Journal of Institute of Control, Robotics and Systems (2025) 31(8):888-893
https://doi.org/10.5302/J.ICROS.2025.25.0102 ISSN:1976-5622 eISSN:2233-4335

명시적·객체·위치 예측 및 미지 영역 탐색에서의 Large Language Model 활용 방법

Leveraging Large Language Model Assistance on Explicit Object Target Prediction and Navigation in Unseen Area

I Made Putra Arya Winata, 이 동 현, Ida Bagus Dwiweka Naratama, 오 정 현*
(I Made Putra Arya Winata[1], Donghyun Lee[1], Ida Bagus Dwiweka Naratama[1], and Junghyun Oh[1,*])
[1]Department of Robotics, Kwangwoon University

Abstract: Object-based goal navigation tasks require embodied agents to locate specified objects in unfamiliar environments, a key challenge in vision-language navigation. Several works use end-to-end models to project observations into actions, but they often suffer from poor interpretability. Conversely, modular approaches provide interpretability by separating perception, planning, and action. There is a notable modular framework that predicts object locations in unexplored areas via supervised learning. However, it is prone to false positives, which can degrade navigation performance. In this work, we propose an enhanced method that integrates large language model assistance into the object prediction pipeline. Specifically, we introduce a validation step that uses GPT-4o to confirm the existence of objects predicted using mask region-based convolutional neural network (Mask R-CNN) before retrieving them in semantic maps. This validation effectively filters false positives, leading to more accurate goal identification. Experiments on the HM3D dataset demonstrate improved success rates and trajectory efficiency, validating the effectiveness of our approach in complex indoor navigation scenarios.

Keywords: LLM, ObjectNav, semantic mapping, vision-language navigation

I. INTRODUCTION

Object navigation (ObjectNav) is a fundamental problem in embodied AI, where an agent is tasked with navigating an unfamiliar environment to locate a specified object [1]. This task has gained increasing importance in the broader field of vision-language navigation (VLN) [2-3], which aims to bridge perception and language for autonomous systems. Effective ObjectNav allows agents not only to interpret natural language commands but also to interact with environments in a meaningful way. Real-world applications include domestic robotics, assistive navigation, and search-and-rescue missions—scenarios where accurate object localization and efficient path planning are critical.

Existing approaches to ObjectNav can be broadly categorized into end-to-end and modular-based models. End-to-end methods train a single policy to map raw sensory input directly into actions. While these models benefit from optimization across the entire navigation pipeline, they often lack interpretability and are difficult to debug or extend. In contrast, modular-based systems divide the problem into interpretable components—such as perception, mapping, goal prediction, and planning—which allows for better transparency, control, and targeted improvements. Among these, PEANUT [4] has shown promising results by explicitly predicting the location of object goals in unexplored areas using a supervised learning framework.

However, PEANUT's reliance on region-based convolutional neural networks (R-CNN) [5] for semantic segmentation introduces significant limitations. In particular, R-CNN are prone to false positives, often misclassifying background textures or occluded regions as target objects. These inconsistent predictions, when incorporated into semantic maps [1], can mislead the agent during planning and navigation. To address this, we propose an enhanced ObjectNav framework that introduces two key improvements to the explicit object prediction process.

First, we replace the convolutional backbone used in PEANUT with a transformer-based architecture, specifically SegFormer. Transformers are inherently capable of modeling long-range dependencies and global spatial relationships, enabling more comprehensive and context-aware goal predictions across the entire semantic map. This global attention mechanism leads to more informed exploration, allowing the agent to identify potential object locations in a broader spatial context and improving its ability to discover target objects before episode termination.

