OAM: Object-Aware Memory and Vision-Language Models for Zero-Shot Object Navigation

Jiahui Wang[1], Wen Liu[1,*] and Zhongliang Deng[1]

[1]*School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Abstract

Object-goal navigation is a key challenge in the field of robotics, which requires robots to navigate and locate a target object in unknown environments. Previous work usually relies on the semantic analysis of single-frame observations, which is prone to semantic understanding bias and instability problems. In this paper, we propose OAM, a novel zero-shot object navigation framework that builds an object-centric spatiotemporal memory. OAM breaks through the limitations of single-frame observations by introducing a spatiotemporal memory buffer module.This module integrates visual, depth, and pose information, enabling the agent to intelligently recall and reason about their historical observation information. In addition, an object-aware semantic focusing mechanism is designed to accurately extract object-related information associated with frontier cells from memory. OAM incorporates this mechanism into the visual language model to identify the most promising frontiers to explore as targets. We evaluated OAM in photorealistic environments from the Gibson and Habitat-Matterport 3D (HM3D) datasets within the Habitat simulator. The results show that our method outperforms existing methods in terms of both SR and SPL.

Keywords

Object Goal Navigation, Memory Buffer, Object Extraction, Zero-shot Object Navigation

1. Introduction

Object-target navigation (ObjectNav) [1] requires the agent to actively explore and locate a specified target object in an unknown environment based on a given instruction (e.g., "bed"). This task has broad applications in real-world scenarios, such as disaster rescue and domestic service. In this task, the semantic understanding of the environment [2] by the agent becomes a key factor affecting its navigation decisions.

With the development of large-scale simulation datasets[3] and reinforcement learning(RL), RL-based methods have been widely applied to address the ObjectNav task. These approaches, including end-to-end[4, 5] and modular[2, 6], aim to implicitly learn semantic relationships from large amounts of data. However, they heavily rely on task-specific training and exhibit limited generalization. Recently, advances in large-language models (LLMs) and vision-language models (VLMs) have introduced new solutions for ObjectNav, notably zero-shot object navigation (ZSON). ZSON eliminates the need for task-specific training by leveraging common sense knowledge encoded in pre-trained models to understand the relationship between the goal and the environment, guiding the agent's navigation decisions. However, most of the existing ZSON methods rely on single-frame observations for semantic reasoning. This makes them vulnerable to issues such as viewpoint bias, poor lighting, occlusion, and background clutter, leading to agent semantic understanding errors and affecting the navigation effect.

Inspired by how humans search for objects in unfamiliar environments, we identify a key mechanism: short-term memory and spatial reasoning. When uncertain about their surroundings, humans tend to recall recently observed information (for example, 'I remember seeing an alarm clock and a bed in that corner') and integrate visual cues from different points of view to make informed decisions. If the agent could simulate this cognitive ability, as shown in Fig. 1, it could combine current and historical observations to select the most promising exploration path most likely to lead to the target object.

IPIN-WCAL 2025: Workshop for Computing & Advanced Localization at the Fifteenth International Conference on Indoor Positioning and Indoor Navigation, September 15–18, 2025, Tampere, Finland

[*] Corresponding author.

jiahui@bupt.edu.cn (J. Wang); liuwen@bupt.edu.cn (W. Liu); dengzhl@bupt.edu.cn (Z. Deng)

