A Visual Target Navigation Method for Quadcopter Based on Large Language Model in Unknown Environment

Yunzhuo Liu †, Zhaowei Ma †, Jiankun Guo, Haozhe Sun, Yifeng Niu *, Hong Zhang, Mengyun Wang

Posted Date: 14 December 2025

doi: 10.20944/preprints202512.1129.v1

Keywords: visual target navigation; UAV; large language model

 doi:10.20944/preprints202512.1129.v1

Article

A Visual Target Navigation Method for Quadcopter Based on Large Language Model in Unknown Environment

Yunzhuo Liu †, Zhaowei Ma †, Jiakun Guo, Haozhe Sun, Yifeng Niu *, Hong Zhang and Mengyun Wang

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China
* Correspondence: niuyifeng@nudt.edu.com
† These authors contributed equally to this work.

Abstract

This paper proposes a novel large language model (LLM)-based approach for visual target navigation in unmanned aerial systems (UAS). By leveraging the exceptional language comprehension capabilities and extensive prior knowledge of LLM, our method significantly enhances unmanned aerial vehicles (UAVs) in interpreting natural language instructions and conducting autonomous exploration in unknown environments. To equip the UAV with planning capabilities, this study interacts with LLM and designs specialized prompt templates, thereby developing the intelligent planner module for the UAV. First, the intelligent planner derives the optimal location search sequence in unknown environments through probabilistic inference.Second, visual observation results are fused with prior probabilities and scene relevance metrics generated by LLM to dynamically generate detailed sub-goal waypoints. Finally, the UAV executes progressive target search via path planning algorithms until the target is successfully localized. Both simulation and physical flight experiments validate that this method exhibits excellent performance in addressing UAV visual navigation challenges, and demonstrates significant advantages in terms of search efficiency and success rate.

Keywords: visual target navigation; UAV; large language model

1. Introduction

Compared to ground robots, quadrotor drones offer distinct advantages in conducting search missions in unknown environments, including rapid response, andsuperior maneuverability. These unmanned aerial vehicles (UAVs) typically rely on onboard sensors (e.g., cameras) for environmental perception in navigation scene. When deployed to explore hazardous unknown areas, such UAVs require explicit instructions defining search tasks, such as target objects and designated destinations. However, enabling UAVs to locate arbitrarily specified targets based on human instructions in unknown environments remains a significant challenge. This challenge fundamentally corresponds to the classical visual target navigation problem, which necessitates that UAVs possess autonomous search capabilities integrated with advanced language comprehension and semantic reasoning skills [1,2].

Humans, by comparison, demonstrate remarkable superiority in searching unknown indoor environments. Leveraging rich prior knowledge of the physical world and robust logical reasoning capabilities, humans can quickly identify probable target locations and develop efficient search strategies. For example, when searching for specific items, humans prioritize relevant areas (e.g., refrigerators in kitchens or sofas in living rooms), which significantly reduces search time—an ability rooted in a profound understanding of environments and commonsense knowledge regarding object distribution. During navigation to target areas, humans perform scene matching via visual observations to confirm arrival before conducting localized searches. If the target remains undetected, they proceed to the next most probable location until the search is successful.

 doi:10.20944/preprints202512.1129.v1

In the current UAV field, conventional approaches for achieving exploration in unknown environments typically construct environmental maps from visual observations prior to planning navigation routes based on geometric representations [3]. Owing to the absence of prior scene knowledge, such methods often entail extensive area-wide searches. In contrast, learning-based methods—designed to emulate human exploration of uncharted territories—leverage deep reinforcement learning (DRL) to directly optimize navigation strategies for locating targets of specified categories using real-time visual inputs [4]. While these learning-based approaches demonstrate superior performance in environmental exploration and target navigation once accurate target information is obtained, they are unable to fully exploit prior knowledge of the physical world to enhance search efficiency.