Second, we introduce a large language model (LLM)-assisted

* Corresponding Author
Manuscript received April 18, 2025; revised May 29, 2025; accepted June 10, 2025
I Made Putra Arya Winata: 광운대학교 로봇학과 대학원생(aryawinata@kw.ac.kr, ORCID 0000-0002-2557-3994)
이동현: 광운대학교 로봇학과 대학원생(tjsqlfkdls@kw.ac.kr, ORCID 0009-0005-8220-2363)
Ida Bagus Dwiweka Naratama: 광운대학교 로봇학과 대학원생(dwiwekan@kw.ac.kr, ORCID 0009-0002-3381-5436)
오정현: 광운대학교 로봇학과 교수(jhyunoh@kw.ac.kr, ORCID 0000-0003-0502-7600)
※ This work was supported by the Technology Innovation Program (RS-2024-00445759, Development of Navigation Technology Utilizing Visual Information Based on Vision-Language Models for Understanding Dynamic Environments in Non-Learned Spaces) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea), by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE)(RS-2024-00406796, HRD Program for Industrial Innovation), and by the Excellent researcher support project of Kwangwoon University in 2025.

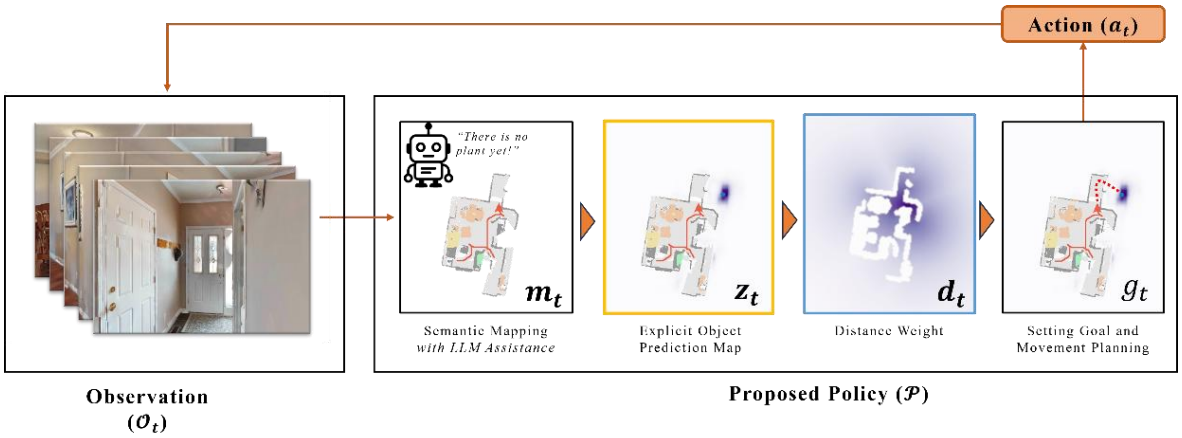

그림 1. 기존 문제 정의 및 그에 대한 제안 정책의 개요(각 단계별).
Fig. 1. Overview of proposed policy on given problem formulation (for each step).

validation mechanism to improve the reliability of object predictions. Given the tendency of Mask R-CNN to produce false positives, especially for small or hard-to-segment objects, we leverage the contextual reasoning capabilities of LLMs to verify the existence of predicted objects within the current observation. This validation step ensures that only credible predictions are incorporated into the semantic map, reducing noise and improving downstream navigation decisions.

II. RELATED WORK

1. Modular Approaches in Object Navigation

Modular methods have become increasingly robust in the field of object navigation (ObjectNav) due to their ability to decompose indoor scene navigation into interpretable and manageable sub-components—typically encompassing perception, planning, and control. This separation not only improves interpretability but also enables targeted improvements within each module, making the overall system more flexible and robust.