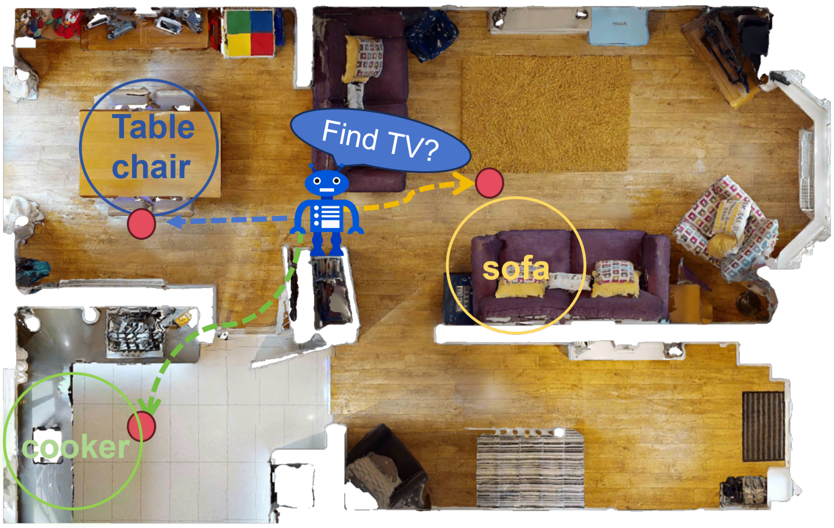

Figure 1: How can an agent find a specified object (e.g., a TV) in an unknown environment? When facing multiple frontier points, the agent needs to select the most suitable exploration target based on the surrounding object information at each frontier.

In this work, we propose OAM, a novel navigation framework built around object-centric spatiotemporal memory. By introducing a module that simulates human-like short-term memory and attention, OAM overcomes the limitations of single-frame observations and instead integrates visual, depth, and pose information from past experiences into a coherent spatiotemporal representation. During navigation, the agent retrieves this memory in the frontier cells, uses an object detector to extract object patches from historical frames, and employs VLM to identify the frontier most likely to lead to the target object. This retrieve–focus–decide mechanism enables more robust and efficient ZSON. The main contributions of this paper are as follows.

- We propose OAM, which overcomes the limitations of single-frame observations by retrospectively retrieving historical observations through short-term memory and utilizing VLM for semantic matching.
- We introduce an object-level semantic focusing mechanism to achieve efficient matching of frontier points with target semantics, thus improving the quality of decision-making in complex environments.

2. Related Work

Object-goal Navigation: The ObjectNav task requires an agent to navigate to a specified object in an unknown environment based on instructions. Traditional methods are divided into end-to-end approaches and map-based modular approaches. End-to-end methods [4] map observations directly to navigation actions using reinforcement learning or imitation learning. These methods typically require extensive training and have poor interpretability. Subsequently, a significant body of research has focused on map-based modular approaches, which consist of modules for map mapping, policy learning, and path planning. The classical modular approach SemExp [2] is to construct an explicit semantic map and combine it with objective-directed exploration strategies. FSE [7] constructs both semantic and frontier maps, using deep learning strategies to select long-term goals for exploration. These methods reduce computational cost and improve interpretability to some extent, but still rely on large-scale training data. The emergence of zero-shot object navigation (ZSON) methods addresses this issue and has gained widespread attention. These methods guide the agent to make navigation decisions without the need for additional training. Recent research has introduced VLMs [8, 9] and LLMs [10]

