Mem2Ego: Empowering Vision-Language Models with Global-to-Ego Memory for Long-Horizon Embodied Navigation

Lingfeng Zhang Yuecheng Liu Zhanguang Zhang Matin Aghaei Yaochen Hu
Hongjian Gu Mohammad Ali Alomrani David Gamaliel Arcos Bravo Raika Karimi
Atia Hamidizadeh Haoping Xu Guowei Huang Zhanpeng Zhang Tongtong Cao
Weichao Qiu Xingyue Quan Jianye Hao Yuzheng Zhuang
Yingxue Zhang
Huawei Noah's Ark Lab
{lingfeng.zhang1, zhanguang.zhang, yingxue.zhang}@huawei.com

Abstract

Recent advancements in Large Language Models (LLMs) and Vision-Language Models (VLMs) have made them powerful tools in embodied navigation, enabling agents to leverage commonsense and spatial reasoning for efficient exploration in unfamiliar environments. Existing LLM-based approaches convert global memory, such as semantic or topological maps, into language descriptions to guide navigation. While this improves efficiency and reduces redundant exploration, the loss of geometric information in language-based representations hinders spatial reasoning, especially in intricate environments. To address this, VLM-based approaches directly process ego-centric visual inputs to select optimal directions for exploration. However, relying solely on a first-person perspective makes navigation a partially observed decision-making problem, leading to suboptimal decisions in complex environments. In this paper, we present a novel vision-language model (VLM)-based navigation framework that addresses these challenges by adaptively retrieving task-relevant cues from a global memory module and integrating them with the agent's egocentric observations. By dynamically aligning global contextual information with local perception, our approach enhances spatial reasoning and decision-making in long-horizon tasks. Experimental results demonstrate that the proposed method surpasses previous state-of-the-art approaches in object navigation tasks, providing a more effective and scalable solution for embodied navigation.

1. Introduction

Embodied navigation is a crucial component of embodied artificial intelligence, with widespread applications in diverse scenarios such as domestic environments, office settings, logistics and delivery, and factory inspections [4, 22, 34]. Its significance stems from its ability to enable agents to autonomously navigate and perform tasks within physical environments [13, 17].

Embodied navigation poses two key challenges. First, unlike autonomous driving, which typically occurs in structured outdoor environments, embodied navigation requires operating in diverse indoor and industrial settings such as factories, shopping malls, and offices. These spaces feature intricate layouts and obstacles, demanding advanced perception and planning [4, 16, 20, 26]. Second, it necessitates a high degree of autonomy, as agents must adapt to unfamiliar environments without relying on predefined maps. They must interpret human instructions and dynamically interact with their surroundings to navigate effectively. This work focuses on Object Goal Navigation (ObjectNav), a task in which agents must locate specified objects within complex spaces [2, 15].

In recent years, the rapid advancement of large language models (LLMs) has opened new possibilities for embodied navigation [29, 32]. These models enable robots to leverage commonsense reasoning, improving their understanding of natural language commands and enhancing the integration of perceptual data. This allows for navigation decisions that better align with human intentions [10, 28]. Furthermore, recent ObjectNav research underscores the importance of historical information in improving environmental understanding, decision-making, and grounding navigation instructions [5, 27]. This has led to the incorporation of memory systems into LLMs, such as episodic memory for past experiences and scene graph memory for structuring environmental data. However, because these memory systems often represent memories using natural language, which lacks geometric information, the spatial reasoning capacity of LLMs is compromised.

Alongside these advancements, there is growing interest in using images as a primary source of guidance by integrating foundation models with low-level planners [3, 18]. This approach takes advantage of the advanced visual and language understanding of foundation models, offering an effective alternative to traditional map-based methods, which often rely on costly and disruption-prone depth sensing and localization. However, these methods predominantly rely on a first-person perspective without incorporating global memory into the decision-making process. As a result, they tend to lead to redundant exploration and reduced efficiency in complex environments.

In this paper, we propose a novel Vision-Language Model (VLM)-based navigation framework that addresses these challenges by adaptively retrieving task-relevant cues from a global memory module, which are then mapped to the agent's ego-view visual observations. By integrating global contextual information with local perceptual inputs, our framework enables more informed action decisions, enhancing the agent's situational awareness and decision-making capabilities. The approach significantly enhances the agent's ability to navigate complex, long-horizon tasks by dynamically aligning global context with egocentric reasoning, offering a more effective and scalable solution for embodied navigation. The experimental results demonstrate that our proposed navigation pipeline outperforms state-of-the-art baselines. Through an ablation study, we verified the essential nature of each component of our method. Using our proposed data collection approach, the supervised fine-tuned Llama3.2-11B model exhibited superior performance compared to both the vanilla Llama3.2-11B model and GPT-4o.

2. Related Work

Existing studies that leverage VLMs and LLMs for navigation can be categorized into the following directions.

2.1. LLM-based Navigation

These approaches often construct a global memory map based on image observations and use natural language to describe candidate points for navigation, with action decisions driven by large language models (LLMs).

Several methods fall within this category, including **LFG** [23], **VoroNav** [28], **ESC** [35], and **openFMNav** [9]. LFG uses frontier-based exploration and large language models to score potential subgoals and guide navigation based on the robot's observations and exploration progress. VoroNav introduces Reduced Voronoi Graphs (RVGs) to optimize the robot's exploration by identifying intersections that provide the best observational opportunities, while the LLM predicts the next best waypoint. ESC uses commonsense knowledge and frontier-based exploration to navigate toward objects in the environment, while openFMNav addresses challenges related to human instructions that imply target objects and zero-shot generalization. These methods employ LLMs to dynamically update a semantic map as the robot explores, enhancing memory and reducing redundant exploration.

While these methods offer the advantage of maintaining a global map and using high-level reasoning, they also face limitations. The language-based reasoning used for decision-making sacrifices high-dimensional semantic information, such as spatial and geometric details, which can constrain performance in complex environments. Furthermore, translating raw ego-view observations into abstract linguistic descriptions may weaken the model's capacity for precise spatial reasoning.