SemExp [1] initiated a spatial semantic map that guides exploration and decision-making. It extends active neural SLAM [6] to explicitly generate a semantic map, allowing the model to create a long-term policy based on prior semantic information. Building upon this foundation, several works start leveraging this feature into the use of various long-term policy strategies. The first to mention is Frontier SemExp [7], which introduced the frontiers concept using deep reinforcement learning. This work demonstrates significant breakthroughs; by exploring through frontiers, the model consistently finds the goal object effectively. However, its policy still lacks consistency, as it does not always succeed in locating the object. Another model adopts a supervised method to define its policy. PEANUT [4] extends the modular paradigm by explicitly modeling object likelihoods in unexplored areas of a scene. It leverages the semantic map constructed via SemExp to predict the most probable object locations even in regions the agent has not yet visited. This probabilistic map significantly enhances navigation efficiency. However, PEANUT's reliance on convolutional architectures, PSPNet [8], to model spatial distributions limits its capacity for long-range dependencies. The use of convolutional layers restricts the receptive field, which may hinder the model's ability to understand spatial relationships across the scene. To address this, attention-based mechanisms—such as those inspired by SegFormer [9] architecture—offer a promising alternative. Attention mechanisms can capture global spatial dependencies more effectively, enabling better reasoning over large, partially explored maps and enhancing object prediction capabilities in unobserved regions.

2. Language Models on Leveraging Visual Perception

Another notable limitation in current modular systems arises from their visual perception module, which typically uses models like Mask R-CNN [5] to segment and classify objects in the environment. While effective in many cases, this pipeline is susceptible to false positives—misclassifications or hallucinated detections that can lead to navigation inefficiencies or failures.

To mitigate this, recent research has explored the integration of large language models (LLMs) into modular navigation pipelines, like L3MVN [10]. This model inherits the Frontier SemExp [7] feature, which is based on frontiers policy. Through this frontiers policy, the model uses masked language model [11] to score the object near frontiers and select a future goal with the highest score. Although promising, this strategy heavily relies on LLM performance. The lack of visual interpretation during global goal selection has led to several mistakes. Therefore, in this paper, we leverage both visual and LLM representation to define the goal, in hope, the integration of these two will improve the performance.

III. PROPOSED METHOD

1. Problem Formulation and Overview

The object navigation (ObjectNav) task is formulated over a set of N episodes. In each episode, the agent is placed within a unique scene s, begins at an initial position p, and its objective is finding an object goal category c. For episode k, a tuple $\{s_k, p_k, c_k\}$ is given as input. The agent must navigate through the environment to find an instance considered to belong to object category c by following a certain policy $\mathcal{P}$. At each time step t, the agent receives an observation O_t, which consists of RGB and depth images capturing the surrounding environment. A navigation policy $\mathcal{P}$, denoted in Equation (1), processes this observation and outputs an action a_t. This process loops until the agent chooses the stop action, indicating that it has either found the target object or terminated the episode. The challenge lies in determining an effective policy $\mathcal{P}$ that enables the agent to

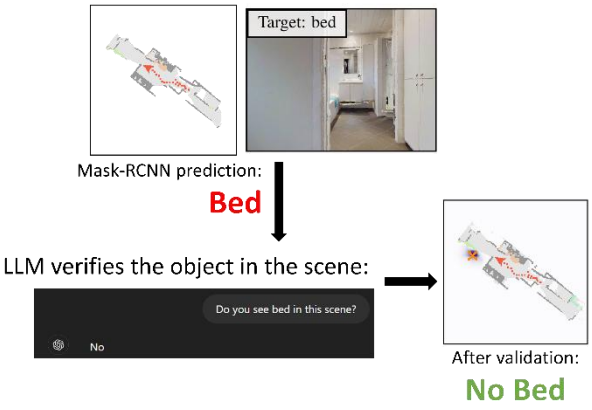

그림 2. LLM 보조 과정을 통한 객체 존재 검증.
Fig. 2. LLM assistance process on validating the existence of an object in the scene.

locate the target object efficiently and accurately across a variety of unseen scenes.

$$a_t = \mathcal{P}(O_t) \tag{1}$$

To resolve the problem mentioned, we present our redesigned model concept for ObjectNav tasks in [4], by leveraging LLM features in the supervised feature. Figure 1 shows the overview of the proposed policy model. The policy begins by projecting the observation into a top-down view using semantic mapping ($\boldsymbol{m_t}$) as proposed in [1]. From this semantic representation, we apply a trained explicit prediction model π to map the probability of a specific object category in unexplored areas. For each navigation episode, only one target object category is selected (c_{target}). For each feature of the policy will be detailed in following sections.