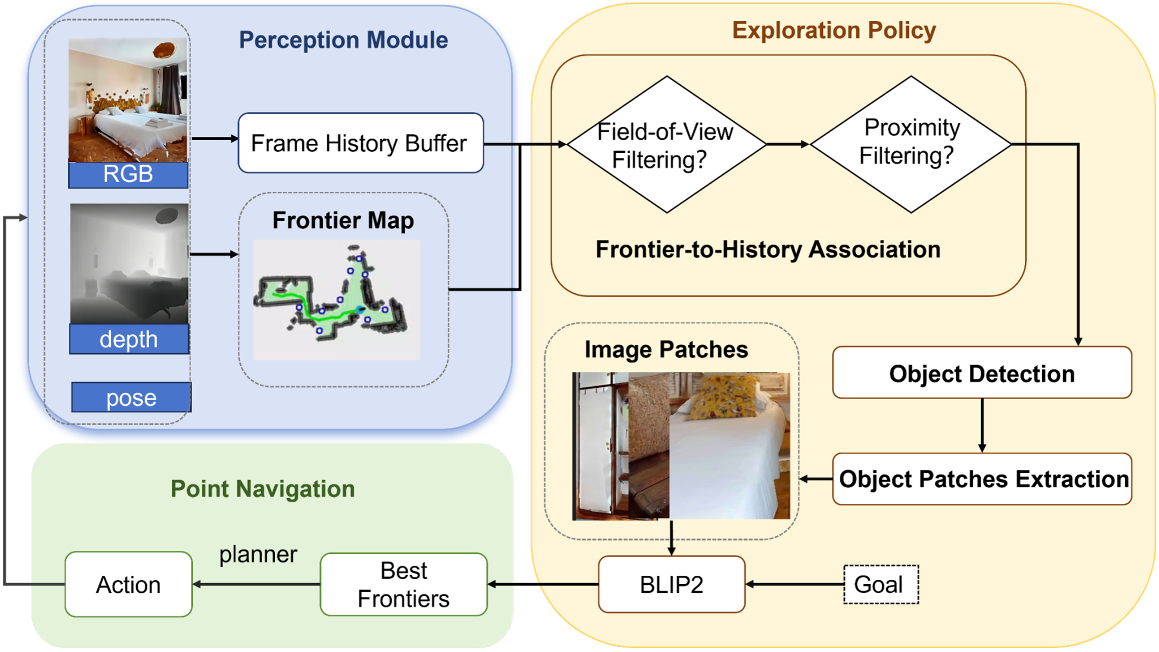

Figure 2: Overview of the proposed OAM navigation framework. A frontier map is first constructed through the perception module. Then, a frontier exploration strategy selects a target frontier point, followed by path planning and action generation, enabling the agent to interact effectively with the indoor environment.

into ObjectNav, using the semantic knowledge embedded in these models as prior knowledge, thereby allowing zero-shot object navigation without task-specific training.

Frontier Exploration Strategies in ZSON: Recent work has introduced frontier-based[11] exploration methods to zero-shot navigation tasks. CLIP on Wheels (CoW) [12] uses FBE to explore the boundaries between free and unknown spaces, employing a simple heuristic. ESC [13] combines LLM with GLIP[9] for object and room recognition, inferring the boundaries that are the most likely to lead to the target object. L3VMN [14] uses textual descriptions of the boundaries of the semantic map, and VLFM [15] directly inputs single-frame images into VLM[8], outputting semantic scores from visual and textual cues. These methods focus on frontier exploration as a heuristic, but we improve upon this by aggregating spatiotemporal semantic information from frontier points as input to the VLM. By leveraging multi-frame object-level observations, we refine semantic similarity scores and more accurately utilize the contextual information around the frontier points, resulting in enhanced navigation performance.

3. Method

3.1. Pipeline

As illustrated in Fig. 2, OAM consists of three main modules.First, the perception module takes RGB images, depth maps, and pose information as input to build an occupancy grid map, extract frontier points, and construct a frontier map(3.2). It also stores the agent's observation sequence as historical memory(3.3).The exploration module filters historical observations for each frontier point based on field of view coverage and spatial distance, then uses an object detector to extract object patches from images. These are fed into a VLM to compute similarity scores, which reflect the semantic relevance of each frontier(3.4). After the final selection of the most valuable frontier point the path is planned and navigated in the navigation module to update the observation inputs(3.5).

3.2. Frontier Map Construction

At each timestep t, the agent receives an RGB image I_t, a depth map D_t, and the current pose p_t. By projecting the depth data into the global coordinate system, we obtain the local 3D structure of the environment. This information is then integrated into a global 2D occupancy grid map $M_{\text{occ}} \in \{0, 1, -1\}^{H \times W}$, where 0 denotes free space, 1 indicates obstacles, and -1 represents unknown regions. We define frontier points as free-space cells that are adjacent to at least one unknown cell, representing the boundary between explored and unexplored areas. To construct the frontier map, we first identify all free cells in M_{occ} and mark those with at least one unknown neighbor as frontier candidates. To reduce redundancy, we apply spatial clustering to group neighboring frontier points into clusters, from which representative points are extracted to form a discrete frontier set:

