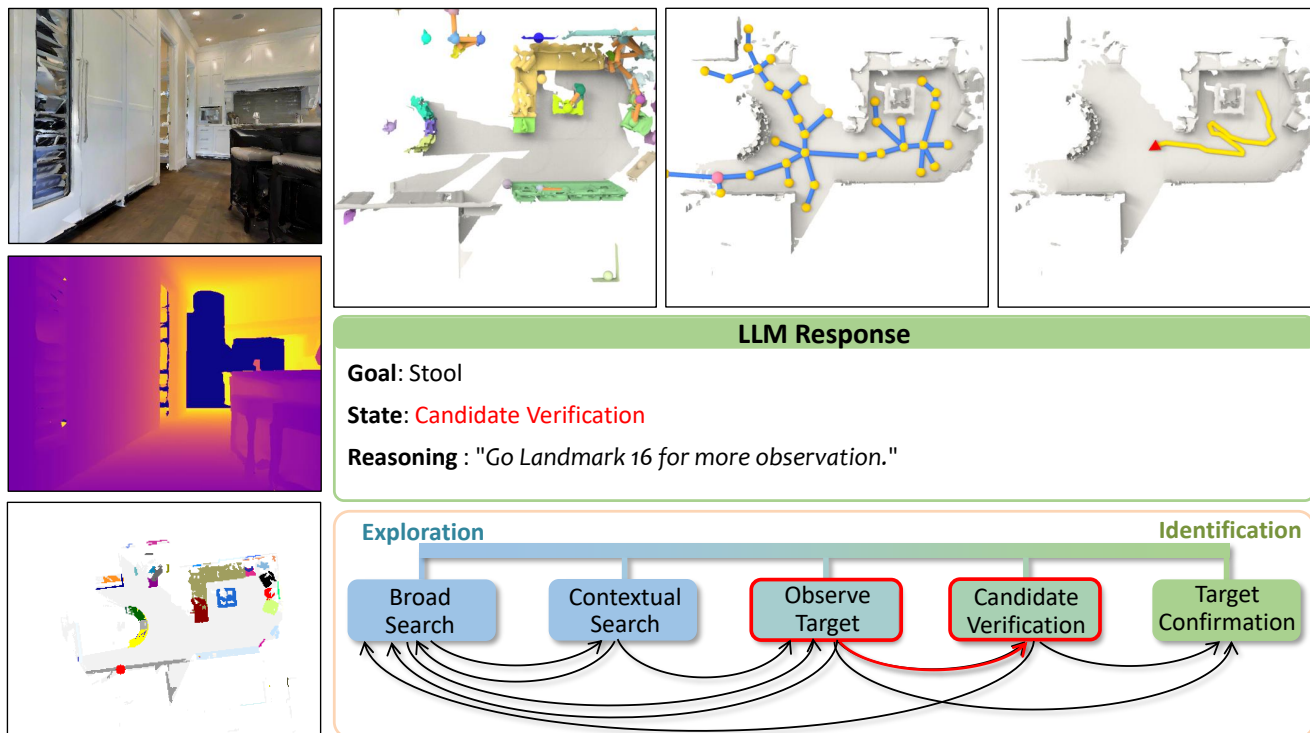


3

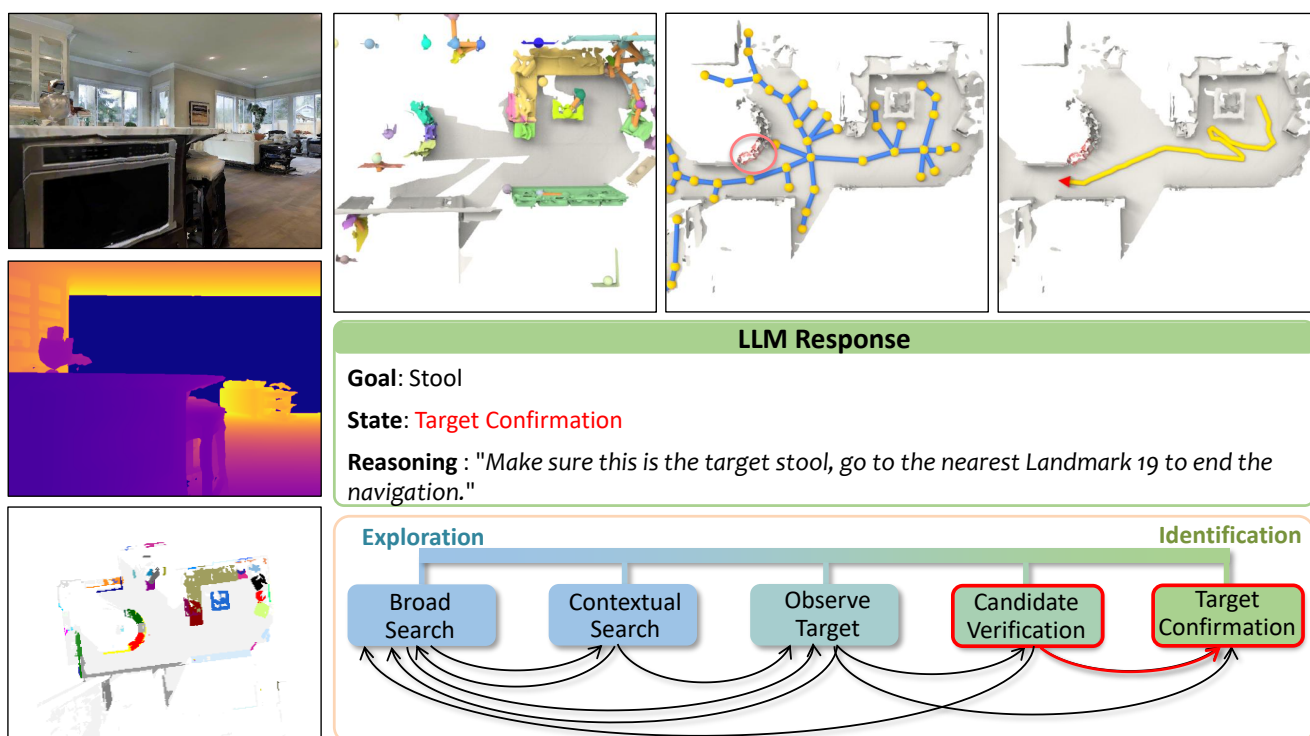


4

Fig 8. State transition in navigation process of finding a stool.



5



6

Fig 9. State transition in navigation process of finding a stool.

References

- [1] Elfin collaborative robot e05. <https://www.hansrobot.net/elfin-collaborative-robot>. Accessed: 2024-11-21.
- [2] Microsoft azure kinect dk. <https://learn.microsoft.com/en-us/azure/kinect-dk/>. Accessed: 2024-11-21.
- [3] Water ii. <https://www.yunji.ai/en/autonomous-robot-platform-chassis.html>. Accessed: 2024-11-21.
- [4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [5] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.
- [7] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.
- [8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [9] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23171–23181, 2023.
- [10] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.
- [11] Wenyi Hong, Wei Han Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024.
- [12] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.
- [13] Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. *arXiv preprint arXiv:2401.02695*, 2024.
- [14] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.
- [15] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *arXiv preprint arXiv:2410.08189*, 2024.
- [16] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3554–3560. IEEE, 2023.
- [17] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1020–1031, 2023.
- [18] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, pages 42829–42842. PMLR, 2023.