2. Semantic Map Projection (with LLM Assistance)

Our semantic mapping module builds on established practices in RGB-D-based scene understanding, incorporating both geometric and semantic follows [1,6]. At each timestep, the agent receives an RGB-D observation along with its pose information. The RGB image is processed using a Mask R-CNN [5] segmentation network to identify object instances and assign semantic category labels to each pixel. This segmentation covers C target object categories and may include additional background or contextual classes. The labeled pixels are then lifted into 3D using the depth data, resulting in a colored point cloud. To create a navigable map, only points within a predefined height range, corresponding to the agent's physical dimensions, are considered for obstacle labeling. The 3D point cloud is then converted into a voxel occupancy grid, which is collapsed along the vertical axis to form a 2D egocentric semantic map. This local map is transformed into an allocentric global frame using the agent's pose and aggregated over time to construct a global semantic map. Finally, this part of pipeline will output $(N+4) \times H \times W$ map, whereas H and W respectively show height and width of the proected map, and N shows the number of listed categories to be projected. There are additional 4 channels, which represent the channel for explored map, obstacle map, agent trajectory (from the beginning until current state), and current position of the agent.

To enhance the reliability of semantic mapping, we address a notable limitation of Mask R-CNN—its tendency to false positives,

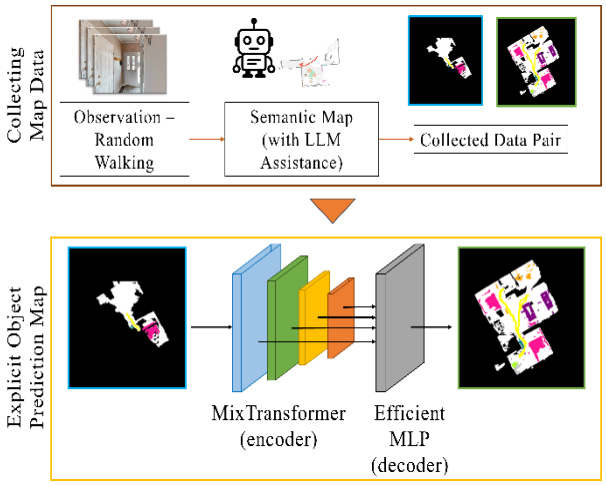

그림 3. 명시적인 객체 예측 맵 생성 과정.
Fig. 3. The process of explicit object prediction map.

especially when instance segmentation fails on similar objects and small objects. To mitigate this, we incorporate a Large Language Model (LLM) GPT-4o [12] to validate each segmented object instance, as shown in Figure 2. The LLM acts as a verifier by assessing whether a predicted object category is contextually consistent and likely to exist in the current scene. More specifically, this LLM is utilized whenever the agent has identified the object goal through Mask R-CNN. In this case, we use a flag called found_goal, set to 1 when the agent detects the object goal, and set to 0 otherwise. The input to the LLM comprises a text prompt and the current scene image. The text prompt asks, "Do you see <object_goal> in this scene?", along with attaching the image of the current scene in which the model detected the object goal. Intuitively, the LLM verifies the agent's judgement regarding the existence of the object goal. If the LLM confirms it as true, the agent remains using current goal. Otherwise, the found_goal flag is turned off, and the exploration continues. This additional validation step significantly reduces incorrect semantic labels before they are projected onto the global map.

3. Explicit Object Prediction Map

To estimate object probability maps in unexplored areas, we used transformer-based model with rich global contextual between pixels, by following [9]. As illustrated in Figure 3 (Explicit Object Prediction Map block), its four-stage Mix-Transformer encoder builds overlapping patch tokens; each stage couple reduction self-attention with a depth-wise 3x3 convolution, simultaneously capturing long-range dependencies and local details. This enables us to avoid using positional embeddings so it can be used on any input size. Finally, the process continues to decoder part which consists of efficient MLP that is more efficient than the use of convolution decoder.