$$F = \{f_1, f_2, \dots, f_N\}, \quad f_i \in \mathbb{R}^2. \tag{1}$$

3.3. Spatiotemporal Memory Buffer

When facing a potential target location, humans often recall objects they previously saw in that area—an ability resembling short-term memory. Inspired by this cognitive mechanism, we design a Spatiotemporal Memory Buffer module that caches visual and geometric information from the agent's recently visited locations, enabling semantic perception of frontier points. This module maintains a first-in-first-out (FIFO) observation queue of length M:

$$H = \{(I_{t-M+1}, D_{t-M+1}, p_{t-M+1}), \dots, (I_t, D_t, p_t)\}. \tag{2}$$

where each entry consists of an RGB image I_t, a depth image D_t, and the agent's global posep_t at time step t. This observation history is temporally continuous and spatially grounded, forming a lightweight short-term perceptual memory that serves as the input for the subsequent PathMatcher semantic scoring module.

3.4. PatchMatcher

After obtaining the candidate frontier set $F = \{\mathbf{f}_i \mid \mathbf{f}_i \in \mathbb{R}^2, i = 1, 2, \dots, N\}$,We design a core component, PatchMatcher, as the key part of the exploration module, which assigns task-relevant semantic value to each frontier point. By comparing the current navigation target with semantic patches extracted from historical observations, the module estimates the likelihood that each frontier leads to the target object. PatchMatcher implements an object-centric, semantics-driven frontier selection strategy and forms the core of our exploration module.

3.4.1. Frontier-to-History Association

The system maintains a history of past observations H. For each frontier point f_i, we first apply geometric priors to filter a subset of historical frames $H_{f_i} \subset H$ that are most likely to contain relevant information. This filtering process is based on two key geometric constraints:

1. Visibility constraint: We consider whether the camera's field of view at a certain past observation covers the current frontier point. Let the agent's pose at time t be $p_t = (x_t, y_t, \theta_t)$ and the frontier point be $f_i = (x_f, y_f)$. We define the relative angle between the frontier and the agent as:

$$\phi_t = \arctan 2(y_f - y_t, x_f - x_t) \tag{3}$$

If this angle falls within the horizontal field of view α, the frontier point is considered potentially visible:

$$|\phi_t - \theta_t| \leq \frac{\alpha}{2} \tag{4}$$

2. Distance constraint: Additionally, the Euclidean distance between the frontier point and the historical observation position must be below a threshold d_{th}:

$$\|f_i - (x_t, y_t)\|_2 \leq d_{\text{th}} \tag{5}$$

To ensure that the retrieved observations are both spatially relevant and reliable, and to reduce the impact of irrelevant data, we define the subset of historical observations retained for frontier point $\mathbf{f}_i$ as:

$$H_{f_i} = \left\{ (I_t, D_t, p_t) \in H \,\middle|\, \|f_i - (x_t, y_t)\|_2 \leq d_{\text{th}} \wedge |\phi_t - \theta_t| \leq \frac{\alpha}{2} \right\} \tag{6}$$

This process is described in Algorithm 1.

Algorithm 1 Associate Frontier Points with Historical Frames

```
Input: Frontier points $F = \{f_1, f_2, \ldots, f_N\}$, memory buffer $H$, distance $d_{th}$, angle threshold $\alpha$
Output: Associated frames $H_{f_i}$ for each $f_i$
for each $f_i$ in $F$ do
    Initialize $H_{f_i}$ as an empty set
    for each $(I_t, D_t, p_t)$ in $H$ do
        $\phi_t \leftarrow \text{arctan2}(y_f - y_t, x_f - x_t)$
        if $|\phi_t - \theta_t| \leq \frac{\alpha}{2}$ and $distance(f_i, (x_t, y_t)) \leq d_{th}$ then
            Add $(I_t, D_t, p_t)$ to $H_{f_i}$
        end if
    end for
end for
Return: $H_{f_i}$ for each $f_i$
```

3.4.2. Object Patch Extraction

We design an Object Patch Extraction module to identify semantically relevant object regions from selected historical observations. This module extracts the corresponding image regions as patches and feeds them into a downstream vision-language model. For each selected historical frame, we use a frozen object detector (Grounding DINO) to extract potential object regions from the image. Each detected bounding box is cropped into an individual image patch p_i, forming a patch set that represent the semantic memory most relevant to the frontier point f_i:

$$P_{f_i} = \{p_1, \ldots, p_n\} \tag{7}$$

Here, each p_j is an image region that is spatially associated with f_i and potentially semantically meaningful, to be used for subsequent matching and reasoning.

