CityEQA: A Hierarchical LLM Agent on Embodied Question Answering Benchmark in City Space

Anonymous ACL submission

Abstract

Embodied Question Answering (EQA) has primarily focused on indoor environments, leaving the complexities of urban settings—spanning environment, action, and perception—largely unexplored. To bridge this gap, we introduce **CityEQA**, a new task where an embodied agent answers open-vocabulary questions through active exploration in dynamic city spaces. To support this task, we present **CityEQA-EC**, the first benchmark dataset featuring 1,412 human-annotated tasks across six categories, grounded in a realistic 3D urban simulator. Moreover, we propose **Planner-Manager-Actor (PMA)**, a novel agent tailored for CityEQA. PMA enables long-horizon planning and hierarchical task execution: the *Planner* breaks down the question answering into sub-tasks, the *Manager* maintains an object-centric cognitive map for spatial reasoning during the process control, and the specialized *Actors* handle navigation, exploration, and collection sub-tasks. Experiments demonstrate that PMA achieves 60.7% of human-level answering accuracy, significantly outperforming competitive baselines. While promising, the performance gap compared to humans highlights the need for enhanced visual reasoning in CityEQA. This work paves the way for future advancements in urban spatial intelligence. Dataset and code are available at `https://anonymous.4open.science/r/CityEQA-3027`.

1 Introduction

Embodied Question Answering (EQA) (Das et al., 2018) represents a challenging task at the intersection of natural language processing, computer vision, and robotics, where an embodied agent (e.g., a UAV) must actively explore its environment to answer questions posed in natural language. While most existing research has concentrated on indoor EQA tasks (Gao et al., 2023; Peña-Narvaez et al., 2023) or traditional indoor/outdoor Visual Question Answering (VQA) tasks (Sun et al., 2024), relatively little attention has been dedicated to EQA tasks in open-ended city space, as shown in Table 1. Nevertheless, extending EQA to city space is crucial for numerous real-world applications, including autonomous systems (Kalinowska et al., 2023), urban region profiling (Yan et al., 2024), and city planning (Gao et al., 2024).

EQA tasks in city space (referred to as CityEQA) introduce a unique set of challenges that fundamentally differ from those encountered in indoor environments. Compared to indoor EQA, CityEQA faces three main challenges:

1) **Environmental complexity with ambiguous objects**: Urban environments are inherently more complex, featuring a diverse range of objects and structures, many of which are visually similar and difficult to distinguish without detailed semantic information (e.g., buildings, roads, and vehicles). This complexity makes it challenging to construct task instructions and specify the desired information accurately (Ji et al., 2025; Xu et al., 2025).

2) **Action complexity in cross-scale space**: The vast geographical scale of city space compels agents to adopt larger movement amplitudes to enhance exploration efficiency. However, it might risk overlooking detailed information within the scene. Therefore, agents require cross-scale action adjustment capabilities to effectively balance long-distance path planning with fine-grained movement and angular control.

3) **Perception complexity with observation dynamics**: Observations can vary greatly depending on distance, orientation, and perspective. For example, an object may look completely different up close than it does from afar or from different angles. These differences pose challenges for consistency and can affect the accuracy of answer generation, as embodied agents must adapt to the dynamic and complex nature of urban environments.

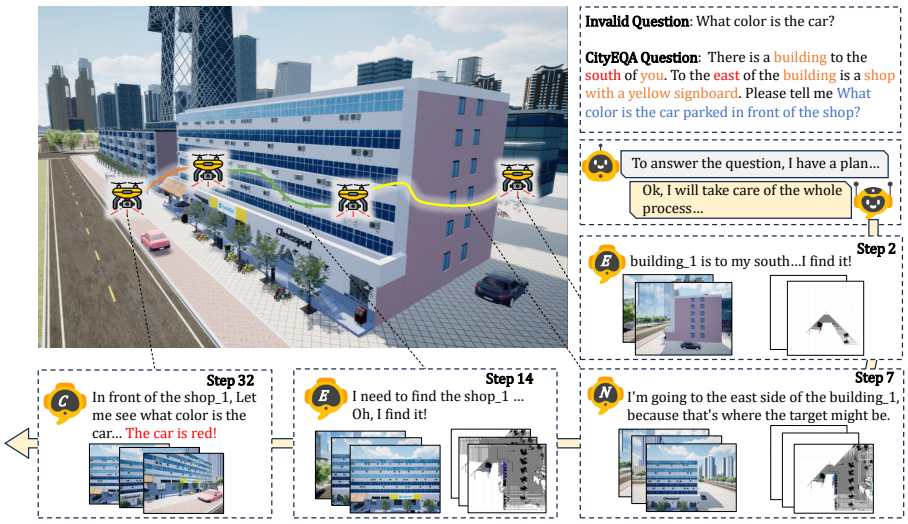

Figure 1: The typical workflow of the PMA to address City EQA tasks. There are two cars in this area, thus a valid question must contain landmarks and spatial relationships to specify a car. Given the task, PMA will sequentially complete multiple sub-tasks to find the answer.

Table 1: CityEQA-EC vs existing benchmarks.

	Platform	Reference	Place	Open Vocab	Active
EQA-v1	House3D	(Das et al., 2018)	Indoor	✗	✓
IQUAD	AI2-THOR	(Gordon et al., 2018)	Indoor	✗	✓
MP3D-EQA	Matterport3D	(Wijmans et al., 2019)	Indoor	✗	✓
MT-EQA	House3D	(Yu et al., 2019)	Indoor	✗	✓
K-EQA	AI2-THOR	(Tan et al., 2023)	Indoor	✗	✓
HM-EQA	HM3D	(Ren et al., 2024)	Indoor	✗	✓
S-EQA	VirtualHome	(Dorbala et al., 2024)	Indoor	✗	✓
NoisyEQA	-	(Wu et al., 2024)	Indoor	✓	✓
OpenEQA	ScanNet/HM3D	(Majumdar et al., 2024)	Indoor	✓	✓
City-3DQA	-	(Sun et al., 2024)	Outdoor	✓	✗
EarthVQA	-	(Wang et al., 2024)	Outdoor	✓	✗
Open3DVQA	-	(Zhan et al., 2025)	Outdoor	✓	✗
CityEQA-EC	EmbodiedCity	-	Outdoor	✓	✓

As an initial step toward CityEQA, we developed **CityEQA-EC**, a benchmark dataset to evaluate embodied agents' performance on CityEQA tasks. The distinctions between this dataset and other EQA benchmarks are summarized in Table 1. CityEQA-EC comprises six task types characterized by open-vocabulary questions. These tasks utilize urban landmarks and spatial relationships to delineate the expected answer, adhering to human conventions while addressing object ambiguity. This design introduces significant complexity, turning CityEQA into long-horizon tasks that require embodied agents to identify and use landmarks, explore urban environments effectively, and refine observation to generate high-quality answers.

To address CityEQA tasks, we introduce the **Planner-Manager-Actor (PMA)**, a novel baseline agent powered by large models, designed to emulate human-like rationale for solving long-horizon tasks in urban environments, as illustrated in Figure 1. PMA employs a hierarchical framework to generate actions and derive answers. The Planner module parses tasks and creates plans consisting of three sub-task types: navigation, exploration, and collection. The Manager oversees the execution of these plans while maintaining a global object-centric cognitive map (Deng et al., 2024). This 2D grid-based representation enables precise object identification (retrieval) and efficient management of long-term landmark information. The Actor generates specific actions based on the Manager's instructions through its components: Navigator, Explorer, and Collector. Notably, the Collector integrates the Vision Language Model (VLM) as its Vision Language Action (VLA) module to refine observations and generate high-quality answers. PMA's performance is assessed against five types of baselines, including humans. Results show that humans perform best in CityEQA, while PMA achieves 60.73% of human accuracy in answering questions, highlighting both the challenge and validity of the proposed benchmarks.

