Warehouse Spatial Question Answering with LLM Agent
1st Place Solution of the 9th AI City Challenge Track 3

Hsiang-Wei Huang[1*] Pyongkun Kim[2*] Jen-Hao Cheng[1] Kuang-Ming Chen[1]
Cheng-Yen Yang[1] Bahaa Alattar[1] Yi-Ru Lin[1] Sangwon Kim[2] Kwangju Kim[2]
Chung-I Huang[3] Jenq-Neng Hwang[1]

[1]Information Processing Lab, University of Washington, USA
[2]Electronics and Telecommunications Research Institute, South Korea
[3]National Center for High-performance Computing, Taiwan

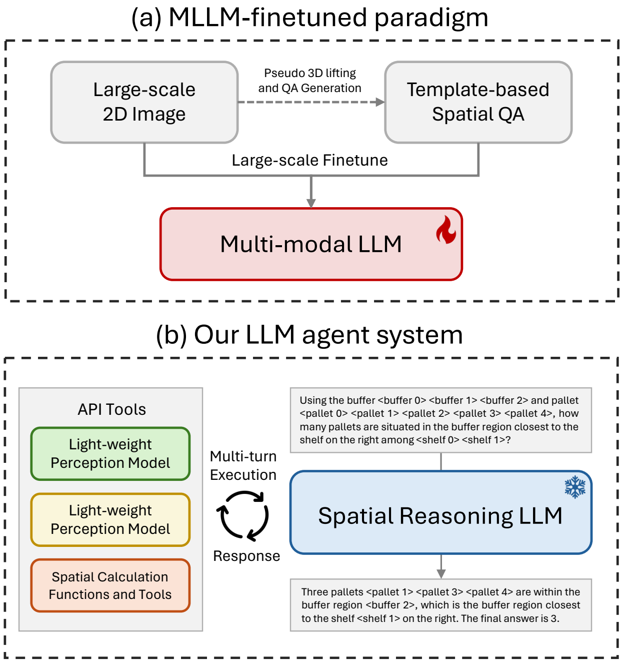
{hwhuang, andyhci, kmchen, cycyang, balattar, yirulin, hwang}@uw.edu,
{iros, eddiekim, kwangju}@etri.re.kr, 1203033@narlabs.org.tw

Abstract

Spatial understanding has been a challenging task for existing Multi-modal Large Language Models (MLLMs). Previous methods leverage large-scale MLLM finetuning to enhance MLLM's spatial understanding ability. In this paper, we present a data-efficient approach. We propose a LLM agent system with strong and advanced spatial reasoning ability, which can be used to solve the challenging spatial question answering task in complex indoor warehouse scenarios. Our system integrates multiple tools that allow the LLM agent to conduct spatial reasoning and API tools interaction to answer the given complicated spatial question. Extensive evaluations on the 2025 AI City Challenge Physical AI Spatial Intelligence Warehouse dataset demonstrate that our system achieves high accuracy and efficiency in tasks such as object retrieval, counting, and distance estimation. The code is available at: https://github.com/hsiangwei0903/SpatialAgent.

1. Introduction

In recent years, the advancement of Large Language Models (LLM) has revolutionized LLM system development, especially on the 3D and spatial understanding fields [2, 3, 5, 6, 14, 15, 17]. A key component of these LLM agent systems is the ability to perceive, localize, and reason about various objects in the 3D scene. However, accurately estimating spatial relationships between objects and conducting complex spatial reasoning remain challenging for these systems, especially in complex indoor scenarios. Specif-


(a) MLLM-finetuned paradigm
Large-scale 2D Image
Pseudo 3D lifting and QA Generation
Template-based Spatial QA
Large-scale Finetune
Multi-modal LLM
(b) Our LLM agent system
API Tools
Light-weight Perception Model
Light-weight Perception Model
Spatial Calculation Functions and Tools
Multi-turn Execution
Response
Spatial Reasoning LLM

Using the buffer <buffer 0> <buffer 1> <buffer 2> and pallet <pallet 0> <pallet 1> <pallet 2> <pallet 3> <pallet 4>, how many pallets are situated in the buffer region closest to the shelf on the right among <shelf 0> <shelf 1>?