Now, we describe the technical process for map projection. Similar to the segmentation process, instead of using 3 channels input, the map projection adapts the segmentation model to receive the semantic map output, as it is an extension of the semantic map, with the $(N+4)$ channels number. We trained the model with output of N_goal channels to explicitly represent the target object map ($\boldsymbol{m_t}$) in the task. Each channel in N_{goal} corresponds to one target object. For the next step, we retrieve the explicit prediction map to a model π, as mentioned earlier, which focuses solely on the target for current

episode (k), where the mapping corresponds to Equation (2). As a result, we obtain the probability map $\boldsymbol{z_t}$ of the object goal. Visualization of all this process is depicted in Figure 3.

$$\boldsymbol{z_t} = \pi(c_{target}|\boldsymbol{m_t}) \tag{2}$$

4. Setting Goal and Movement Planning

As we obtain $\boldsymbol{z_t}$, intuitively, we want the agent to go directly to the point with the highest value—indicating where the policy predicts the object is most likely located. However, this is not always optimal. For example, consider the first and second highest probability clusters on the map, where both are located in different regions. If the agent is significantly closer to the second-highest probability cluster, it would be reasonable for the agent to visit that area first. This suggests that distance plays a role in goal selection. Based on z_t, the goal location ($g_t \in \mathbb{N}^2$) is determined by selecting the point with the highest weighted probability, where the weights are influenced by geodesic distance ($\boldsymbol{d_t}$). The goal is selected according to the Equation (3):

$$g_t = \text{argmax}_{(i,j)} \exp(-\boldsymbol{d_t}[i,j])\, \boldsymbol{z_t}[i,j] \tag{3}$$

Finally, a movement plan is computed to navigate toward the selected goal g_t, guiding the agent's actions at each step throughout the episode.

IV. EXPERIMENT

1. Experiment Setup

We conduct our experiments on the HM3D [13] dataset, which offers a diverse environment and is well-suited for ObjectNav tasks. Prior to the main evaluation, we train our explicit goal prediction model using data collected through a random walk policy. Specifically, for every 4,000 newly explored pixels, the semantic map is annotated, and this process is repeated across 1,000 episodes to build a comprehensive dataset. The prediction model π is trained for 60,000 steps using all collected pair input-output maps. We train this model using Adam optimizer with a learning rate of 0.0005. We set the objective learning of binary cross-entropy as the loss function to ensure the result will show a probability in [0,1] each region, so that a region only belongs to at most one object.

For evaluation, we use the validation split of HM3D, running 500 episodes to test the agent's ability to navigate toward specified objects. We compare our method with several baseline models, both types that are without and with LLM.

a. **Random walk policy**, which serves as a lower bound to demonstrate the model's effectiveness is not a random success.
b. **Frontier SemExp** [7] is a model that improves frontier-based and mapping by using deterministic policy trained with reinforcement learning.
c. **PEANUT** [4], which serves as the primary benchmark for explicit prediction mapping, without relying on LLM.
d. **d.L3MVN [10] is a** model that uses the frontier-based like in Frontier SemExp. However, instead of training its deterministic policy, it leverages LLM to decide which frontier points the agent should explore.

To assess the contribution of the distance-based weighting scheme in goal selection, we perform an ablation study that contrasts two ways: (i) using the explicit object-prediction map alone and (ii) utilizing explicit object-prediction map and distance-based weight. This comparison reveals whether the weighting strategy with distance-based further optimizes agent's performance on the Object-Goal Navigation task.

2. Result and Discussion

Quantitatively, the experiment results are shown in Table 1. We are using two metrics, success rate (SR) that shows how many success episodes over all episodes and success weighted by path length (SPL) which follows Equation (4) that uses: binary point whether the episodes success ($S_i = 1$) or not ($S_i = 0$); path trajectory of the agent takes (P_i) and the shortest path from the start point to the actual goal (L_i). As an initial experiment, the random walk policy clearly demonstrates that a well-defined policy is required to find objects in this environment, as the random policy fails in all episodes. This is followed by experiments on several baseline models.