In summary, this paper makes the following significant contributions:

- To the best of our knowledge, we present the first open-ended embodied question answering benchmark for city space, namely CityEQA-EC.
- We propose a novel baseline model, PMA, which is capable of solving long-horizon tasks for CityEQA tasks with a human-like rationale.
- Experimental results demonstrate that our approach outperforms existing baselines in tackling the CityEQA task. However, the gap with

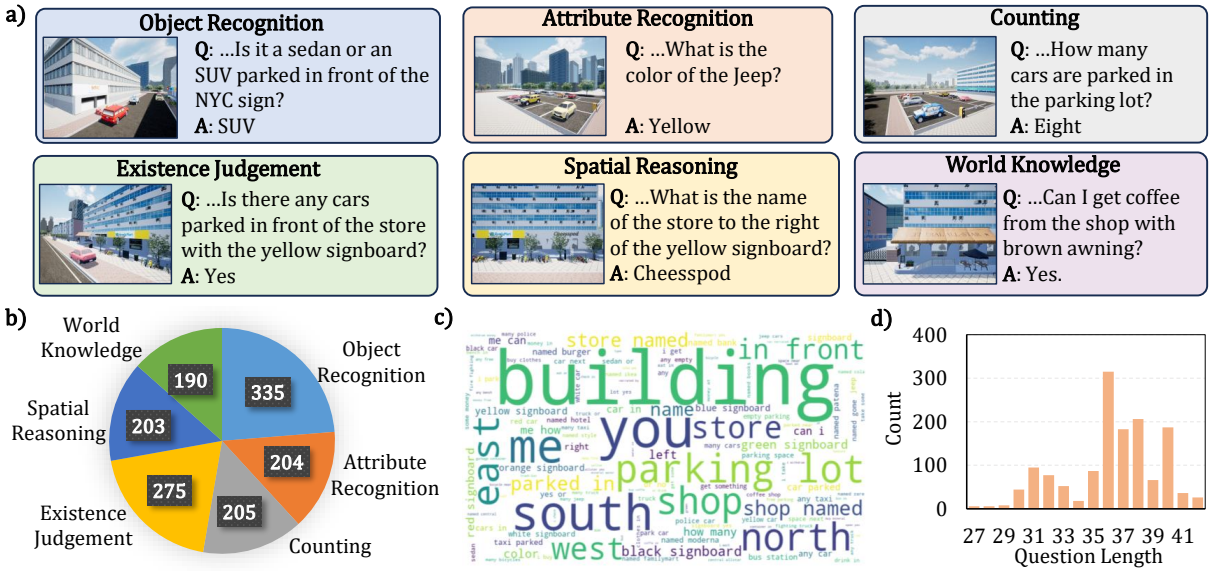

Figure 2: Task examples and dataset statistics of the CityEQA-EC.

human performance highlights opportunities for future research to improve visual thinking and reasoning in embodied agents for city spaces.

2 CityEQA-EC Dataset

In this section, we outline the formulation of the EQA task and describe the dataset collection process for CityEQA-EC. To address real-world demands, such as urban governance and public services, we draw upon previous research (Majumdar et al., 2024; Das et al., 2018) to define six distinct task types. Examples and statistics of the dataset are presented in Figure 2.

2.1 Task Formulation

An instance of the EQA task is defined by the 4-tuple: $\xi = (e, q, y, p_0)$, where e is the simulated or real 3D scene that agent can interact with, q is the question, and y is the ground truth answer. The p_0 denotes the agent's initial pose, including 3D position and orientation. Given the instance ξ, the goal is for the embodied agent (e.g., drones) to complete the task by gathering the required information from e and generating the answer $\hat{y}$ in response to q. Specifically, the agent starts at the initial pose p_0 and interacts with the scene e step by step. At each time step t, the agent can move to a specific pose p_t, and obtain an observation $o_t = (I_t^{rgb}, I_t^d)$ from the scene, where $I_t^{rgb} \in \mathbb{R}^{H\times W\times 3}$ is the RGB image and $I_t^d \in \mathbb{R}^{H\times W}$ is the depth image. Based on these observations, the agent generates the answer $\hat{y}$. The key challenge is to produce a high-quality answer while minimizing the time steps required.

2.2 Dataset Collection and Validation

To obtain a high-quality dataset, we employed EmbodiedCity (Gao et al., 2024), which is a highly realistic 3D simulation platform based on the buildings, roads, and other elements in a real city. It is implemented using Unreal Engine 4 (Sanders, 2016) and Microsoft AirSim plugins (Shah et al., 2018). The collection process is to determine the 4-tuple elements $\xi = (e, p_0, q, y)$ of each instance. Unlike indoor simulators with many different scenes, EmbodiedCity is a coherent and extensive scene. As a result, for all instances, their scene e corresponds to EmbodiedCity.

The dataset collection process involves two steps, completed by five human annotators. The first step is raw Q&A generation, where raw questions and answers are created. The second step is task supplementation, which includes determining the agent's initial pose and and refining the question descriptions accordingly. Once these steps are completed, the dataset undergoes validation and filtering. More details can be found in Appendix A.1.

Raw Q&A Generation We instructed human annotators to explore the EmbodiedCity environment freely and generate question-answer pairs based on their observations of RGB images. The raw questions q^r and answers y are presented as open-vocabulary text. In addition to documenting the question-answer pairs, annotators were also required to record the pose p^{obs} from which the RGB images were captured, along with the pose p^{tar} of the target object referenced in each question. These

information can be leveraged for a comprehensive evaluation of the agent's performance. After basic revision process, we have finally collected a total of 443 such instances, with each raw task instance denoted as $\xi^r = (q^r, y, p^{obs}, p^{tar})$.

Task Supplementation Building upon the raw task instances, we further established the agent's initial pose and refined the questions accordingly. For each raw task, the initial pose p_0 of the agent was set within a 200-meter range of the target object's pose p^{tar}. Given the complexity of urban environments, and to ensure that each expected answer is unique, we enriched the questions with descriptions based on landmarks. An example of this process is illustrated in Figure 1. For each raw task, we generated at least four distinct initial poses and transformed each raw question into at least four different inquiries. Ultimately, this process yielded a total of 2,212 task instances.

Dataset Validation Each task instance created by human annotators was rigorously evaluated by two independent human reviewers. These reviewers were responsible for determining whether the questions posed were answerable and clear, as well as verifying the uniqueness and accuracy of the target objects and their corresponding answers. Any task instance identified with issues was excluded. The final dataset comprises 1,412 task instances, with detailed statistics presented in Figure 2.

3 PMA: A Hierarchical LLM Agent for CityEQA Task

3.1 Overview

An overview of the proposed PMA agent for CityEQA tasks is shown in Figure 3. The PMA comprises three major modules: Planner, Manager, and Actor, all powered by pre-trained foundation models. Planner is responsible for parsing the question q and formulating an executable plan before any actions are taken. Manager serves as the core module, receiving structured information from Planner and processing observations at each time step to maintain an object-centric cognitive map using an VLM. Additionally, through a process control module, Manager issues task instructions to Actor, which then utilizes various action generators to execute the required responses. Once the plan is completed, Manager generates an answer based on its accumulated memory.

3.2 Planner Module

The question descriptions in CityEQA tasks contain extensive information, including several objects, spatial relationships, and the information that needs to be collected. To address the open-ended question descriptions, we leveraged pre-trained LLMs and designed a few-shot prompt that employs a three-step Chain of Thought (CoT) reasoning (Wei et al., 2022) to parse the question and formulate a plan.