2.2. Value Map-based Navigation

In this class of methods, a global value function is computed based on ego-view observations, and actions are chosen based on the generated value map instead of using VLMs for decision-making.

Notable approaches in this category include **VLFM** [31] and **InstructNav** [14]. VLFM uses a pre-trained vision-language model to generate a language-grounded value map, guiding the agent to explore optimal frontiers. InstructNav extends the idea of goal-directed navigation by introducing a Dynamic Chain of Navigation that breaks down tasks into sequences of actions and landmarks. These methods partially address memory forgetting by integrating global value maps, but they still face challenges. The value map is still constructed based on local observations, and decision-making driven by vision-language models (VLMs) often lacks a comprehensive global perspective. As a result, these approaches frequently lead to suboptimal solutions constrained by local decision-making.

2.3. VLM-based Navigation

These approaches directly leverage first-person perspective images as the input of vision-language models (VLMs) to generate action decisions. By using the spatial reasoning capabilities of VLMs, these methods enable the model to interpret complex environmental features from the robot's current viewpoint, facilitating more informed and context-aware navigation decisions.

CoNVOI [21] and **PIVOT** [18] exemplify approaches that process first-person images with VLMs to facilitate real-time navigation and decision-making. While effective in leveraging immediate visual inputs, these methods lack mechanisms for incorporating historical observations, often resulting in redundant exploration. This limitation poses challenges in long-horizon tasks, where maintaining contextual awareness of past actions is critical for efficient navigation. **VLMNav** [6] addresses some of these limitations by integrating both RGB-D images and the robot's pose in-

formation to construct a navigability mask that identifies reachable regions. The model incrementally builds a voxel-based map and refines its action proposals by prioritizing unexplored areas.

NoMaD [24] unifies goal-directed navigation and exploration by using the robot's current image and the goal's image as input. The model includes a transformer backbone for processing visual data and a diffusion model for predicting action sequences. A binary mask is applied to the input to focus on either exploration (excluding the goal) or goal-reaching (including the goal). Despite its innovative design, NoMaD remains constrained by the absence of a global memory, relying solely on the most recent three observations. This limitation restricts its capacity for sustained long-term exploration.

Recent methods have sought to integrate VLMs more effectively for embodied navigation. **OpenIN** [25] focuses on navigation tasks where the robot must locate specific objects that have been moved, introducing a Carrier-Relationship Scene Graph (CRSG) to track objects and their locations. The system uses VLMs to process multimodal instructions and commonsense knowledge to guide navigation decisions.

Uni-NaVid [33] takes a significant step toward unifying different navigation tasks in a single model. It processes both video streams and natural language instructions as input, creating a framework that can generalize across a range of navigation tasks. By training on diverse data, including video question answering and captioning tasks, Uni-NaVid improves its performance in real-world scenarios and enables asynchronous execution for efficiency.

These methods move toward integrating both global and local information more effectively, enabling the robot to navigate complex environments with a better understanding of spatial context. However, challenges remain in optimizing the trade-off between VLMs' generalization capabilities and the need for precise, real-time navigation.

3. Method

3.1. Problem Formulation

In this work, we focus on the object navigation (ObjNav) task, where an agent begins at a random location within an unseen environment and is tasked with finding and navigating to a target object, denoted by g. The agent has no access to a pre-built map and must rely entirely on its sensory inputs for navigation. At each time step t, the agent captures an egocentric RGB-D image, denoted by o^t, from its onboard RGB-D camera. Additionally, the agent has access to its current location and orientation, which are represented by the extrinsic matrix M_{ext} of the camera. Using these inputs, the agent must compute and execute a low-level action, a^t, that efficiently guides it toward the target object.

The workflow of our proposed method is illustrated in Figure 1. The VLM-based navigation relies on the integration of a memory module that encompasses three distinct types of memories. The construction and maintenance of this memory module, as well as the VLM-based navigation process, will be discussed in detail in the subsequent sections.

3.2. Memory Construction

The memory module is composed of three distinct types of memories, each serving a different purpose:

- **Frontier Map**: Denoted as M_f, frontier map has been proven to be effective for environment exploration in object navigation tasks, as demonstrated in Shah et al. [23], Zhou et al. [35]. We adopt an approach similar to that used in ESC [35] to construct the frontier map. Using the agent's position and camera parameters, RGB-D images are transformed into 3D space, where each 2D pixel is mapped to a 3D voxel in the global coordinate system. Voxels located near the floor, with no obstacles along the height dimension, are classified as free space. A frontier in this map is defined as the boundary between free and unexplored areas. This frontier map is maintained throughout the navigation task.
- **Landmark Semantic Memory**: Denoted as M_l, this memory stores descriptions of the landmarks that the agent has seen in the past. Each entry includes the global coordinates of the landmark and a description of the nearby semantic information, such as objects or decoration texture. For example: "*[13.2, 5.4]: Located on the floor near a sink. There is a bath tub nearby.*". The description of each landmark is generated by the VLMs, as explained in Section 3.3.
- **Visitation Memory**: Denoted as M_v, this memory keeps track of the landmarks that the agent has already visited. By maintaining a record of visited locations, the visitation memory serves as a crucial mechanism to prevent redundant exploration and improve overall exploration efficiency.

3.3. Mem2Ego Navigation

At each time step t, given the image-based observation o^t and the three types of memories—M_f^t, M_l^t, and M_v^t—introduced in section 3.2, the proposed memory-to-egocentric (Mem2Ego) navigation process can be formulated as follows:

$$a^t = f_\theta(o^t, M_f^t, M_l^t, M_v^t, g) \tag{1}$$

Further details are provided in the following sections.