$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{L_i}{\max(L_i, P_i)} \tag{4}$$

At first glance, we observe that explicit models, PEANUT and Ours, achieve better metrics compared to frontier-based approaches. The primary reason might be that the goal policy in Frontier SemExp relies solely on a simple linear layer without further extraction of semantic knowledge. Then, we compare PEANUT and Ours, both of which do not include LLM features. The SPL metrics between these two indicate that replacing convolution-based models with attention modules for explicit prediction maps leads to more extensive exploration. This is reflected in the lower SPL of the model using attention mechanisms, which, in turn, results in a higher success rate.

Finally, we compare models with LLM-integrated policies. Observing direct comparisons, L3MVN againts Frontier SemExp and Ours againts Ours (without LLM), the metrics show improved performance for models with LLM. This suggests that LLM indeed contributes to enhanced scene understanding. Furthermore, among LLM-integrated models, the strategy of validating object existence to create robust semantic maps yields better performance. This improvement is likely due to the direct supervision that allows full utilization of the LLM's semantic reasoning capabilities.

We next evaluate whether integrating the distance-based weighting strategy improves goal selection. Table 2 compares explicit prediction alone and additional distance-based weight, for both without and with LLM. It shows adding distance-based weight improves both metrics. SPL increases significantly because the agent checks nearby candidate regions first, making the path more efficient before moving on to farther locations. Meanwhile, relying on the prediction map alone often causes worst performance because it wastes time searching. Hence, it may lead to exceeding the timestep limit and the episode is considered fail. Note that in these failed episodes, the found_goal flag is less likely active, so the LLM has no impact.

We conducted an additional analysis to check both complexities based on inference time and LLM related risk. For inference time related, we measured the average latency of each module on one navigation step (refer to Figure 1). Processing the raw observation and updating the semantic map takes 22 ms; the explicit prediction module adds 120 ms; distance-weight computation takes 136 ms; and goal-action planning requires 36 ms. Additionally, the GPT-4o as the LLM model could cause 2.6 s. However, we only use this LLM feature once when the found_goal is active. Hence, it is expected not to cause significant bottlenecks in the inference process.

그림 4. LLM을 활용한 목표 객체 발견 검증.

Fig. 4. Verification by LLM when the target object is found.

표 1. 기존 모델들 간의 비교.

Table 1. Comparison among prior models.

	Model	SR	SPL
w/o LLM	Random Walking	0.000	0.000
	Frontier SemExp	0.538	0.246
	PEANUT	0.606	**0.309**
	Ours (w/o LLM)	**0.614**	0.276
w/ LLM	L3MVN	0.542	0.255
	Ours	**0.636**	**0.312**

표 2. 거리 기반 최적화 실험.

Table 2. Experiment on distance-based optimization.

Goal Selection	LLM	SR	SPL
Explicit Prediction Only	No	0.520	0.228
Explicit Prediction with Distance Weight	No	0.614	0.276
Explicit Prediction Only	Yes	0.540	0.231
Explicit Prediction with Distance Weight	Yes	0.636	0.312

To assess risk, we examined cases in which the LLM reported object absence. In most scenarios, the LLM successfully resolves the false positive; for example, in Figure 4 (top), a bed frame initially misclassified as a chair. Then the model continued the exploration until it found a real chair. We observed only one failure, where LLM mistakenly dismissed a detected sofa as in Figure 4 (bottom). Then after exploring nearby, looking from a different pose, the sofa is detected again and LLM correctly verified. Overall, the LLM can be still reliable for various indoor scenario based on given scenes.

V. CONCLUSION

In this work, we propose an improved object navigation method by integrating large language models (LLMs) to validate explicit object goal predictions. This LLM-assisted validation reduces false positives from segmentation and enhances detection of small or ambiguous objects, leading to improved performance over prior methods such as PEANUT. The limitation of this study is that it only evaluates LLM support for explicit prediction on unexplored maps; applying the approach to real world scenarios remains future work, whether related to complexity or accuracy. Further research might be needed to test generalization to an open-set formulation—so it is not limited to a fixed set of object classes—in the future.