As illustrated in Figure 3, all objects and spatial relationships mentioned in the question are first extracted. Simultaneously, the information necessary to answer the question is identified as corresponding requirements. Based on these requirements, a plan is created consisting of three distinct types of sub-tasks: (1) **Collection sub-tasks** gather the requisite information, (2) **Exploration sub-tasks** identify landmarks or target objects, and (3) **Navigation sub-tasks** enable efficient access to specific areas, thereby narrowing the exploration scope. To ensure the plan is executable, we have developed several strategies to guide the LLMs, with details provided in Appendix A.2.

3.3 Manager Module

The Manager possesses the capability to oversee and manage the gradual implementation of long-term plans. This is made possible by its Memory module and Map module, which facilitate the organized storage of observations and track execution progress as the plan unfolds.

Object-Centric Cognitive Map The object-centric cognitive map takes the initial pose of the agent as the origin, uses 2D grids to discretize the surrounding environment, and records the distribution of landmark objects based on grid indices. The map at time step t-1 is represented as M_{t-1}=$\{obj_1, obj_2, ...\}$, where the obj_1 and obj_2 are the object IDs corresponding to specific objects in the environment. At each time step t, the agent leverages egocentric observations represented as $o_t = (I_t^{rgb}, I_t^d)$ to construct the added map m_t to record the landmark objects appeared at current observation, denoting as $m_t = Construct(o_t, p_t)$. To implement the functionality of $Construct()$, we utilized the Ground-SAM model (Bousselham et al., 2024) for grounding and segmenting landmark objects from I_t^{rgb}. By integrating pose information with depth data from I_t^d, we can obtain a 3D point cloud representation of these objects, subsequently projected

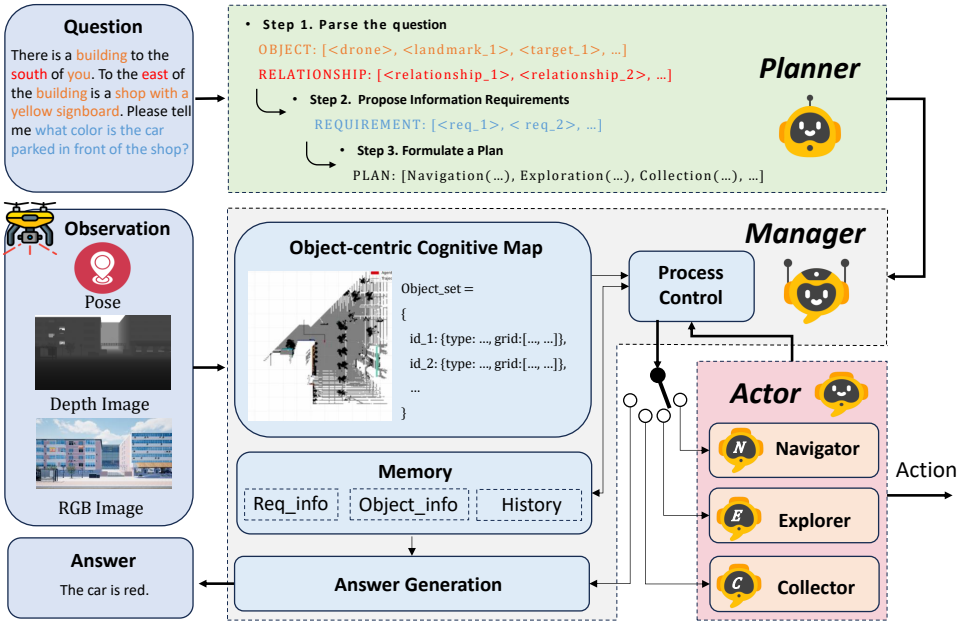

Figure 3: The overview of our proposed PMA agent.

onto 2D grids. After denoising and filtering, we obtained the finalized added map, denoted by m_t.

The added map m_t will be fused with the $M_{t\text{-}1}$ by merging the same object observed at different time steps, so objects are guaranteed to be unique in the map, denoting as $M_t = Merge(m_t, M_{t-1})$. More details can be found in Appendix A.2.

Other Modules Memory module records important information in the perceptual process, which mainly includes three aspects. *Req_info* records the collected information, and *Object_info* records object information, such as the object's ID in the map. *History* records the completion progress of sub-tasks and the execution results of actions.

Process Control is designed to determine the next sub-task to be executed based on the current progress of the plan. It also serves as the interface for interaction with the Actor. Once all sub-tasks in the plan have been completed, Process Control invokes the Answer Generation module to produce the final response. The Answer Generation process is also driven by LLMs, employing a zero-shot prompt specifically crafted to generate answers based on the *Req_info* stored in memory.

3.4 Actor Module

To address the distinct objectives of the three types of sub-tasks, we introduce three specialized low-level action generators: Navigator, Explorer, and Collector. The Navigator and Explorer rely on distinct deterministic policies to generate actions based on the cognitive map. In contrast, the Collector uses a VLA policy, which directly derives actions from RGB images. These action models serve as fundamental baselines and provide a foundation for future research enhancements.

Navigator The navigation sub-task instructions specify a landmark and a directional relationship. For instance, *Navigation(building_1, west)* indicates that building_1 serves as the landmark, with navigation directed to the west of it, where the target object is likely located. The Navigator identifies the nearest navigation point on the map by analyzing the landmark's distribution in conjunction with its spatial relationship. It then employs the A* algorithm to plan a path from the agent's current position to this navigation point. Given the potential incompleteness of recorded landmarks on the map, a multi-step approach is adopted, restricting each step's path length L^{nav} to 10 meters. The navigation point is updated following each cognitive map update.

Explorer The typical exploration sub-task is described as *Exploration(building_1, west, red_car)*, which means the goal is to explore the west side of *building_1* to find a red car. The explorer uses the Move and Look Around strategy due to the complexity of outdoor environments, where re-observing previously explored areas from different angles can yield different results. The exploration area is defined on the map based on land-

mark distribution and spatial relationships. A set of exploration points is generated within this area, maintaining a fixed distance of $L^{exp} = 10$ meters between them. At each point, the agent thoroughly observes its surroundings by looking in four directions: front, back, left, and right. After completing observations at one point, the agent moves to the next closest point and continues until either the target object is found or all points are covered. A VLM is employed to determine whether the target appears in any given observation.

Collector The collection sub-task instructions only include an information requirement. We provide a VLM-driven Collector to gather the required information from observations. Additionally, the Collector can select an action from a predefined action set to fine-tune its observation view, enabling the collection of higher-quality information. More details of Collector is presented in Appendix A.2.

4 Experiment

4.1 Experiment setup

Evaluation Metrics In CityEQA, we adopted three widely used metrics for evaluating EQA tasks (Das et al., 2018): Question Answering Accuracy (QAA) assesses the correctness of the answers by comparing them to the ground truth. The open-vocabulary nature of the CityEQA task poses challenges for evaluation. Inspired by OpenEQA (Majumdar et al., 2024), we employed an LLM as the judge to assign scores $\theta \in \{1, 2, ..., 5\}$ to the answers. For detailed information, please refer to the Appendix A.3. Navigation Error (NE) is measured by the distance between the agent's final position and the target object p^{tar} upon task completion, reflecting whether the agent successfully located and approached the target. Mean Time Step (MTS) is calculated as the average number of time steps required to complete all tasks, indicating the efficiency of the embodied agent's action strategy.

Implementation Details For each task, the object-centric cognitive map is constructed centered around the agent's initial pose, with a side length of 400 meters and a resolution of 1 meter. The dimension of the images obtained by the agent is 640×480, and we considered buildings as landmarks and accounted for four spatial relationships: north, south, east, and west. Additionally, the total number of time steps for navigation and exploration is limited to 50 steps and the maximum steps for collection is 10. GPT-4o and GPT-4 are the default VLM and LLM used in the PMA. Due to API limitations, 200 tasks are randomly selected from CityEQA-EC for the experiments.