3.4.3. Semantic Froniter

Inspired by VLFM[15] methods, we adopt the vision-language model **BLIP-2**[8] to estimate the semantic relevance between each frontier point and the target object described in natural language. BLIP-2 computes a similarity score between an image patch and a text prompt representing the navigation goal. The core innovation of this paper is patch-level semantic matching based on object memory, which avoids redundant processing of the whole image. For each patch $p_j \in P_{f_i}$, we construct a text prompt T using natural language to describe the navigation objective, e.g., `"Seems like there is a <target object> ahead."`. We input both the patch and text into BLIP-2 to calculate the consine similarity score:

$$s_j = \text{BLIP2}(p_j, T) \tag{8}$$

To obtain the overall semantic score for each frontier point f_i, we average the similarity scores of all its associated patches:

$$S(f_i) = \frac{1}{|P(f_i)|} \sum_{j=1}^{|P(f_i)|} s_j \tag{9}$$

where $S(f_i)$ denotes the semantic relevance score of frontier point f_i.

Finally, we select the frontier point with the highest semantic score as the next exploration target:

$$f^* = \arg\max_{f_i \in F} S(f_i) \tag{10}$$

where F is the set of all currently detected frontier points.

This method achieves goal-driven semantic exploration usingge Model (VLM) for memory-based matching. By identifying semantically meaningful visual regions from historical observations, our approach significantly enhances both the efficiency and accuracy of semantic reasoning, thereby improving the object-centric frontier selection process based on memory.

3.5. Waypoint Navigation

The system uses the fast marching method (FMM) on the occupancy grid map to plan a collision-free path from the agent's position to a frontier point and guides navigation. During navigation, it follows VLFM's strategy to detect objects matching the target semantic label g. If found, the task succeeds and terminates; otherwise, the system re-evaluates semantic scores, selects a new goal, and repeats the process.

4. Experiments

4.1. Datasets

We evaluate our method on the HM3D and Gibson datasets within the Habitat simulator. HM3D includes 20 scenes with 2000 episodes, and Gibson includes 5 scenes with 1000 episodes. Task settings follow SemExp. Each episode lasts up to 500 steps and is successful if the agent issues a STOP within the goal object's region, as defined in the Habitat `ObjectNav` challenge.

4.2. Implementation details

Our experimental code is based on the open-source framework VLFM, which we extend with our own strategies, and is implemented using the PyTorch deep learning framework. We set the size of the historical frame buffer to $N = 50$ frames. The association distance between frontier points and historical frames is set to 3 meters, and the field-of-view (FoV) angle for visibility checks is set to 79°. For object patch extraction, we adopt the Grounding-DINO model.

4.3. Metrics

For all navigation experiments, we adopt Success Rate (SR) and Success weighted by Path Length (SPL) as evaluation metrics. SR is defined as the proportion of tasks in which the agent successfully navigates to the target object (within a distance of less than 1 meter) and correctly triggers the STOP action within a limited number of steps (500 steps):

$$SR = \frac{1}{N} \sum_{i=1}^{N} S_i \tag{11}$$

SPL is used to evaluate path efficiency and is defined as:

$$SPL = \frac{1}{N} \sum_{i=1}^{N} S_i \cdot \frac{L_i}{\max(p_i, L_i)} \tag{12}$$

where N is the total number of tasks, $S_i \in \{0, 1\}$ indicates whether task i is successful, L_i denotes the shortest path distance from the start to the target object and p_i is the actual path length taken by the agent.