3.3.1. Panoramic Observation Generation

After the environment is initialized or the agent reaches a new location, the agent captures four egocentric RGB-D

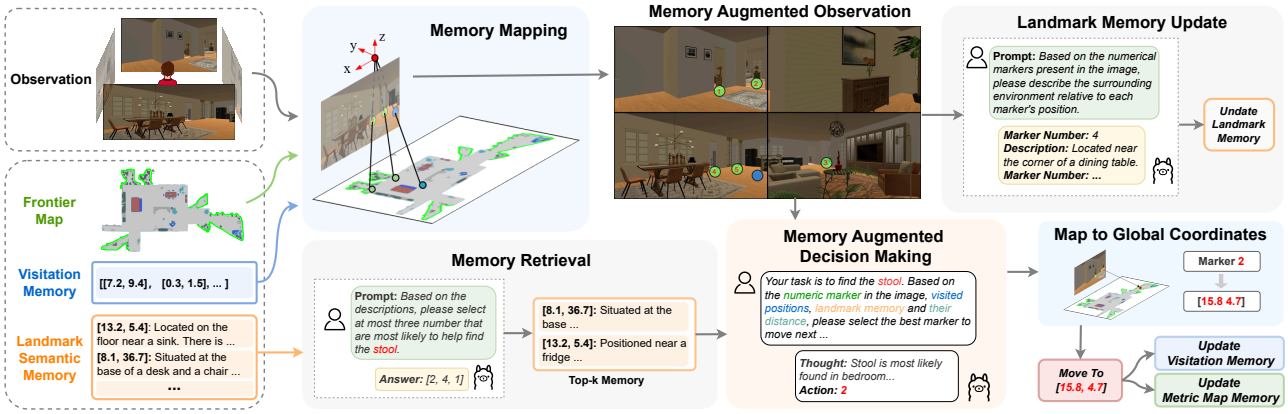

Figure 1. Workflow of our proposed method. Our method maintains three types of memories and project cues from them onto the egocentric images for goal location prediction. Further details are provided in Section 3

images by rotating its viewpoint 90 degrees at each step. These images are then stacked to construct a 360-degree panoramic observation o^t_{pano} (see Equation 2), offering a comprehensive representation of the surrounding environment. Compared to navigation methods relying on a single egocentric view, this panoramic approach enhances the agent's spatial awareness and scene understanding. A similar strategy has been employed in Long et al. [14].

$$o^t_{\text{pano}} = \text{Concatenate}([o^t_0, o^t_{\pi/2}, o^t_{\pi}, o^t_{3\pi/2}]) \tag{2}$$

3.3.2. Frontier and Visitation Memory Projection

Based on the agent's position and the newly captured depth images, the navigation map and corresponding frontiers are updated following the method outlined in Section 3.2. Candidate locations, denoted as $[\mathbf{C}_1, ..., \mathbf{C}_N]$ in Equation 3, are generated by combining frontier clustering and grid-based sampling. The centroid of each frontier segment is computed by clustering all points within the segment. However, using the centroid directly as a candidate may result in unreachable goal positions. To mitigate this, we identify the nearest grid point on the floor area to the centroid, ensuring that the candidate is accessible to the agent. Similarly, visited locations, $[\mathbf{V}_1, ..., \mathbf{V}_M]$, are extracted from the visitation memory M^t_v, as shown in Equation 4.

$$[\mathbf{C}_1, ..., \mathbf{C}_N] = \text{CandidatesGeneration}(M^t_f) \tag{3}$$

$$[\mathbf{V}_1, ..., \mathbf{V}_M] = \text{VisitationExtraction}(M^t_v) \tag{4}$$

Once determined, the global coordinates of these candidates and visitations are projected onto the egocentric image plane as pixel locations $[\mathbf{c}_1, ..., \mathbf{c}_N]$ and $[\mathbf{v}_1, ..., \mathbf{v}_M]$, as shown in Equation 5, where K and M_{ext} represent the camera intrinsics and extrinsics, respectively.

$$\begin{gathered}\mathbf{c}_i = \text{Projection}(\mathbf{C}_i), \quad \mathbf{v}_i = \text{Projection}(\mathbf{V}_i) \\ \text{where } \mathbf{c}_i = (x_i, y_i),\ \mathbf{C}_i = (X_i, Y_i, Z_i),\ \text{similar for } \mathbf{v}_i \text{ and } \mathbf{V}_i \\ [x'_i, y'_i, w_i]^T = K \cdot M_{\text{ext}} \cdot [X_i, Y_i, Z_i, 1]^T \\ (x_i, y_i) = (\frac{x'_i}{w_i}, \frac{y'_i}{w_i})\end{gathered} \tag{5}$$

An annotation function is then applied to map these locations onto the panoramic observation o^t_{pano}, resulting in an annotated observation o^t_{anno}, as outlined in Equation 6. In the annotated image, candidate locations are depicted as green circles, each labeled with a unique identifier corresponding to its position in the image. Similarly, visited locations are marked as blue circles, but only if they are visible within the current view.

$$o^t_{\text{anno}} = \text{AnnotateImage}(o^t_{\text{pano}}, [\mathbf{c}_1, ..., \mathbf{c}_N], [\mathbf{v}_1, ..., \mathbf{v}_M]) \tag{6}$$

3.3.3. Landmark Memory Retrieval

The panoramic image, augmented with frontier candidates, highlights potential navigation targets within the agent's immediate field of view. However, it is common that no suitable targets are visible, and more promising options may exist among the landmarks the agent has previously encountered but not yet explored. These previously encountered landmarks are stored in the dynamic landmark semantic memory M^t_l. To manage the rapid expansion of this memory during navigation, we utilize large language models (LLMs) to retrieve the top-k landmarks most relevant to the target object. This retrieval process generates an additional observation from memory, o^t_{mem}, which is then incorporated into the decision-making process. The prompt used for memory retrieval is detailed in Appendix 5.1.

$$o^t_{\text{mem}} = \text{MemoryRetrieval}_{\text{LLMs}}(M^t_l, k) \quad (7)$$

3.3.4. Memory Augmented Decision Making

At this stage, the panoramic image with annotations, o^t_{anno}, along with the top-k landmarks retrieved from memory, o^t_{mem}, is used to query VLMs to select the next target location to visit (described in Equation 8). The VLMs are tasked with identifying the marker on the image most likely to lead to the target object, while avoiding markers that are too close to previously visited locations. To ensure consistency in the output format, the top-k landmarks are numbered, and their descriptions are considered only if no suitable marker is identified directly from the panoramic image. A Chain-of-Thought (CoT) prompting strategy is employed to guide the VLMs in generating a structured reasoning process before producing a single numerical output corresponding to the selected marker. The full prompt used for decision-making is provided in Appendix 5.1.