Baselines We compare various models in a zero-shot setting, including five categories of baselines that are widely used in studies of EQA tasks. More details of baselines can be found in Appendix A.3.

- **Blind Agents (Majumdar et al., 2024)** generate answers based solely on the text of questions without obtaining any visual inputs. It serves as a reference for assessing the extent to which one can rely purely on prior world knowledge and/or random guessing.
- **Socratic Agents (Jiang et al., 2025)** use the VLM (GPT-4o) to convert the visual input during the exploration process into image captions, and then uses LLMs to generate answers based on these descriptions.
- **VQA Agents** bypass the active exploration process and is directly provided with the RGB image obtained from the p^{obs} to answer the questions. This approach aims to assess the visual perception and reasoning capabilities of VLMs in urban environments, while eliminating the interference of embodied actions.
- **Exploring Agents (Ren et al., 2024)** actively acquire visual inputs using Random Exploration (RE) and Frontier-Based Exploration (FBE), both commonly used as indoor baselines.
- **Human Agents** are employed to establish human-level performance metrics on our benchmark. We categorize human agents as H-VQA or H-EQA, depending on whether they actively acquire visual inputs.

4.2 Comparison with State-of-the-art

As shown in Table 2, human agents in both VQA and EQA settings achieve the highest QAA scores—4.87±0.72 for H-VQA and 4.94±0.21 for H-EQA—representing the upper bound for answer quality. They also demonstrate exceptional efficiency, with the lowest navigation error (38.72±40.17m) and completion steps (9.31±6.32), setting the gold standard for both quality and efficiency.

For automated methods, VQA agents like GPT-4o reach QAA scores up to 4.37±1.35, approaching human performance in answer quality, but lack active exploration abilities, preventing assessment of

Table 2: Performance of baselines and the proposed PMA on the CityEQA tasks.

	QAA (1-5) ↑	NE (m) ↓	MTS ↓
Blind Agents			
GPT-4	1.90±1.64	-	-
Qwen-2.5	2.34±1.88	-	-
LLaMA-v3.1-8b	2.31±1.72	-	-
DeepSeek-v3	2.03±1.41	-	-
Socratic Agents (VLM/LLM)			
GPT-4o/GPT-4	2.71±1.72	-	-
GPT-4o/Qwen-2.5	2.77±1.49	-	-
GPT-4o/LLaMA-v3.1-8b	2.70±1.71	-	-
GPT-4o/DeepSeek-v3	2.82±1.53	-	-
VQA Agents			
GPT-4o	4.37±1.35	-	-
Qwen-2.5	4.00±1.67	-	-
LLaVA-v1.5-7b	3.81±2.01	-	-
Exploring Agents			
RE	2.19±2.64	73.31±45.43	46.41±10.41
FEB	2.31±2.54	86.92±53.71	39.31±32.17
Human Agents			
H-VQA	4.87±0.72	-	-
H-EQA	4.94±0.21	38.72±40.87	9.31±6.32
PMA (ours)	3.00±1.96	46.56±36.39	24.44±14.39

their overall task efficiency. Blind and Socratic agents perform significantly worse, with QAA between 1.90 and 2.82, showing the shortcomings of methods without visual information or with only language-based reasoning.

Exploring agents such as RE and FEB can handle active exploration and answering, but their QAA scores are low (2.19–2.31) and their NA and MTS are much higher (e.g., FEB: 86.92±53.71m, 39.31±32.17), resulting in less effective execution. In contrast, PMA achieves a balanced performance: its QAA of 3.00±1.96 is higher than all exploring, blind, and Socratic agents, though still just 60.73% that of H-EQA. Importantly, PMA's navigation error (46.56±36.39m) and completion step (24.44±14.39) are dramatically less than traditional exploring agents, demonstrating notable practical gains.

Overall, the comparison with baselines reveals that *accurate visual inputs and reasoning are crucial for improving performance in CityEQA tasks*. Additionally, obtaining accurate visual inputs *relies on the efficient exploration using landmarks and spatial relationships* in urban environments.

4.3 Ablation Studies

We conduct ablation studies on the Object-Centric Cognitive Map, navigator, and explorer modules in PMA, as shown in Table 3. Removing any of these modules leads to a significant decline in performance. Without the map, the agent becomes

Table 3: Ablation results.

	QAA (1-5) ↑	NE (m) ↓	MTS ↓
PMA w/o map	2.31±1.82	76.41±48.64	43.27±31.92
PMA w/o navigator	2.33±1.64	68.31±46.91	38.83±27.71
PMA w/o explorer	2.68±1.87	57.13±41.43	20.62±15.11
PMA	3.00±1.96	46.56±36.39	24.44±14.39

confused by similar landmarks in the environment and fails to perform effective active perception, resulting in the worst ablation outcome. Furthermore, the absence of the navigator is more detrimental than that of the explorer, further highlighting the importance of landmark-based navigation in urban environments.

4.4 Effectiveness of Collector Module

This section further investigates the effectiveness of the collector module, specifically the impact of fine-grained observation adjustments on performance. We recorded the observation at each step (10 steps in total) during the collection phase and calculated relevant metrics, as shown in Figure 4.

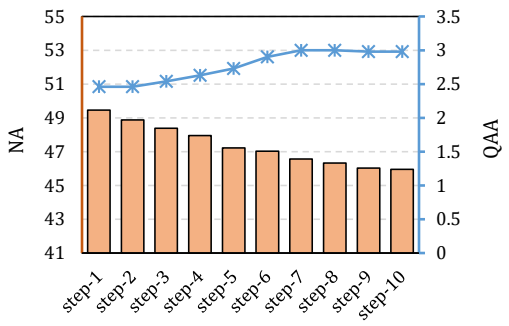

Figure 4: The performance of the Collector module at different steps.

The Collector significantly affects outcomes: as steps increase, NE decreases and QAA rises, helping the agent approach targets and improve accuracy. However, QAA plateaus, with Step 10 slightly lower than Step 9, possibly due to "over-adjustment" degrading visual input quality.

We further analyzed the Collector's taken actions, as detailed in Appendix A.3. The most frequent action was *KeepStill*, reflecting effective Navigation and Exploration sub-tasks that help the agent successfully approach the target object. Additionally, the proportions of *MoveForward*, *TurnLeft*, and *TurnRight* were also relatively high. Case analysis revealed that when a target object enters the agent's view, it tends to stop, possibly cause the object too far away or only partially visible. In

such instances, the agent must either *MoveForward* to reduce distance or use *TurnLeft* and *TurnRight* to adjust its orientation for better observation and information gathering about the target object. However, these adjustments remain limited, as illustrated in two cases presented in Appendix A.3.

5 Related Works

5.1 QA and EQA

Early research on using language to guide perception from given input is known as Question Answering (QA), such as Visual QA (VQA) (Ishmam et al., 2024) and 3DQA (Zhan et al., 2025). These QA tasks require agents to answer questions based solely on provided information (images or cloud points) (Chandrasegaran et al., 2024). In contrast, EQA involves agents actively exploring within an environment to seek visual inputs and enhance answer reliability (Das et al., 2018). Due to cost and hardware limitations, several virtual indoor simulators have been developed for EQA tasks (Liu et al., 2024a), resulting in indoor-focused datasets such as EQA-v1 (Das et al., 2018) and MT-EQA (Yu et al., 2019). However, although there are already several QA task datasets for outdoor environments, such as City-3DQA (Sun et al., 2024) and Open3DVQA (Zhan et al., 2025), EQA tasks have yet to be extended to outdoor settings, as shown in Table 1.