4.4. Baselines

To evaluate the performance of our method, we compare it with supervised baselines including SemExp, FSE, PONI, and zero-shot methods such as ESC, L3MVN, and VLFM. All these methods adopt frontier-based exploration strategies:

- **SemExp** [2]is a classical modular method that constructs an explicit semantic map and combines it with goal-directed exploration strategies.
- **Frontier Semantic Exploration (FSE)** [7]builds semantic and frontier maps and leverages deep reinforcement learning strategies to select long-term exploration goals.
- **PONI**[6] predicts "region potential" and "goal potential" on the boundary of the semantic map, and learns to infer the likely goal location under environment-agnostic conditions.
- **ESC**[13] leverages large language models (LLMs) and GLIP-detected objects to reason about likely target boundaries and guide exploration.
- **L3MVN-2**[14] constructs semantic maps, forms simple sentences from objects around frontier regions and target-related entities, and scores them using a BERT model.
- **VLFM**[15] integrates visual observations and target categories into a VLM to generate a value map, which is then combined with frontier points for exploration.

5. Results

5.1. Benchmark results

Table 1
Results of Ablation Study in HM3D

	Method	**Gibson (val)**		**HM3D (val)**	
		SR ↑	**SPL ↑**	**SR ↑**	**SPL ↑**
Supervised	SemExp	65.2	33.6	37.9	18.8
	FSE	71.5	36.0	53.8	24.6
	PONI	73.6	41.0	–	–
Zero-shot	ESC	–	–	39.2	22.3
	L3MVN-Z	76.1	37.7	50.4	23.1
	VLFM	84.0	52.2	52.5	30.4
	OAM	**84.5**	**53.3**	**54.3**	**32.0**

As shown in Table 1, OAM consistently outperforms all baselines on both Gibson and HM3D datasets. Compared with supervised methods (SemExp, FSE, and PONI), OAM improves SR and SPL by up to 10. 9% and 12. 2% in Gibson and by 0. 5% and 7. 4% in HM3D. These gains come from leveraging pre-trained vision-language models instead of task-specific training, enhancing generalization in unseen environments. Compared with zero-shot methods, OAM surpasses ESC by 15. 1% SR and 9. 7% SPL in HM3D. It also outperforms L3MVN-Z by 8.4%/15.6% on Gibson and 3.9%/8.9% on HM3D (SR/SPL), demonstrating the advantage of using frontier-aligned semantic memory over constructing full semantic maps. Furthermore, OAM outperforms the second-best VLFM by 1. 8% SR and 1. 6% SPL in HM3D, indicating that object-level memory provides richer semantic cues for frontier selection, especially under occlusions or viewpoint shifts. The relatively lower SR of OAM on HM3D is due to the current lack of multi-floor navigation support, which limits performance on tasks involving cross-floor goal localization.

5.2. Ablation study

To understand the importance of each module in our framework, we conducted the following two ablation experiments on the HM3D dataset:

- **w/o memory buffer**: We remove the temporal memory buffer module. The agent is unable to reference historical observation information and can only make navigation decisions based on current visual input.
- **w/o object patches**: We remove the object-level semantic aggregation mechanism. The agent does not extract relevant patch information from historical frames around target objects, but instead relies solely on semantic understanding of complete images.

Table 2
Ablation Results on HM3D (val)

Method	SR ↑	SPL ↑
w/o memory buffer	51.2	31.3
w/o object patches	52.3	30.9
Ours	**54.3**	**32.0**

Table 2 reports the ablation results on the HM3D validation set, evaluating the contributions of two core components in our OAM framework: the temporal memory buffer and the object-level patch extraction module. When the memory buffer is removed, the agent can no longer utilize historical observations and must rely solely on the current frame for semantic reasoning. This leads to a performance drop in both SR (from 54.3% to 51.2%) and SPL (from 32.0% to 31.3%), demonstrating the importance of memory in maintaining consistent semantic understanding over time. Similarly, removing the object patch module, which prevents the model from aggregating fine-grained semantic cues around the frontier, leads to a slight degradation in SR (52.3%) and a more noticeable drop in SPL (30.9%). This suggests that patch-based local semantic alignment improves navigation efficiency by guiding the agent toward more semantically meaningful frontiers. In general, these results confirm that both modules contribute significantly to the robustness and efficiency of our zero-shot navigation system.