Three pallets <pallet 1> <pallet 3> <pallet 4> are within the buffer region <buffer 2>, which is the buffer region closest to the shelf <shelf 1> on the right. The final answer is 3.

Figure 1. An illustration of (a) MLLM-finetuned paradigm like SpatialVLM [2] or SpatialRGPT [3], which lift the 2D image to pseudo 3D point cloud and generate template-based QA pairs for large-scale MLLM finetuning. In contrast, (b) illustrates our proposed LLM agent system, which utilize a LLM agent that can conduct spatial reasoning and utilize multiple light-weight perception models and tools for complex spatial question answering task.

[*] Equal contribution.

ically, these systems focus mainly on addressing simpler tasks like view-selection [15] or perception and grounding tasks [8, 12]. The research of using an LLM agent to solve complex spatial reasoning problems using functions and tools remain underexplored by current methods.

On the other hand, some recent methods [2, 3] directly conduct Multi-modal Large Language Model (MLLM) training for spatial visual question answering tasks. These approaches rely on large-scale training data, which leads to extensive training cost. Moreover, the template-based training QA generation paradigm also limits their ability to conduct complicated spatial reasoning.

In this paper, we propose a spatial understanding agent system designed to robustly analyze object relationships in complex indoor warehouse scenarios. Our system leverages a reasoning LLM as an AI agent to conduct spatial reasoning, function calling, and question answering. We integrate a series of functions and models to interact with the agent. Several common tools and functions such as distance estimation and spatial relationship recognition—that together enable comprehensive spatial reasoning about object relationships and support high-level decision-making tasks.

We evaluate our system on the challenging 2025 AI City Challenge Physical AI Spatial Intelligence Warehouse benchmark. We show that our approach achieves state-of-the-art QA accuracy, providing a practical solution for warehouse spatial understanding systems and facilitate LLM agent research in the spatial understanding domain.

We summarize our contributions as follows:

- We propose a spatial understanding LLM agent system with SoTA performance on warehouse spatial question answering tasks.
- Our designed LLM agent system possesses the ability to conduct complex spatial reasoning and further interact with multiple API tools, achieving data-efficient spatial question answering.
- We design multiple light-weight perception models and functions that can be used by the LLM agent, providing LLM agents active interaction and spatial reasoning over the given spatial query.

2. Related Work

2.1. LLM Agent

With the recent advancements of Large Language Models (LLM), many research have explored using LLM as an agent. These LLM agent systems possess strong ability in language interaction, action planning, function calling, and interaction with tools. LLM agent system has demonstrated success for a wide range of tasks including video understanding [11], visual question answering [1], embodied agent [7, 16], and 3D understanding [15]. Several recent works [12, 13, 15] focus on solving the 3D and spatial question answering tasks using LLM agents, yet they mostly focus on more simple sub-tasks such as view selection [8, 15], or only focus on visual grounding-related question [12–14]. The study of using LLM agents to perform both spatial reasoning and function calling to solve complex spatial understanding questions still remain underexplored. To this reason, we developed an intelligent LLM agent system that can perform both spatial reasoning and function calling to solve the challenging spatial question-answering task that involves diverse and complex natural language queries.

2.2. Spatial Understanding MLLM

Spatial understanding has been challenging for Multi-modal Large Language Models (MLLM), as it requires inferring the 3D information from the 2D image and performing complex spatial reasoning. Existing works like SpatialVLM [1] and Spatial-RGPT [3] leverage large-scale data for spatial understanding MLLM training, which incurs extensive collection and training cost. In this work, we leverage multiple light-weight perception models and a spatial reasoning-enabled LLM agent to achieve data-efficient spatial question answering.