Recently, urban environment simulators like EmbodiedCity (Gao et al., 2024), CityNav (Lee et al., 2024), and AerialVLN (Liu et al., 2023) have emerged, though they mainly focus on navigation task. EmbodiedCity provides an urban EQA dataset, but it functions more like VQA and ignores the active perception. Moreover, due to the limited generalization capabilities of models at the time, only simple questions about basic attributes of objects were considered in these indoor datasets (Ren et al., 2024). However, with the continuous improvement in the understanding and reasoning capabilities of pre-trained VLMs for visual inputs, several open-ended EQA datasets have recently been released, such as Express-bench (Jiang et al., 2025) and OpenEQA (Majumdar et al., 2024). In comparison, this paper is the first to study the EQA tasks in city space and introduces the benchmark CityEQA-EC — a high-quality dataset featuring diverse, open-vocabulary questions.

5.2 LLMs-driven Embodied Agents

The indoor EQA tasks mainly involve exploration and answer generation sub-tasks (Ren et al., 2024). In early work (Duan et al., 2022; Das et al., 2018; Lu et al., 2019), the two sub-tasks are mainly addressed by building and fine-tuning various deep neural networks. Recently, researchers attempt to utilize pre-trained LLMs to solve EQA tasks without any additional fine-tuning (Mu et al., 2024; Xiang et al., 2024; Huang et al., 2024). NaviLLM employed a scheme-based instruction that flexibly casts various tasks into generation problems, including the EQA task (Zheng et al., 2024). OpenEQA employed a Frontier-Based Exploration (FBE) strategy for indoor environment exploration and tested the performance of various VLMs on the answer generation (Majumdar et al., 2024). Besides, VLMs was also used to determine which room to explore in indoor environment based their commonsense reasoning capabilities (Yin et al., 2025).

These agents, however, cannot be directly used for CityEQA tasks. Unlike indoor spaces, which are confined and divided into rooms, city spaces are vast and open. Agents in cities must navigate using landmarks and spatial relationships for long-term exploration (Liu et al., 2024b). The proposed PMA addresses this by breaking down and planning for long-horizon CityEQA tasks, using large models across multiple modules to effectively handle open-ended questions and unseen environments.

6 Conclusion

This paper pioneers the exploration of EQA tasks in outdoor urban environments. First, we introduced CityEQA-EC, the inaugural open-ended benchmark for CityEQA, comprising 1,412 tasks divided into six distinct categories. Second, we proposed a novel agent model (the PMA), designed to tackle long-horizon tasks through hierarchical planning, sensing, and execution. Experimental results validated the effectiveness of PMA, achieving 60.73% accuracy relative to human performance and outperforming traditional methods such as the FBE Agent. Nevertheless, challenges remain, including efficiency discrepancies (24.44 vs. 9.31 mean time steps taken by humans) and limitations in visual thinking capabilities. Future research could focus on enhancing PMA with self-reflection and error-correction mechanisms to mitigate error accumulation that can arise in long-horizon tasks.

7 Limitations

The work primarily focuses on object-centric question-answering tasks, such as identifying specific objects (e.g., buildings, vehicles) within city spaces. Further, while our approach is effective for tasks involving static physical entities, it overlooks the importance of social interactions and dynamic events, which are also critical in urban settings. For instance, questions related to dynamic events (e.g., "Is there a traffic jam on Main Street?"), or environmental conditions (e.g., "Is the park crowded right now?") are not considered up to now. These types of questions require some different sets of reasoning capabilities, such as temporal reasoning, event detection, and social context understanding, which are not currently supported by the Planner-Manager-Actor (PMA) agent. Future work should expand the scope of CityEQA to include these non-entity-based tasks, further extending PMA and enabling embodied agents to handle a broader range of urban spatial intelligence challenges.

8 Ethics Statement

In the data collection, we ensure there is no identifiable information about individuals (faces, license plates) or private properties. Thus, there is no ethical concern.

References

Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. 2024. Grounding everything: Emerging localization properties in vision-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3837.

Keshigeyan Chandrasegaran, Agrim Gupta, Lea M Hadzic, Taran Kota, Jimming He, Cristobal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. 2024. Hourvideo: 1-hour video-language understanding. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10.

Yinan Deng, Jiahui Wang, Jingyu Zhao, Xinyu Tian, Guangyan Chen, Yi Yang, and Yufeng Yue. 2024. Opengraph: Open-vocabulary hierarchical 3d graph representation in large-scale outdoor environments. *arXiv preprint arXiv:2403.09412*.

Vishnu Sashank Dorbala, Prasoon Goyal, Robinson Piramuthu, Michael Johnston, Reza Ghanadhan, and Dinesh Manocha. 2024. S-eqa: Tackling situational queries in embodied question answering. *arXiv preprint arXiv:2405.04732*.

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

Chen Gao, Si Liu, Jinyu Chen, Luting Wang, Qi Wu, Bo Li, and Qi Tian. 2023. Room-object entity prompting and reasoning for embodied referring expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen Gao, Baining Zhao, Weichen Zhang, Jinzhu Mao, Jun Zhang, Zhiheng Zheng, Fanhang Man, Jianjie Fang, Zile Zhou, Jinqiang Cui, and 1 others. 2024. Embodiedcity: A benchmark platform for embodied agent in real-world city environment. *arXiv preprint arXiv:2410.09604*.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4089–4098.

Siyuan Huang, Iaroslav Ponomarenko, Zhengkai Jiang, Xiaoqi Li, Xiaobin Hu, Peng Gao, Hongsheng Li, and Hao Dong. 2024. Manipvqa: Injecting robotic affordance and physically grounded information into multi-modal large language models. *arXiv preprint arXiv:2403.11289*.

Md Farhan Ishmam, Md Sakib Hossain Shovon, Muhammad Firoz Mridha, and Nilanjan Dey. 2024. From image to language: A critical analysis of visual question answering (vqa) approaches, challenges, and opportunities. *Information Fusion*, page 102270.

Yatai Ji, Zhengqiu Zhu, Yong Zhao, Beidan Liu, Chen Gao, Yihao Zhao, Sihang Qiu, Yue Hu, Quanjun Yin, and Yong Li. 2025. Towards autonomous uav visual object search in city space: Benchmark and agentic methodology. *arXiv preprint arXiv:2505.08765*.

Kaixuan Jiang, Yang Liu, Weixing Chen, Jingzhou Luo, Ziliang Chen, Ling Pan, Guanbin Li, and Liang Lin. 2025. Beyond the destination: A novel benchmark for exploration-aware embodied question answering. *arXiv preprint arXiv:2503.11117*.

Aleksandra Kalinowska, Patrick M Pilarski, and Todd D Murphey. 2023. Embodied communication: How robots and people communicate through physical interaction. *Annual review of control, robotics, and autonomous systems*, 6(1):205–232.

Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. 2024. Citynav: Language-goal

aerial navigation dataset with geographic information. *arXiv preprint arXiv:2406.14240*.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. 2023. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15384–15394.

Yang Liu, Weixing Chen, Yongjie Bai, Guanbin Li, Wen Gao, and Liang Lin. 2024a. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *CoRR*.

Youzhi Liu, Fanglong Yao, Yuanchang Yue, Guangluan Xu, Xian Sun, and Kun Fu. 2024b. Navagent: Multi-scale urban street view fusion for uav embodied vision-and-language navigation. *arXiv preprint arXiv:2411.08579*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, and 1 others. 2024. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2024. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36.

Juan Diego Peña-Narvaez, Francisco Martín, José Miguel Guerrero, and Rodrigo Pérez-Rodríguez. 2023. A visual questioning answering approach to enhance robot localization in indoor environments. *Frontiers in Neurorobotics*, 17:1290584.