$$a^t = f_{\text{VLMs}}\left(\text{prompt}(g), o^t_{\text{anno}}, o^t_{\text{mem}}\right) \quad (8)$$

3.3.5. Action Execution

The marker selected in step 3.3.4 is transformed to the global coordinate system to determine the global coordinates of the target location. Shortest path follower provided by habitat simulator is then executed to navigate agent to the target location while avoiding obstacles. Object detection is performed each time the agent moves or adjusts its viewing angle. The task is deemed successful if the target object is detected within the agent's field of view and the agent successfully navigates to the target object's viewpoints provided by the Habitat dataset. If the target object is not detected, the process continues until the agent either reaches the designated viewpoints or exceeds the maximum allowed number of exploration steps.

3.3.6. Memory Update

While only one landmark from the current view is selected as the next-step navigation target, other landmarks may still be valuable for future exploration. The landmark semantic memory is updated before target position navigation described in Section 3.3.5. VLMs are prompted to describe the surrounding environment near each marker annotated on the panoramic image. The output from the VLMs includes a list of marker IDs paired with corresponding descriptions. The marker IDs are then converted to global coordinates and, together with their descriptions, saved to the landmark semantic memory for use in future exploration processes. The prompt used for landmark description is provided in Appendix 5.1.

Meanwhile, the navigation map is updated along the navigation process, using the RGB-D images captured along the way. Additionally, the agent's most recent location is added to the visitation memory to facilitate future exploration.

3.4. Data Collection and Model Finetuning

To enhance the capabilities of open-sourced VLMs and narrow their performance gap with GPT-4o, we design a pipeline to collect training data for supervised finetuning (SFT). The data collection pipeline is illustrated in Figure 2. To improve data diversity and validate the generalization ability of the model, we gather 40 new categories of objects from the HSSD dataset, rather than using the original 6 categories provided. First, new target objects are sampled from the HSSD scenes. For each frame of data, ground-truth trajectories from the current position to these targets are calculated based on the A^* algorithm and subsequently smoothed using Bézier curves. Egocentric images and the corresponding ground-truth target pixel (x, y) (defined as the endpoint of the ground-truth trajectory shown in the image) for each image are saved. To construct the multiple marker annotated image that VLMs encounter in the marker selection task, we generate a few candidate landmarks for each image by sampling from the edge of the floor area. Both ground-truth and sampled candidate landmarks are annotated on the egocentric image in the same way as in the Section 3.

We collect two types of data for VLM fine-tuning: **marker description** and **target marker selection with rationale**. To generate marker description data, we use GPT-4o to describe the surrounding environment of each marker on the image. For example, "*Marker Number: 1 Description: Positioned near a dining chair...; Marker Number: 2 ...*". Each target marker selection data entry includes both a rationale and the ID of the selected marker. To ensure a robust rationale, we utilizes egocentric images annotated with the ground-truth trajectory and employ a dual-phase prompting strategy: first, GPT-4o is prompted to describe all the objects along the ground-truth trajectory, then to predict the location of the target marker based on its relationship to these objects. Importantly, the rationale generated by GPT-4o must not reference the ground-truth trajectory itself; the trajectory is only used to guide the generation of the rationale. The generated rationale is then automatically validated using GPT-4o, assessing both the accuracy of detected objects and the correctness of the rationale. This dual-phase prompting strategy has proven to be more reliable than a single-phase prompting approach. The prompts used for rationale generation are provided in Appendix 5.2. The validated rationale is then concatenated with the ground-truth marker ID to enforce a CoT-like thinking process. An example of the resulting response is "*Think: The candle is most likely located on the shelf on the right side ... Action: 2*". Note that the resulting marker selection data used for model fine-tuning relies on images annotated with numerical markers, rather than those anno-

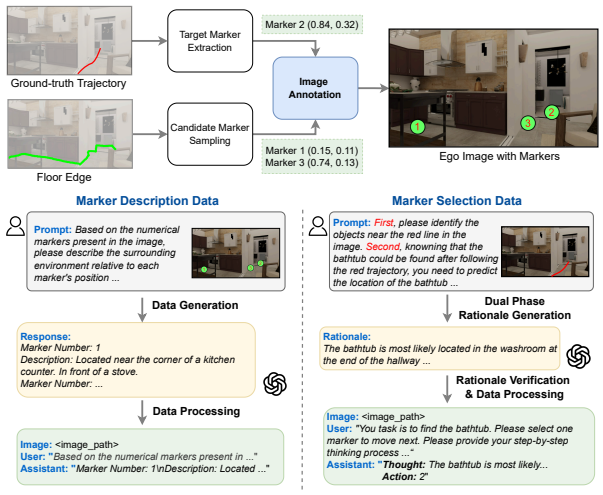

Figure 2. Pipeline of SFT data collection. The ground-truth trajectory and floor edge are used to extract target marker and candidate markers, respectively. Marker description and selection data is generated for model fine-tuning.

tated with the ground-truth trajectory. In total, we generated 30,352 visual question answer (VQA) pairs of data from 104 scenes and 5678 object navigation tasks. This data was used to fine-tune a Llama3.2-11B-Vision model [7] following the configuration recommended by official Llama repository. The model was fine-tuned for 3 epochs with a learning rate of 1e-5 and an effective batch size of 128.