3. Method

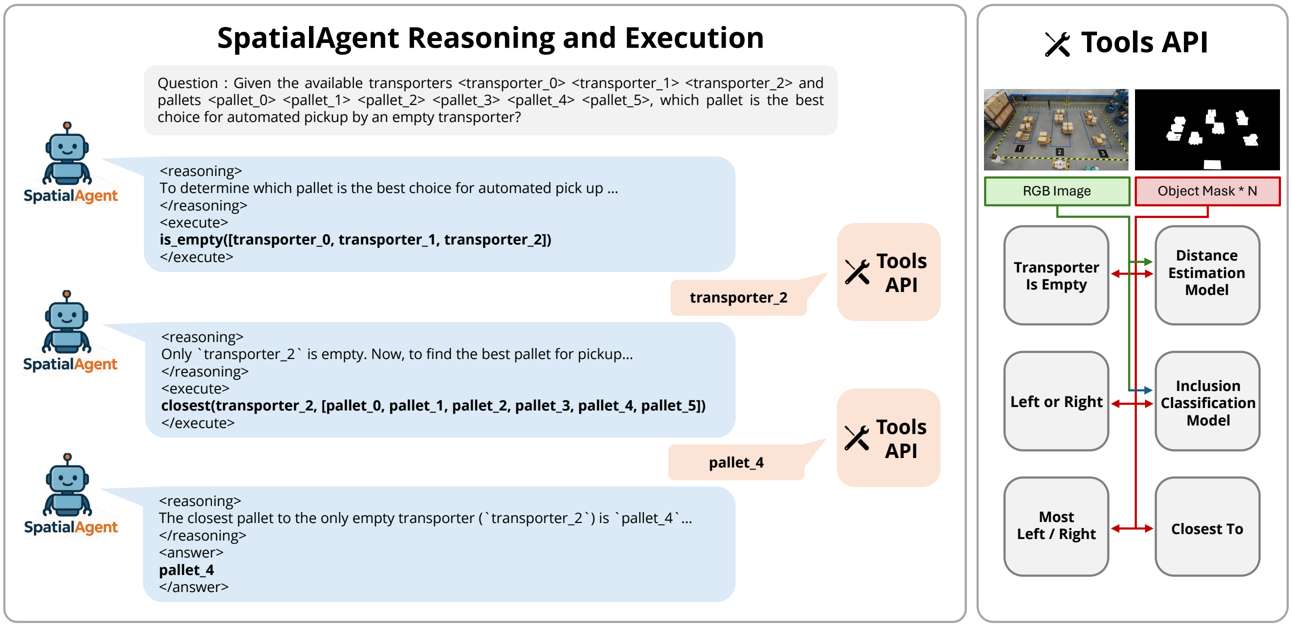
3.1. Spatial Agent

Our spatial agent is designed to answer complex spatial questions by leveraging a Large Language Model (LLM) with function-calling capabilities. The agent uses Gemini 2.5-Flash [10], and optionally supports its think mode configuration that activates Gemini's built-in reasoning budget to enhance multi-step reasoning performance.

Given an input image, a pair of binary masks, and a spatial question, the agent first parses and identifies relevant object masks using a rule-based parser. These masks are mapped to region identifiers and registered in the tool API. The question is then combined with a few-shot prompting template and passed to the Gemini model. The agent maintains a structured message history and a multi-turn conversation with the LLM to guide the reasoning process.

During inference, the agent interacts with our provided functions and tools via our specified `<execute>` tag (e.g., `<execute> dist(obj_1,obj_2) </execute>`). These commands are parsed and then dispatched to a pre-defined set of spatial APIs, including distance estimation, object inclusion, relative positioning (left/right), and region queries (e.g., most left, middle). Execution results are returned to the LLM, which may iteratively refine its reasoning before producing the final answer, enclosed in `<answer>` tags when the LLM confirms the final answer.

The provided spatial APIs are a series of common functions and tools that can help the agent find out the answer. For simpler spatial relationships like left/right, we utilized



Question : Given the available transporters <transporter_0> <transporter_1> <transporter_2> and pallets <pallet_0> <pallet_1> <pallet_2> <pallet_3> <pallet_4> <pallet_5>, which pallet is the best choice for automated pickup by an empty transporter?

```
<reasoning>
To determine which pallet is the best choice for automated pick up ...
</reasoning>
<execute>
is_empty([transporter_0, transporter_1, transporter_2])
</execute>
```

transporter_2

```
<reasoning>
Only `transporter_2` is empty. Now, to find the best pallet for pickup...
</reasoning>
<execute>
closest(transporter_2, [pallet_0, pallet_1, pallet_2, pallet_3, pallet_4, pallet_5])
</execute>
```

pallet_4

```
<reasoning>
The closest pallet to the only empty transporter (`transporter_2`) is `pallet_4`...
</reasoning>
<answer>
pallet_4
</answer>
```

Figure 2. An illustration of our SpatialAgent framework.

the object mask centroid coordinate on the image to determine its spatial relationship. For more complex spatial relationships like distance estimation and determining whether an object is inside a specific region, we train a deep learning model to conduct end-to-end prediction. We introduce more details on these models in Sec. 3.2 and Sec. 3.3.