Allen Z Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. 2024. Explore until confident: Efficient exploration for embodied question answering. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

Andrew Sanders. 2016. *An introduction to Unreal engine 4*. AK Peters/CRC Press.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics: Results of the 11th International Conference*, pages 621–635. Springer.

Penglei Sun, Yaoxian Song, Xiang Liu, Xiaofei Yang, Qiang Wang, Tiefeng Li, Yang Yang, and Xiaowen Chu. 2024. 3d question answering for city scene understanding. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2156–2165.

Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. 2023. Knowledge-based embodied question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11948–11960.

Junjue Wang, Zhuo Zheng, Zihang Chen, Ailong Ma, and Yanfei Zhong. 2024. Earthvqa: Towards queryable earth via relational reasoning-based remote sensing visual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5481–5489.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. 2019. Embodied question answering in photorealistic environments with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6659–6668.

Tao Wu, Chuhao Zhou, Yen Heng Wong, Lin Gu, and Jianfei Yang. 2024. Noisyeqa: Benchmarking embodied question answering against noisy queries. *arXiv preprint arXiv:2412.10726*.

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. 2024. Language models meet world models: Embodied experiences enhance language models. *Advances in neural information processing systems*, 36.

Haotian Xu, Yue Hu, Chen Gao, Zhengqiu Zhu, Yong Zhao, Yong Li, and Quanjun Yin. 2025. Geonav: Empowering mllms with explicit geospatial reasoning abilities for language-goal aerial navigation. *arXiv preprint arXiv:2504.09587*.

Yibo Yan, Haomin Wen, Siru Zhong, Wei Chen, Haodong Chen, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. 2024. Urbanclip: Learning text-enhanced urban region profiling with contrastive language-image pretraining from the web. In *Proceedings of the ACM on Web Conference 2024*, pages 4006–4017.

Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. 2025. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L Berg, and Dhruv Batra. 2019.

Multi-target embodied question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6309–6318.

Weichen Zhan, Zile Zhou, Zhiheng Zheng, Chen Gao, Jinqiang Cui, Yong Li, Xinlei Chen, and Xiao-Ping Zhang. 2025. Open3dvqa: A benchmark for comprehensive spatial reasoning with multimodal large language model in open space. *arXiv preprint arXiv:2503.11094*.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. 2024. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13624–13634.

A Appendix

A.1 Dataset Collection and Validation

The collection and validation process of the CityEQA-EC dataset is shown in Figure 5, including Initialization (Step 1), Raw Q&A Generation (Step 2 to 4), Task Supplementation (Step 5 to 6), and Dataset Validation (Step 7).

In the initialization phase, human annotators were provided with comprehensive briefings and training, during which they were introduced to six distinct types of tasks. Subsequently, in the raw question-and-answer generation stage, annotators were randomly placed within the environment, allowing them to move freely and explore in order to generate questions and answers. Additionally, both the target pose p^{tar} and observed pose p^{obs} were recorded manually. Then, each question-answer pair was then reviewed by two additional annotators to identify specific issues: (1) Task Duplication, indicating that a similar instance had already been collected; (2) Task Invalidity, meaning that there was no match between the question and answer based on the image. Any tasks identified as problematic were discarded. Furthermore, to ensure the accuracy of pose annotations, we randomly selected 20% of raw task examples for two rounds of verification regarding their pose annotations.

In the task supplementation phase human annotators were asked to add the initial pose for the task and expand the question. Buildings are primarily used as landmark objects to expand the question. Then, in the validation stage, each task was independently evaluated by two human reviewers. The details of the review policy are as follows:

- Spelling and grammar check is conducted.
- The target object must be uniquely identifiable based on descriptions of landmarks and spatial positions.
- The distance between the initial pose and the target pose must be less than 200 meters.
- The initial pose is located at a movable position rather than within an obstacle.

Any tasks identified as problematic were removed. To ensure the annotation consistency in the data collection and validation, we conducted Kappa statistical analyses for the raw annotation data from both Question and Answer revision phase and the task validation phase. The Kappa coefficients κ for the two phases were 0.93 and 0.89, respectively, indicating a high level of agreement among annotators.

A.2 PMA Agent Details

Details of Planner We present the detailed CoT used by the Planner here.

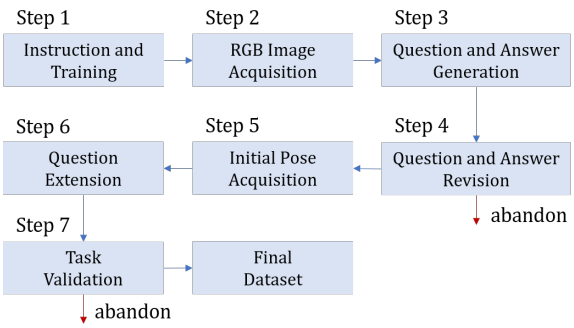

Figure 5: The collection and validation process of the CityEQA dataset.

Step 1. All the objects mentioned in the question are extracted, along with the spatial relationships between them. Each object is assigned a unique identifier to ensure distinction. Additionally, the state of each object is marked as `Unknown` as their locations remain uncertain. The agent itself is treated as a special object, with its state marked as `Known`, allowing it to serve as a unique initial landmark.

Step 2. The information necessary to answer the question is extracted as corresponding information requirements. This step forms the purpose for the following plan generation, as the entire perception process is driven by the need to gather this critical information.

Step 3. An executable plan is formulated by combining three types predefined sub-tasks based on information requirements. To guide LLMs reasoning and constructing an executable plan, we establish a set of simple rules. First, collecting information requires the Collection sub-task. However, before executing this sub-task, the states of the relevant objects must be `Known`, meaning the objects must already have been located in the environment. Second, the Exploration sub-task can transition an object's state from `Unknown` to `Known`. Third, before performing Exploration, the Navigation sub-task can be employed to leverage a `Known` object as the landmark, enabling the agent to efficiently reach specific locations. This sub-task can

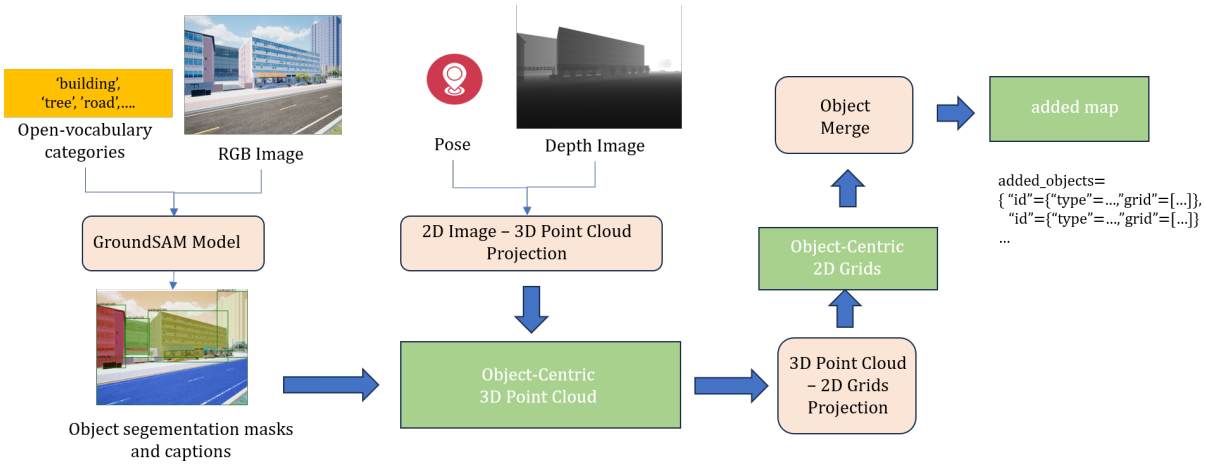

Figure 6: The workflow of the construction of the added map.

reduce the exploration scope and enhances overall efficiency.