4. Experiments

4.1. Experimental Setup

We evaluated our method on the navigation tasks using the Habitat 3.0 [19] simulation platform. We adopt similar setup as the Habitat ObjectNav 2022 challenge [30] for all the experiments. The action space of the agent consists of: `STOP`, `MOVE_FORWARD`, `TURN_LEFT`, `TURN_RIGHT`, with a forward movement distance of 0.25 meters and a turning angle of 30 degrees per step. For low-level movement control, we utilized Habitat's built-in shortest-path follower. The maximum number of steps allowed per task is set to 500 by default. Due to limitations in the image quality within the Habitat environment and the suboptimal performance of state-of-the-art perception modules, such as GroundingDINO [12], we opted for Habitat's built-in semantic ground truth with object size conditions as the perception module. In this context, we can assume that the perception module is sufficiently effective. The LLMs and VLMs used in this study was GPT-4o and Llama3.2-11B.

4.2. Datasets

Our method is evaluated on the following two object navigation datasets:

- **Habitat Synthetic Scenes Dataset (HSSD) [8]:** We use the HSSD validation dataset to evaluate our method. HSSD consists of 41 scenes and six object goal categories: `chair`, `couch`, `potted plant`, `bed`, `toilet`, and `tv`. To ensure task diversity, we select only one episode per scene-object pair. After filtering out erroneous episodes—such as cases where the agent's initial position was in mid-air—the final number of evaluated episodes is 213.
- **HSSD-Hard:** Since some HSSD episodes are relatively easy, with the agent finding the target object in just a few steps, we created a more challenging dataset, HSSD-Hard, by selecting HSSD episodes with longer search distance. We calculated the geodesic distance from the agent's starting point to the target object for each episode and selected the top 50% of episodes with the longest searching distances to form the HSSD-Hard dataset. The total number of episodes in HSSD-Hard is 102.

4.3. Baselines

We compare our method against the following state-of-the-art (SOTA) baselines that represent different strategies to address the object navigation problem:

- **PIVOT [18]:** This approach casts the navigation task as an iterative visual question answering problem by annotating the image with numerical markers that represent the navigation subgoals. The method is adapted for the HSSD object navigation tasks. Without a frontier map, visitation memory, and landmark semantic memory, our proposed navigation pipeline degenerates to PIVOT.
- **LFG [23]:** This method employs frontier-based exploration and LLMs to score potential subgoals and guide the navigation.
- **VLFM [31]:** The approach utilizes VLMs to generate a language-grounded value map, from which the location with the highest value is selected as the next subgoal for navigation.
- **InstructNav [14]:** InstructNav introduces a Dynamic Chain of Navigation, breaking down navigation tasks into sequences of actions and landmarks. It employs four value maps, each with different semantic representations, to assist in selecting the appropriate landmark.
- **VLMNav [6]:** This approach relies on a voxel map built from RGB-D images and the agent's pose to narrow down action space.

To ensure a fair comparison, all experimental setups are conducted under the same conditions described earlier.

Table 1. Main results. Our proposed method is compared to the baselines on HSSD and HSSD-Hard datasets. SR: Success Rate. SPL: Success Weighted by Path Length. All experiments are conducted using gpt-4o.

	HSSD		HSSD-Hard	
	SR ↑	SPL ↑	SR ↑	SPL ↑
LFG	0.6244	0.3371	0.6176	0.3454
VLMNav	0.6526	0.3620	0.5294	0.1973
InstructNav	0.7605	0.3722	0.6372	0.4187
VLFM	0.7652	0.5574	0.6078	0.4270
PIVOT	0.7840	0.5658	0.6372	0.4744
Ours	**0.8685**	**0.5788**	**0.7647**	**0.4790**

4.4. Metrics

We employ the following metrics to evaluate the performance of all the methods:

- **Success Rate (SR):** A task was deemed successful when the distance between the agent and any viewpoint of the target object was less than 0.2 meters.
- **Success Weighted by Path Length (SPL) [1]:** This metric evaluates how efficient the agent's path is compared to the optimal path. SPL is calculated as:

$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{max(p_i, l_i)} \tag{9}$$

where l_i is the length of the optimal path for episode i. p_i is the length of path taken by the agent. S_i is the binary indicator of success in episode i.

4.5. Main Results

We evaluate our method and all baselines on both HSSD and HSSD-Hard datasets using the same setup described in Section 4.1, with results summarized in Table 1. Performance is evaluated based on Success Rate (SR) and Success weighted by Path Length (SPL). While SR indicates the overall ability to find the target object, SPL measures the efficiency of the navigation process. Notably, these two metrics are not correlated, as a method can achieve a high SR by sacrificing navigation efficiency. As shown in Table 1, on the HSSD dataset, our proposed method achieves an SR of 0.8685 and an SPL of 0.5788, both of which are higher than all the baseline methods.

Compared to HSSD, tasks in the HSSD-Hard dataset are more challenging due to the relatively longer search distance, requiring additional steps to locate target objects. As shown in Figure 1, the performance of all methods decreases on the HSSD-Hard dataset, though the impact varies by model. Notably, our method demonstrate an even greater advantages in these more difficult scenarios, achieving an SR that is 12.75% higher than the second-best baseline (PIVOT). Additonally, our method outperforms others in SPL as well, further highlighting its efficiency. These results underscore the effectiveness and robustness of our approach in tackling challenging navigation tasks.

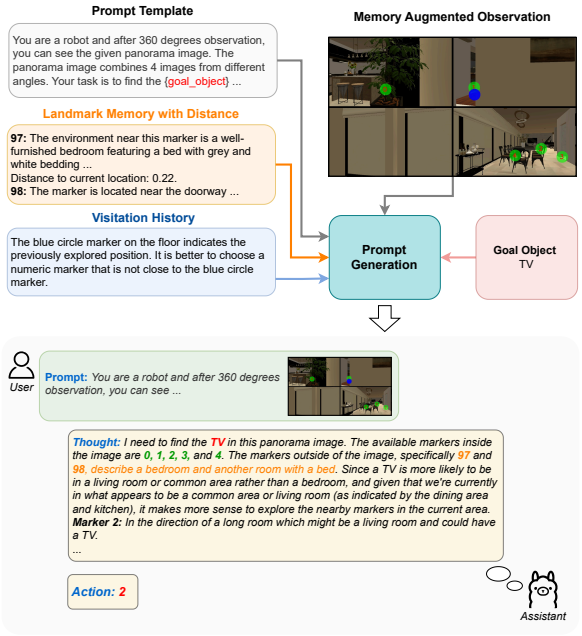

Figure 3. Demonstration of memory-augmented decision making process from a real HSSD episode. The full prompts and GPT-4o response are provided in Appendix 5.3

Figure 3 illustrates the memory-augmented decision-making process from a real HSSD episode. VLMs, such as GPT-4o, analyze all memory cues on the image before reasoning about the most likely location of the target object and selecting the next marker to explore. The full prompt and responses for this case are provided in Appendix 5.3.