REFERENCES

[1] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol 33, pp.4247-4258. 2020.
doi: 10.48550/arXiv.2007.00643

[2] H. J. Sim, J. Kim, J. J. Hong, S. W. Nam, Y. H. Kim, J. Heo, H. C. Hwang, and K. K. Kim, "Visual navigation in unstructured environments and the development of a multi-purpose mobile robot platform," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 30, no. 9, pp. 913-923, 2024.
doi: 10.5302/J.ICROS.2024.24.0101

[3] J. Y. Yun and P. Kim, "Query-based object-aware mapping for on-device visual language mapping and navigation," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 30, no. 10, pp. 1068-1075, 2024.
doi: 10.5302/J.ICROS.2024.24.0169

[4] A. J. Zhai and S. Wang, "PEANUT: Predicting and navigating to unseen targets," *Proceedings of the International Conference on Computer Vision*, pp. 10926-10935, 2023.
doi: 10.1109/ICCV51070.2023.01003

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *Proceedings of the International Conference on Computer Vision*, pp. 2961-2969, 2017.
doi: 10.1109/ICCV.2017.322

[6] D. S. Chaplot, D. Gandhi, S. Gupta, S. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," arXiv Preprint, 2020.
doi: 10.48550/arXiv.2004.05155

[7] B. Yu, H. Kasaei, and M. Cao, "Frontier semantic exploration for visual target navigation," *Proceedings of the International Conference on Robotics and Automation*, pp. 4099-4105, May 2023.

doi: 10.1109/ICRA48891.2023.10161059

[8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 2881-2890, 2017.
doi: 10.1109/CVPR.2017.660

[9] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol 34, p. 12077-12090. 2021.
doi: 10.48550/arXiv.2105.15203

[10] B. Yu, H. Kasaei, and M. Cao, "L3MVN: Leveraging large language models for visual target navigation," *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 3554-3560, October 2023.
doi: 10.1109/IROS55552.2023.10342512

[11] W. Chen, S. Hu, R. Talak, and L. Carlone, "Leveraging large (visual) language models for robot 3D scene understanding," arXiv Preprint, 2022.
doi: 10.48550/arXiv.2209.05629

[12] OpenAI, "GPT-4o technical report", https://openai.com/index/gpt -4o, accessed: 2025-04-04, 2024.

[13] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, and D. Batra, "Habitat-matterport 3D Dataset (HM3D): 1000 large-scale 3D environments for embodied AI," arXiv Preprint, 2021.
doi: 10.48550/arXiv.2109.08238

I Made Putra Arya Winata

He graduated for B.S. of Mechanical Engineering Department, Udayana University in 2023. 2024~present, he studies in Department of Robotics, Kwangwoon University. His research interests include simultaneous localization and mapping, vision-language navigation and deep learning for robotics.

Donghyun Lee

He graduated for B.S. of Department of Robotics, Kwangwoon University in 2023. 2023~present, he studies in Department of Robotics, Kwangwoon University. His research interests include simultaneous localization and mapping, vision-language navigation and deep learning for robotics.

Ida Bagus Dwiweka Naratama

He graduated for B.S. of Information Technology Department, Udayana University in 2023. 2025~present, he studies in Department of Robotics, Kwangwoon University. His research interests include 3D reconstruction, vision-language navigation and deep learning for robotics.

Junghyun Oh

He received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from Seoul National University, Seoul, Korea, in 2012, 2014, and 2018, respectively. From 2018 to 2019, he served as a Senior Engineer at Samsung Research, Samsung Electronics Co., Ltd., Seoul, Korea. He joined the School of Robotics at Kwangwoon University, Seoul, Korea, as an Assistant Professor in 2019 and has been an Associate Professor since 2024. His research interests include vision-language navigation, multi-robot applications, and artificial intelligence for robotics.