Details of Object-Centric Cognitive Map The processing procedure of the function $Construct()$ is illustrated in Figure 6. Firstly, the GroundSAM model is utilized to process the RGB image to obtain object segmentation masks and captions. Meanwhile, the pose and depth image are combined with the camera intrinsic parameters to obtain 3D point cloud data. Then, these two data are merged to obtain the object-centric 3D point cloud. Further, this data is projected onto a 2D grid, and the point cloud data outside the map range is filtered out to obtain the object-centric 2D grids. Finally, objects with repetitive grids are fused to obtain the object-centric added map.

The purpose of the function $Merge()$ is to fuse the added objects in added map into the global map. This is to ensure that the same object observed from different views is uniquely recorded and retrieved on the map. Therefore, for each added object, we first determine whether the distribution of the object overlaps or is adjacent to any object in the global map. If so, the two objects are merged; if not, the object is directly added to the global map. This paper adopts a simple and effective strategy to determine whether objects are adjacent: when at least one pair of grids in which the two objects are distributed are adjacent, they are considered to have an adjacent relationship. Additionally, it should be noted that multiple object merges may occur in the same round, so the merged object needs to be judged against all other objects in the global map in another round.

Details of Collector The prompt provided for MM-LLM in Collector is presented in Figure 7. The Collector needs to complete two tasks in sequence. The first is the VQA task, which involves answering the corresponding questions based on the provided RGB image. The second is action selection, which requires choosing an appropriate action from a discrete set of actions to adjust the observation. The action set used in this study includes {*MoveForward*, *MoveBack*, *MoveLeft*, *MoveRight*, *MoveUp*, *MoveDown*, *TurnLeft*, *TurnRight*, *KeepStill*}.

A.3 Experiments Details

LLM Scoring For QAA, we designed an LLM-based automated scoring method by referring to the LLM-Match mechanism in OpenEQA (Majumdar et al., 2024). We show the designed prompt for LLM in Figure 8.

To investigate the validation of using the LLM as judge, a double blind study is conducted. We randomly sampled 100 answers from the results including the answer generated by the 4 baselines and PMA. Then 2 human evaluators are required to provide their score of the answers while using the prompt in Figure 8 as the task instruction. Since the distribution of scores did not conform to a normal distribution, Spearman's correlation analysis was adopted. The results indicated a significant positive correlation between the scores given by human evaluators and those by LLM judges (R_s = 0.85, p = 0.002). This suggests that using LLMs as judges can effectively evaluate open-ended question-answering results and align well with human judgments.

You are an autonomous UAV (Unmanned Aerial Vehicle) tasked with performing visual perception operations in an urban environment.
For each step, you will receive the following inputs:
-Image: An RGB image representing your current view.
-Question: A query requiring specific information to be extracted from the Image.
-Reference answer: An answer generated during the previous step.

Your mission consists of completing the following two tasks in sequence:

Task 1: Visual Q&A
Analyze the content of the current Image and provide a concise and meaningful answer to the Question.
Guidelines:
-If the image is insufficient to answer the Question, use reasoning and common sense to guess an answer.
-Your answer must be meaningful and informative. Avoid vague responses like "It is not legible/visible..." or "It is not possible to determine...".
-Provide a concise response without including explanations, reasoning, or thought processes.
-Compare your answer to the Reference Answer and select the better one as your final answer.
-Do not consider Task 2 until you have completed Task 1.

Task 2: Action Selection
Please, select one action from the following 9 actions
...

Guidelines:
-Analyze the drawback of the current image, such as occlusion, sidelong view, too far away, etc., and then select the appropriate action to adjust you view to obtain a better image.
-Think this step is your last step to adjust view, so choose the most urgent action.
-If the object mentioned in the question is on the edge of the image, you can use a TurnLeft or a TurnRight to make the object fully appear in the image.
-Keep the current view if the answer is clear and confident.
-Use TurnLeft or TurnRight to look around if the current image does not contain the answer.

Figure 7: The prompt used for Collector.

You are an AI assistant who will help me to evaluate the response given the question and the correct answer. To mark a response, you should output a single integer between 1 and 5 (including 1, 5).5 means that the response perfectly matches the answer.1 means that the response is completely different from the answer, or the answer is meaningless, such as "It's not possible to determine..."

Output format:
{
  "mark": <integer>
}

Example 1:
Question: What's the name of the shop to the left of the supermarket?
Answer: Starbucks
Response: Starbuks
Output:
{
  "mark": 4
}

Example 2:
  ……

Your Turn:
Question: {question}
Answer: {answer}
Response: {prediction}

Figure 8: The prompt used for LLM scoring.

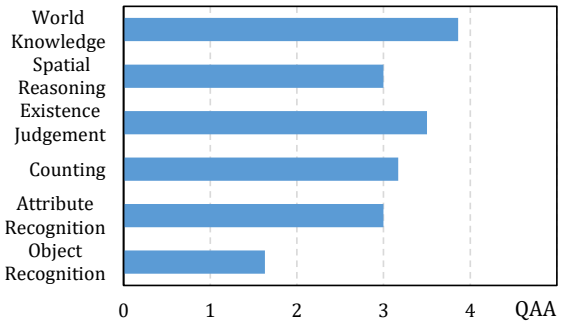

Figure 9: Categroy-level performance of the proposed PMA.

Baselines Details This section provides additional details for the baselines.

- **Blind Agents.** We choose four State-of-the-Art LLMs as blind agents, including GPT-4, Qwen-2.5, LLaMA-v3.1-8b, and DeepSeek-v3. They generate the answer purely based on the question, formulated as $\hat{y} = \text{LLMs}(q)$.

- **Socratic Agents.** We sample efficient trajectories generated by H-EQA to simulate the observations available to Socratic Agents. Specifically, we select the last five frames from each trajectory and use GPT-4o to generate image captions C. Different LLMs—including GPT-4, Qwen-2.5, LLaMA-v3.1-8b, and DeepSeek-v3—are then used to produce the final answers, formulated as $\hat{y} = \text{LLMs}(q, C)$.

- **VQA Agents.** They have direct access to images containing the answers. We use GPT-4o, Qwen-2.5, and LLaVA-v1.5-7b as VLMs to generate answers based on the images and questions, formulated as $\hat{y} = \text{VLMs}(q, p^{obs})$.

- **Exploring Agents.** They are guided by different exploration strategies such as RE and FBE, and generate the answer based on the visual input I at the termination position, formulated as $\hat{y} = \text{VLMs}(q, I)$. RE randomly selects an action from {*MoveForward*, *TurnLeft*, *TurnRight*, *Stop*} at each step. The angles for *TurnLeft* and *TurnRight* are set at 30°, and the distance for *MoveForward* is 10 meters, consistent with the setting of the Navigator in the PMA. FBE identifies the frontiers between explored and unexplored regions, samples one as the navigation point, and employs the A* algorithm to find a path. The maximum path length is also limited as 10 meters. To avoid excessive exploration, GPT-4o is employed to decide when to stop.