Most failed cases with our method result from reaching the maximum allowable number of steps. This can occur due to the VLM selecting a suboptimal position or the task being inherently challenging. Additionally, we occasionally observe that even state-of-the-art VLMs like GPT-4o can exhibit visual hallucinations [11], where they select a marker ID that does not appear in the image or prompt. We have provided two examples in the Appendix 5.4.

4.6. VLM Model Supervised Fine-tuning (SFT)

To evaluate the impact of the VLM used, we assess the performance of various VLMs within our proposed method. As shown in Table 2, the vanilla Llama3.2 model performs significantly worse than GPT-4o. Given the substantial differences in model size and training data, it is not surprising that smaller open-source VLMs like Llama3.2-11B un-

Table 2. Results with different VLM models. The fine-tuned Llama3.2-11B model (SFT Llama3.2-11B) trained on our collected data outperforms GPT-4o, achieving the best overall performance.

	HSSD		HSSD-Hard	
	SR ↑	SPL ↑	SR ↑	SPL ↑
GPT-4o	0.8685	0.5788	0.7647	0.4790
Vanilla Llama	0.7511	0.5582	0.7352	0.4626
SFT Llama	**0.8732**	**0.5995**	**0.7843**	**0.5274**

derperform compared to state-of-the-art proprietary models such as GPT-4o. Failure analysis reveals that Llama3.2-11B is more prone to visual hallucinations and struggles with instruction following, particularly in marker selection and description tasks. This may stem from a lack of relevant training data for Llama3.2-11B, limiting its ability to generalize effectively in these scenarios.

To improve the performance of VLMs, we collected over 30,000 VQA samples by generating data from 40 new object categories within the HSSD dataset. Rationales generated using a dual-phase prompting strategy are used to guide the marker selection process. We then fine-tuned Llama3.2-11B following the approach detailed in Section 3.4. As shown in Table 2, the performance of Llama3.2 is improved significantly after supervised fine-tuning (SFT), surpassing GPT-4o in both SR and SPL metrics on the HSSD and HSSD-Hard datasets. These results are particularly promising, considering that Llama3.2-11B is substantially smaller than GPT-4o (11B vs. an estimated 175B) and more cost-effective to train and deploy. This highlights the effectiveness of our data collection strategy and fine-tuning approach. This performance improvement can be attributed to enhanced instruction adherence and a more effective reasoning process grounded in the given environment.

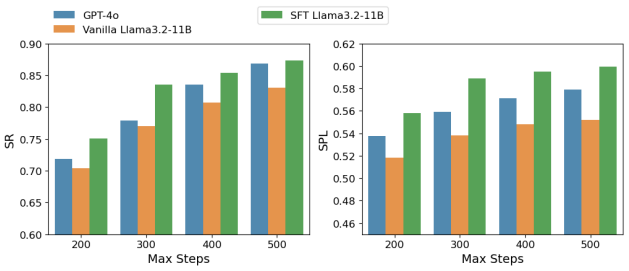

Figure 4. Results for different maximum steps. Experiments are conducted on the HSSD dataset. While SFT Llama3.2-11B consistently achieves the best performance, its advantage over the other two models is most pronounced at 300 maximum steps.

Numerous factors influence the performance of the navigation pipeline, particularly the maximum number of allowed steps. Consequently, we assessed the impact of this factor on performance. As shown in Figure 4, with the maximum number of steps increasing, both the Success Rate (SR) and Success weighted by Path Length (SPL) improve for each model. When the maximum number of steps is relatively low, such as 200 or fewer, all methods exhibit suboptimal performance, with only minor differences between models. This is likely because the limited step count prevents the full exploitation of each model's capabilities. Conversely, when the maximum number of steps is large, such as 500 or more, all models reach their respective performance limits, making the differences between them less distinct. It is noteworthy that the performance gap between different models becomes more apparent when the maximum number of steps is set to 300 or 400.

4.7. Ablation Study

To assess the contribution of each component in our proposed method, we conducted an ablation study by evaluating its performance without landmark semantic memory or visitation memory. As shown in Table 3, removing either of these memory modules leads to a decline in performance. The frontier map could not be excluded from this analysis, as it is essential for marker generation. The absence of visitation memory results in redundant exploration in some cases, thus reducing both the success rate and navigation efficiency. Meanwhile, without landmark semantic memory, the agent is unable to select a navigation goal globally when no suitable marker is present in its current view, which harms the performannce as well. These findings highlight the crucial role of both memory modules in fully leveraging the potential of our proposed method.

Table 3. Ablation study. Removing any of the main components of our design leads to degraded performance on HSSD dataset.

	HSSD		HSSD-Hard	
	SR ↑	SPL ↑	SR ↑	SPL ↑
Ours	**0.8685**	**0.5788**	**0.7647**	0.4790
w/o Visitation	0.8450	0.5761	0.7450	**0.4961**
w/o Landmark Semantic	0.8356	0.5669	0.7352	0.4795

5. Conclusion

This study proposes an efficient fusion strategy that integrates task-relevant global memory information with first-person perspective information, thereby overcoming the suboptimal solution problem associated with existing multimodal navigation frameworks due to local observability. Moreover, this method can simultaneously activate and

utilize the complex spatial understanding, reasoning, and commonsense reasoning capabilities of VLMs, thus significantly enhancing the ability and efficiency of navigation decisions in complex spatial scenarios. Theoretically, enhanced spatial cognitive abilities can reduce the required travel distance and number of actions, thereby increasing the task completion success rate and overall efficiency of the navigation scheme.