3.2. Distance Estimation Model

Since the AI City Challenge Physical AI Spatial Intelligence Warehouse dataset does not provide camera parameters or absolute depth information, we formulate the distance estimation task as a direct regression problem using only the available image and mask data. Given an RGB image $I \in \mathbb{R}^{H \times W \times 3}$ and two binary object masks $M_1, M_2 \in \{0,1\}^{H \times W}$, we aim to learn a model F that predicts the absolute distance $D \in \mathbb{R}$ between the two objects. This is expressed as $D = F(I, M_1, M_2)$. The model is trained to minimize the error between the predicted distance $\hat{D}$ and the ground-truth distance D_{gt}, using a standard regression L2 loss:

$$\mathcal{L}_{\text{dist}} = \left\| \hat{D} - D_{\text{gt}} \right\|_2^2$$

We adopt ResNet-50 [4] with 5 input channels that take an RGB image and two binary object masks as input.

In our experiments, we found that the model does not achieve satisfactory accuracy on smaller distance estimation (less than 3m). To address this, we further fine-tuned another distance estimation model F_{small}, which only trained on data with groundtruth distances smaller than 3m. In our final implementation, we cascade the two distance estimation models F and F_{small}, whenever F predicts a value smaller than 3m, we use F_{small} to predict again, and use its prediction as our final answer for distance estimation questions.

3.3. Inclusion Classification Model

In addition to distance estimation, we also design a binary classification model to determine whether one object is spatially included within another. Specifically, given an RGB image $I \in \mathbb{R}^{H \times W \times 3}$ and two binary object masks $M_1, M_2 \in \{0,1\}^{H \times W}$, we train a model G to predict an inclusion label $y \in \{0,1\}$, where $y = 1$ indicates that object A (corresponding to M_1) is spatially included inside object B (corresponding to M_2), and object B is required to be a buffer region object. The model is trained using the focal loss [9] which can be expressed as:

$$\mathcal{L}_{\text{inc}} = -\alpha_t (1 - p_t)^\gamma \log(p_t)$$

where p_t is the model's predicted probability for the true class label y, and α_t, γ are hyperparameters that control the focusing and weighting behavior of the loss. We follow the distance estimation model and use ResNet-50 as the backbone for the inclusion classification model.

4. Experiments

4.1. Benchmark

In our experiments, we utilize the 2025 AI City Challenge Physical AI Spatial Intelligence Warehouse dataset, a large-scale synthetic benchmark focusing on spatial reasoning and question answering within warehouse environments. This dataset is generated using the NVIDIA Omniverse platform and provides rich multimodal inputs, including RGB-D image pairs, object masks, and natural language QA pairs.

The QA pairs are categorized into four main types: spatial relations, multi-choice selection, distance estimation, and object counting. Each question is accompanied by both a free-form answer and a normalized single-word response for quantitative evaluation. The dataset comprises approximately 499K question and answer pairs for training, 1.9K for validation, and 19K for testing.

4.2. Evaluation

We follow the official evaluation protocol of the AI City Challenge Physical AI Spatial Intelligence Warehouse benchmark. The primary metric is the weighted average success rate across all question types. A prediction is considered successful if it satisfies the Acc@10 criterion (within ±10% of ground truth) for distance estimation and counting questions. For multi-choice and spatial relation questions, exact match accuracy is used.

4.3. Implementation Details

Question Pre-processing. To enable the LLM Agents understanding of the mask-to-object correspondence from the given question, we adopt a simple heuristic rule to pre-process the input question. We search the very first target object (buffer, pallet, transporter, or shelf) that appears before each `<mask>`, and modify each `<mask>` to our specified object-aware format `<object_ID>` (e.g. `<buffer_0>`, `<transporter_1>`). After this pre-processing, the LLM can understand the mask-to-object relationship and conduct spatial reasoning and function calling. In some corner case questions where there is no preceding target object before `<mask>`, we query an LLM [10] to rephrase the question to our specified object-aware format.