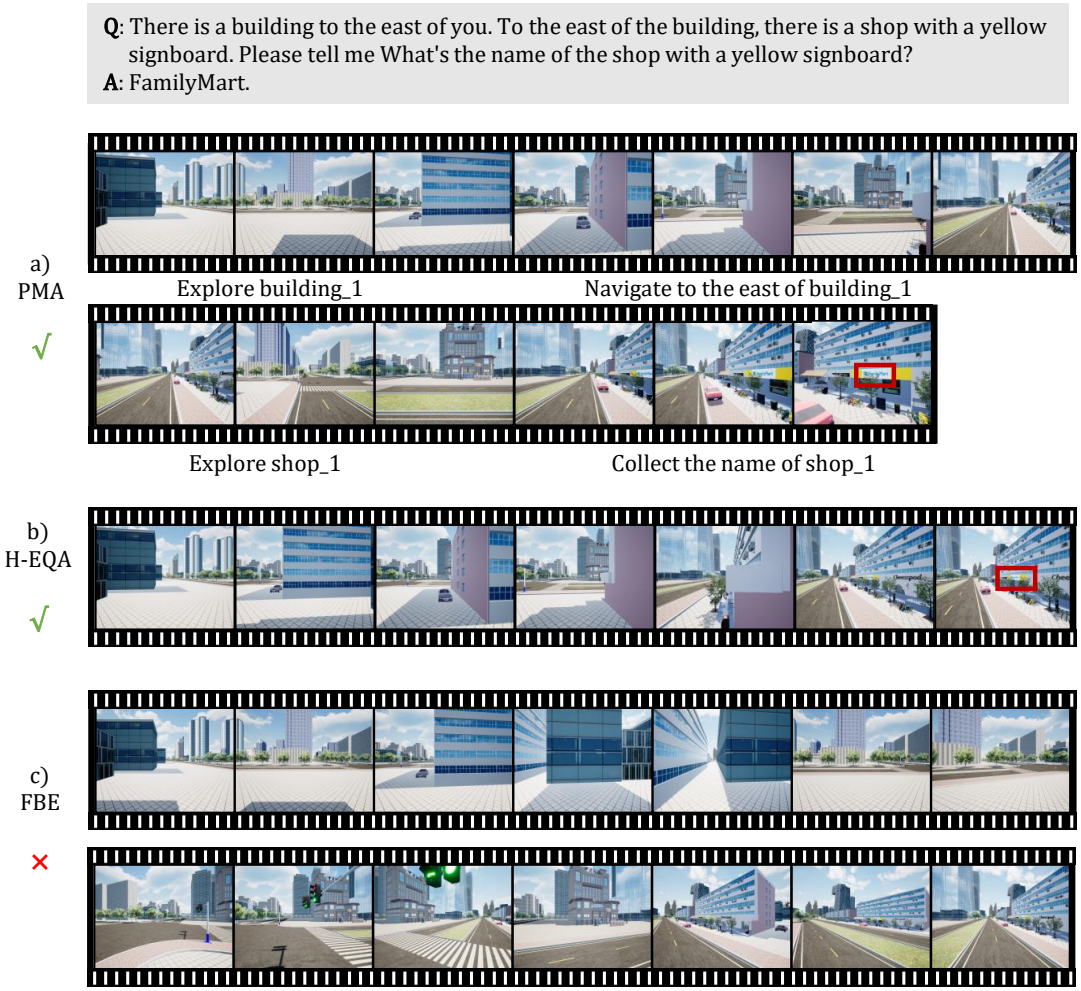

Figure 10: Examples of different EQA methods.

- **Human Agents.** At each step, H-EQA can only access the RGB image of the current pose and must choose one action from {*MoveForward*, *TurnLeft*, *TurnRight*, *Stop*}. The angles for *TurnLeft* and *TurnRight* are set at 30°. When selecting *MoveForward*, the agent must also provide an integer distance within 10 meters. When choosing *Stop*, the H-EQA is required to provide the answer.

Categroy-level performance of the PMA The category-level performance of the proposed PMA is shown in Figure 9, and it varies across task types. PMA achieves the highest QAA on World Knowledge tasks, likely because these tasks rely partially on the LLM's inherent knowledge and require minimal visual inputs. However, it performs the worst on Object Recognition tasks due to their open-ended answers and greater reliance on visual inputs.

Comparison between different EQA methods We present the trajectories of PMA, H-EQA, and FBE to illustrate the different strategies adopted by them when searching for the answer to the same question, as shown in Figure 10. PMA finds the answer by decomposing the perception process into several sub-tasks and completing them step by step. H-EQA, with its stronger visual understanding and spatial reasoning abilities, can locate the answer in fewer steps. Moreover, H-EQA is often able to determine the answer from a greater distance, likely due to its extensive world knowledge, which allows it to fill in missing information even with incomplete observations. In contrast, FBE, lacking the ability to utilize landmarks such as building_1 and shop_1, can only fully explore the environment, resulting in lower perception efficiency. This highlights the differences between performing EQA tasks in urban spaces versus indoor environments.

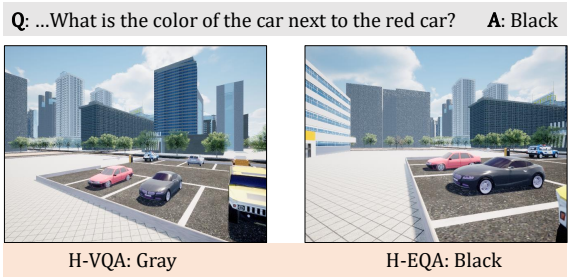

Figure 11: Examples of the H-VQA and H-EQA.

Comparison among Human Agents In Figure 11, we provide a case to illustrate why the performance of H-EQA is superior to that of HVQA. The given question is "What is the color of the car next to the red car?" The ground truth answer is "Black". HVQA was provided with the RGB image on the left for question answering. However, in this image, due to the influence of outdoor lighting, the originally black car appears gray, thus H-VQA provided an incorrect answer. In contrast, H-EQA can actively adjust the observation pose, observing the side of the car to reduce the impact of the lighting, and thereby providing the correct answer.

Analysis of Collector's action The statistics of various actions taken by Collector are shown in Figure 12. Besides, we present two cases to illustrate the effect of the collector. In the first case, as shown in Figure 13 (a), since the shop with black signboard was discovered too early in the Exploration stage, the starting pose of the collector was far from the target pose. Even after moving 10 steps promptly, it still failed to recognize the text on the black signboard. In the second case, as shown in Figure 13 (b), the yellow signboard that the collector needed to recognize was on the left side of the picture and seemed not to be fully displayed. At this time, the collector took the *TurnLeft* action, thus observing the entire yellow signboard and easily providing the correct answer.

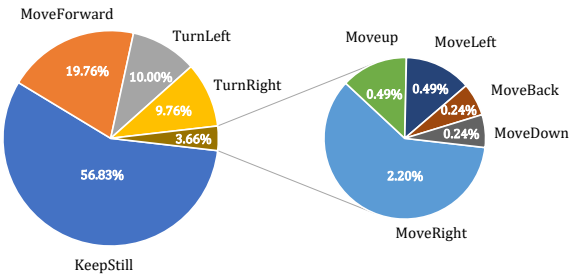

Figure 12: The proportion of different actions taken by Collector.

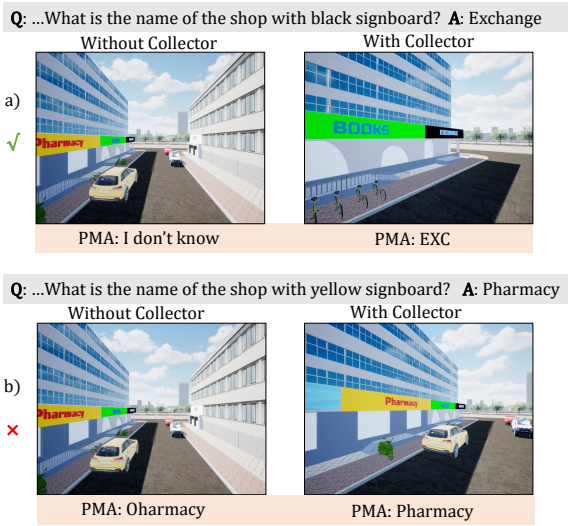

Figure 13: Examples of the Collection phase.