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 7

[2] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 1

[3] Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5228–5234. IEEE, 2024. 2

[4] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022. 1

[5] Rui Fukushima, Kei Ota, Asako Kanezaki, Yoko Sasaki, and Yusuke Yoshiyasu. Object memory transformer for object goal navigation. In *2022 International conference on robotics and automation (ICRA)*, pages 11288–11294. IEEE, 2022. 1

[6] Dylan Goetting, Himanshu Gaurav Singh, and Antonio Loquercio. End-to-end navigation with vision language models: Transforming spatial reasoning into question-answering. *arXiv preprint arXiv:2411.05755*, 2024. 2, 6

[7] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and et al. The llama 3 herd of models, 2024. 6

[8] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2024. 6

[9] Yuxuan Kuang, Hai Lin, and Meng Jiang. Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models. *arXiv preprint arXiv:2402.10670*, 2024. 2

[10] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *arXiv preprint arXiv:2410.07166*, 2024. 1

[11] Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*, 2024. 7

[12] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2025. 6

[13] Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *arXiv preprint arXiv:2407.06886*, 2024. 1

[14] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024. 2, 4, 6

[15] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022. 1

[16] Christoforos Mavrogiannis, Patrícia Alves-Oliveira, Wil Thomason, and Ross A Knepper. Social momentum: Design and evaluation of a framework for socially competent robot navigation. *ACM Transactions on Human-Robot Interaction (THRI)*, 11(2):1–37, 2022. 1

[17] Ronja Möller, Antonino Furnari, Sebastiano Battiato, Aki Härmä, and Giovanni Maria Farinella. A survey on human-aware robot navigation. *Robotics and Autonomous Systems*, 145:103837, 2021. 1

[18] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv preprint arXiv:2402.07872*, 2024. 2, 6

[19] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023. 6

[20] Pericle Salvini, Diego Paez-Granados, and Aude Billard. Safety concerns emerging from robots navigating in crowded pedestrian areas. *International Journal of Social Robotics*, 14(2):441–462, 2022. 1

[21] Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, Mohamed Elnoor, Anuj Zore, Brian Ichter, Fei Xia, Jie Tan, Wenhao Yu, and Dinesh Manocha. Convoi: Context-aware navigation using vision language models in outdoor and indoor environments. *arXiv preprint arXiv:2403.15637*, 2024. 2

[22] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of*

the IEEE/CVF international conference on computer vision, pages 9339–9347, 2019. 1

[23] Dhruv Shah, Michael Robert Equi, Błażej Osiński, Fei Xia, Brian Ichter, and Sergey Levine. Navigation with large language models: Semantic guesswork as a heuristic for planning. In *Conference on Robot Learning*, pages 2683–2699. PMLR, 2023. 2, 3, 6

[24] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024. 3

[25] Yujie Tang, Meiling Wang, Yinan Deng, Zibo Zheng, Jingchuan Deng, and Yufeng Yue. Openin: Open-vocabulary instance-oriented navigation in dynamic domestic environments. *arXiv preprint arXiv:2501.04279*, 2025. 3

[26] Kristinn Thórisson and Helgi Helgasson. Cognitive architectures and autonomy: A comparative review. *Journal of Artificial General Intelligence*, 3(2):1–30, 2012. 1

[27] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15625–15636, 2023. 1

[28] 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. 1, 2

[29] Yuchen Wu, Pengcheng Zhang, Meiying Gu, Jin Zheng, and Xiao Bai. Embodied navigation with multi-modal information: A survey from tasks to methodology. *Information Fusion*, page 102532, 2024. 1

[30] Karmesh Yadav, Santhosh Kumar Ramakrishnan, John Turner, Aaron Gokaslan, Oleksandr Maksymets, Rishabh Jain, Ram Ramrakhya, Angel X Chang, Alexander Clegg, Manolis Savva, Eric Undersander, Devendra Singh Chaplot, and Dhruv Batra. Habitat challenge 2022. https://aihabitat.org/challenge/2022/, 2022. 6

[31] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48. IEEE, 2024. 2, 6

[32] Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S Yu. Large language models for robotics: A survey. *arXiv preprint arXiv:2311.07226*, 2023. 1

[33] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024. 3

[34] Tianyao Zhang, Xiaoguang Hu, Jin Xiao, and Guofeng Zhang. A survey of visual navigation: From geometry to embodied ai. *Engineering Applications of Artificial Intelligence*, 114:105036, 2022. 1

[35] 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. 2, 3

Mem2Ego: Empowering Vision-Language Models with Global-to-Ego Memory for Long-Horizon Embodied Navigation

Supplementary Material

5.1. Prompts for inference

Prompt 1

```
You are an automated system with
the capability to analyze the
provided image. Based on the
numerical markers present in the
image, please describe the
surrounding environment relative
to each marker's position. Ensure
that descriptions of different
markers are distinct to maintain
the uniqueness of each marker.
The marker number should not
appear in the description. Please
adhere to the following format:
Marker Number: [insert the number
of the first marker here]
Description: [provide a
description corresponding to the
first marker here]
Marker Number: [insert the number
of the second marker here]
Description: [provide a
description corresponding to the
second marker here]
...
Marker Number: [insert the number
of the last marker here]
Description: [provide a
description corresponding to the
last marker here]
```

Prompt 2

```
Based on the provided descriptions
for each number, please select at
most three number whose
corresponding descriptions are
most likely to help identify the
{goal_object}. {Number:
Description; Number: Description;
... Number: Description} If the
total number is less than 3,
please use -1 to occupy the empty
position. Please adhere to the
following format for the output:
Number List: [first number,
second number, third number]
```