Spatial Agent. We use Gemini-2.5-Flash [10] as our LLM agent, building on top of Google Vertex API. We set the temperature to 0.2 and other LLM parameters as default. During the question answering process, if the agent fails to successfully execute a function, we re-run by adding 128 tokens of thinking budget, this enables LLM to conduct more detailed reasoning and thus prevent format error.

Distance Estimation Model. We use ResNet-50 as our distance estimation model backbone, the training data is collected from the distance estimation question in the training set. A total of 245K data points collected from the training set are used to train our distance estimation model. We trained both distance estimation models for 5 epochs with L2 loss, using image resolution of (640, 480) and a learning rate of 1e-4. We also scale the groundtruth unit by 100 (from meter to centimeter), which helps to provide stronger supervision during training.

Ranking	Team Name	Accuracy
1	**UWIPL_ETRI (Ours)**	**95.8638**
2	HCMUT.VNU	91.9735
3	Embia	90.6772
4	MIZSU	73.0606
5	HCMUS_HTH	66.8861
6	MealsRetrieval	53.4763
7	BKU22	50.3662
8	Smart Lab	31.9245
9	AICV	28.2993

Table 1. Leaderboard of the 9th AI City Challenge Track 3: Warehouse Spatial Intelligence.

Inclusion Classification Model. We use ResNet-50 as our inclusion classification model, the training data is collected from the counting question's free-form answer in the training set. 158K data points collected from the training set are used to train our inclusion classification model. We trained the model for 5 epochs using focal loss, with an image resolution of (640, 480) and a learning rate of 1e-4.

4.4. Performance

We compared our performance with other teams on the 2025 AI City Challenge Track 3 leaderboard in Tab. 1. Our proposed system achieve 95.86% accuracy on the testing set of the Physical AI Spatial Intelligence Warehouse Benchmark, ranking 1st place among all teams.

5. Conclusion

In this work, we present a spatial understanding LLM agent system for complex indoor warehouse environments. Our system bridges the gap between perception and high-level reasoning by equipping a reasoning LLM with specialized API tools to support complex spatial question answering. Our method achieve state-of-the-art accuracy on the 2025 AI City Challenge Physical AI Spatial Intelligence Warehouse benchmark, demonstrating the effectiveness and generalizability of our approach.

6. Acknowledgement

This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government (Development of ICT Convergence Technology for Daegu-GyeongBuk Regional Industry) under Grant 25ZD1120. We also want to acknowledge and thank NCHC from Taiwan for providing the computing resources.

References

[1] Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *arXiv preprint arXiv:2406.13642*, 2024. 2

[2] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024. 1, 2

[3] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2025. 1, 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[5] Hsiang-Wei Huang, Wenhao Chai, Kuang-Ming Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Tosa: Token merging with spatial awareness. *arXiv preprint arXiv:2506.20066*, 2025. 1

[6] Hsiang-Wei Huang, Fu-Chen Chen, Wenhao Chai, Che-Chun Su, Lu Xia, Sanghun Jung, Cheng-Yen Yang, Jenq-Neng Hwang, Min Sun, and Cheng-Hao Kuo. Zero-shot 3d question answering via voxel-based dynamic token compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19424–19434, 2025. 1

[7] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37: 100428–100534, 2024. 2

[8] Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3707–3717, 2025. 2

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3

[10] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2, 4

[11] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2024. 2

[12] Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. In *CoRL*, 2024. 2

[13] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F. Fouhey, and Joyce Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7694–7701, 2024. 2

[14] Zhihao Yuan, Jinke Ren, Chun-Mei Feng, Hengshuang Zhao, Shuguang Cui, and Zhen Li. Visual programming for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20623–20633, 2024. 1, 2

[15] Sha Zhang, Di Huang, Jiajun Deng, Shixiang Tang, Wanli Ouyang, Tong He, and Yanyong Zhang. Agent3d-zero: An agent for zero-shot 3d understanding. In *European Conference on Computer Vision*, pages 186–202. Springer, 2024. 1, 2

[16] Zhonghan Zhao, Wenhao Chai, Xuan Wang, Boyi Li, Shengyu Hao, Shidong Cao, Tian Ye, and Gaoang Wang. See and think: Embodied agent in virtual environment. In *European Conference on Computer Vision*, pages 187–204. Springer, 2024. 2

[17] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8995–9006, 2025. 1