6. Conclusion

In this paper, we present OAM, a novel zero-shot object-goal navigation framework that integrates object-level spatiotemporal memory with a vision-language model. By retrieving semantically relevant historical observations around frontier regions, OAM enables goal-driven exploration without requiring task-specific training. This design overcomes the limitations of single-frame semantic reasoning and significantly improves both the navigation success rate and efficiency. Experimental results on the Gibson and HM3D benchmarks validate the effectiveness and generalizability of our approach. In future work, we aim to extend OAM to support multi-floor navigation and deploy it in real-world scenarios, further exploring the potential of memory-augmented semantic reasoning in embodied AI.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant No.62372049.

Declaration on Generative AI

In the preparation of this thesis, no generative AI tools were used.

References

[1] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, E. Wijmans, Objectnav revisited: On evaluation of embodied agents navigating to objects, 2020. URL: https://arxiv.org/abs/2006.13171. arXiv:2006.13171.

[2] D. S. Chaplot, D. Gandhi, A. Gupta, R. Salakhutdinov, Object goal navigation using goal-oriented semantic exploration, 2020. URL: https://arxiv.org/abs/2007.00643. arXiv:2007.00643.

[3] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, D. Batra, Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai, 2021. URL: https://arxiv.org/abs/2109.08238. arXiv:2109.08238.

[4] Y. Bai, X. Song, W. Li, S. Zhang, S. Jiang, Long-short term policy for visual object navigation, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023, pp. 9035–9042. doi:10.1109/IROS55552.2023.10341652.

[5] R. Fukushima, K. Ota, A. Kanezaki, Y. Sasaki, Y. Yoshiyasu, Object memory transformer for object goal navigation, in: 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 11288–11294. doi:10.1109/ICRA46639.2022.9812027.

[6] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, K. Grauman, Poni: Potential functions for objectgoal navigation with interaction-free learning, 2022. URL: https://arxiv.org/abs/2201.10029. arXiv:2201.10029.

[7] B. Yu, H. Kasaei, M. Cao, Frontier semantic exploration for visual target navigation, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, p. 4099–4105. URL: http://dx.doi.org/10.1109/ICRA48891.2023.10161059. doi:10.1109/icra48891.2023.10161059.

[8] J. Li, D. Li, S. Savarese, S. Hoi, Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL: https://arxiv.org/abs/2301.12597. arXiv:2301.12597.

[9] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, J. Gao, Grounded language-image pre-training, 2022. URL: https://arxiv.org/abs/2112.03857. arXiv:2112.03857.

[10] OpenAI, J. Achiam, S. Adler, S. Agarwal, et al., Gpt-4 technical report, 2024. URL: https://arxiv.org/abs/2303.08774. arXiv:2303.08774.

[11] 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', 1997, pp. 146–151. doi:10.1109/CIRA.1997.613851.

[12] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, S. Song, Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 23171–23181. doi:10.1109/CVPR52729.2023.02219.

[13] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, X. E. Wang, Esc: Exploration with soft commonsense constraints for zero-shot object navigation, 2023. URL: https://arxiv.org/abs/2301.13166. arXiv:2301.13166.

[14] B. Yu, H. Kasaei, M. Cao, L3mvn: Leveraging large language models for visual target navigation, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2023, p. 3554–3560. URL: http://dx.doi.org/10.1109/IROS55552.2023.10342512. doi:10.1109/iros55552.2023.10342512.

[15] N. Yokoyama, S. Ha, D. Batra, J. Wang, B. Bucher, Vlfm: Vision-language frontier maps for zero-shot semantic navigation, 2023. URL: https://arxiv.org/abs/2312.03275. arXiv:2312.03275.