Prompt 3

```
You are a robot and after 360
degrees observation, you can see
the given panorama image. The
panorama image combines 4 images
from different angles. Your task
is to find the {goal_object}.
Based on the numerical markers in
the image, select one of these
numbers to move next. If you're
not confident in moving to the
marker to find the {goal_object},
you can choose one of the
numerical markers located outside
of this image. The descriptions
of these markers are as follows:
{the top 3 numbers with
descriptions, corresponding to the
response of Prompt 2}.
If you're still not confident in
moving to the marker to find the
{goal_object}, your action should
be 'None'. The blue circle marker
on the floor indicates the
previously explored position. It
is better to choose a numeric
marker that is not close to the
blue circle marker. Please note
all closed doors cannot be opened.
Please follow the format like
this,
Thought: [put your step-by-step
thinking process here]
Action: [put a single marker id
or None here]
```

5.2. Prompts for SFT data generation

Prompt for dual-phase rational generation

You are given an image with a red movement trajectory on it. Please first identify the objects near the red line in the given image. If there is no red trajectory in the image, please directly return "None". Second, knowning that {goal_object} could be found after following the red trajectory, you need to predict the location of goal_object or the region where {goal_object} could be most likely located. This can be achieved by reasonably imagining the unseen areas after the red trajectory based on the room layout. **Do not mention the red trajectory/line or "the image" in your output!** Please structure your output in the following way:
OBJECTS_RED_LINE:
LOCATION_PREDICTION_AND_REASONING:

Prompt for rationale filtering

You are given an image with a movement trajectory marked in a red line. Please first verify if all of the objects in a given list are present near the red line in the given image. If there is no red line in the image or any of the objects not present, please ignore the rest and directly return "NONE". Second, verify if the reasonings of why {goal_object} may be put close to the objects in the list. A good reasoning should be logical and perfectly reflect common sense knowledge. A good reasoning gives convincing reasons while a bad reasoning gives vague or untruthful reasons. If the reasonings are good, output "GOOD REASONINGS", otherwise, output "BAD REASONINGS". Example of a good reasoning: "The book is most likely located on the shelves in the background. The shelves are a common place for storing books, and they are visible in the room at the end of the path.". Example of a bad reasoning: "The mirror is most likely located on the dark wall to the right of the doorway. This is inferred from the visible portion of the mirror reflecting the room, indicating its position on the dark wall.". Please structure your output in the following way:
OBJECTS_PRESENCE_CHECK:
REASONING_CHECK:
Object list: object_list
Reasonings: reasonings

5.3. Case Study

Prompt and response for case study

Prompt: You are a robot and after 360 degrees observation, you can see the given panorama image. The panorama image combines 4 images from different angles. Your task is to find the tv. Based on the numerical markers in the image, select one of these numbers to move next. If you're not confident in moving to the marker to find the tv, you can choose one of the numerical markers located outside of this image. The descriptions of these markers are as follows: 97: The environment near this marker is a well-furnished bedroom featuring a bed with grey and white bedding. There is a grey headboard attached to the wall, from which two contemporary pendant lights hang. Next to the bedside is a white nightstand with drawers. In the background, large windows or glass doors bring light into the room. To the far right, through the doorway, part of an adjacent room is visible where a small trolley or shelf holding white objects can be seen. 98: The marker is located near the doorway of a

room. To the left of the marker, there is a green potted plant. Through the doorway, the room is furnished with a bed positioned near the wall with large windows that extend almost to the floor, allowing for an expansive view of the outside. Additionally, there is a chair or possibly a sofa situated to the left inside the room. The floor is covered with a dark blue surface adding a touch of color to the environment. If you're still not confident in moving to the marker to find the tv, your action should be 'None'. The blue circle marker on the floor indicates the previously explored position. It is better to choose a numeric marker that is not close to the blue circle marker. Please note all closed doors cannot be opened. Please follow the format like this, Thought: [put your step-by-step thinking process here] Action: [put a single marker id or None here]
Response: Thought: I need to find the TV in this panorama image. The available markers inside the image are 0, 1, 2, 3, and 4. The markers outside of the image, specifically 97 and 98, describe a bedroom and another room with a bed. Since a TV is more likely to be in a living room or common area rather than a bedroom, and given that we're currently in what appears to be a common area or living room (as indicated by the dining area and kitchen), it makes more sense to explore the nearby markers in the current area. Marker 0: Leads to a kitchen area, not likely to have a TV. Marker 1: Near a closed door with a blue circle, so less likely. Marker 2: In the direction of a long room which might be a living room and could have a TV. Marker 3: Looks like another part of the same long room, potentially a living area. Marker 4: Seems to be in the dining area, also less likely to have a TV. Action: 2

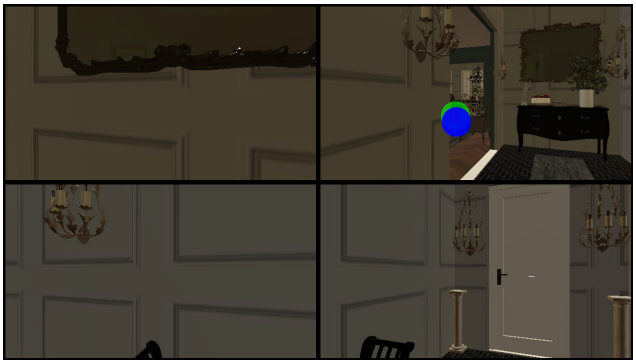

Figure 5. Visual Hallucination Example 1

Figure 6. Visual Hallucination Example 2

5.4. Visual Hallucinations

Visual Hallucination Example 1

Thought: 1. The top-left, bottom-left, and bottom-right sections of the panorama show closed doors and walls. 2. The top-right section shows an open doorway leading to another room that might be more likely to have a toilet. 3. The numeric marker "2" is located near the open doorway in the top-right section. Action: 2

Visual Hallucination Example 2

Thought: The given panorama shows parts of a room and possibly adjacent areas. The upper left image shows a workspace with a computer, the upper right shows a hallway and another room, the lower left has a part of a bed, and the lower right includes windows with curtains. There is no indication of a couch in these views. The only promising lead is the room visible through the doorway marked with a green number "1" in the upper right image, which might show a larger area including a couch. Action: 1