The emergence of large language models (LLMs) such as BERT [5], GPT-3 [6], and GPT-4 [7] has rendered this limitation addressable. These models not only demonstrate robust language comprehension capabilities to effectively interpret human intentions and generate contextually appropriate responses but also embed extensive prior knowledge of the physical world that provides valuable guidance for decision-making. Consequently, recent research has begun integrating LLMs with robotic systems—including UAVs—to enhance their intelligent autonomy: specifically in understanding complex human instructions for task planning [8], and in converting natural language-described targets into structured planning languages to generate executable solutions [9,10].

This study focuses on LLM-based autonomous decision-making for UAV planning in unknown environments, leveraging LLMs' inherent prior knowledge as directional guidance. Specifically, we first integrate vision-language models (VLMs) to generate real-time environmental descriptions, then adopt an LLM-driven active search decision-making mechanism to dynamically generate subtarget waypoints for UAV navigation. Our approach selects search directions using linguistic models and prior probabilities as knowledge bases, thereby reducing learning costs while enhancing the generalizability of scene prior knowledge.

Our contributions are summarized as follows:

1. We propose a hierarchical prompt engineering framework that systematically integrates task instructions, scene prior knowledge, role definitions, and demonstration examples into structured prompt templates. This design effectively guides LLMs in task comprehension and reliably meets downstream UAV navigation requirements.

2. We develop a novel active search decision - making approach that integrates dataset-derived prior probabilities with LLMs. By jointly leveraging these two knowledge sources for search direction selection, our method achieves efficient target search in unknown environments, and its effectiveness is further validated through physical experiments.

3. We propose a novel integration method that bridges LLMs with UAV visual target navigation tasks, addressing the long-standing gap between natural language understanding and autonomous UAV navigation. This method is elaborated in Section 3, and its effectiveness is experimentally demonstrated in Section 4.

2. Related Works

2.1. LLM for Robot Control

Natural language generation, a core capability of large language models (LLMs), is widely used in task planning for its interpretability. However, natural language's ambiguity can lead to unclear instructions, potentially causing task failures. To address this, Kim et al. [11] introduced a sub-goal planner that extracts key sub-goals and objects from high-level instructions, reducing ambiguity through hierarchical decomposition and external validation. Huang et al. [12] leveraged GPT-4 [7] and chain-of-thought reasoning to generate detailed plans for long-term tasks, utilizing LLMs' world knowledge and reasoning. Zhu et al. [13] proposed a dual-system approach, distinguishing fast thinking for simple actions (e.g., grasping) from slow thinking for complex reasoning (e.g., rearrangement), enhancing both responsiveness and execution quality.

 doi:10.20944/preprints202512.1129.v1

Large language models (LLMs) can generate planning formats ranging from structured Planning Domain Definition Language (PDDL) to flexible natural language. Structured formats like PDDL offer precision and interpretability, facilitating robot execution, while natural language provides adaptability but may require complex controllers for interpretation. Liu et al. [8] proposed converting planning problems into PDDL via LLMs, parsing them with a PDDL planner, and translating results back into natural language, enhancing interpretability and usability. Yang et al. [14] extended this by transforming parsed PDDL into finite state automata, creating a full planning-control loop. Zhou et al. [15] introduced an iterative optimization method, using LLMs to generate, score, and refine PDDL-based plans, improving solution quality and reliability through automated evaluation.

2.2. Visual Target Navigation

Recent studies have also applied LLMs to visual target navigation tasks, demonstrating their effectiveness in this context.Majumdar et al.[16] employed the BERT model to construct a scoring function between language instructions and paths, aiding embodied agents in navigation. Shah et al.[17] introduced LM-NAV, which extracts landmark names from instructions using a language model, matches these landmarks in the world through an image-language model, and then navigates to them using a navigation model.Xie et al.[18] improved the path generation of Rapidly-exploring Random Tree (RRT) by introducing LLM for semantic relevance measurement to achieve outdoor navigation for UAVs.Chen et al.[19] utilized the prior knowledge of LLM for semantic inference to obtain scene labels, achieving 3D scene understanding for robots.Yu et al.[20] introduced LLM into the problem of visual target navigation, enabling the use of language to find relevant boundaries from semantic maps as long-term goals and effectively explore the environment.

3. Proposed Method

3.1. Overview

To address the challenge of UAV visual target navigation guided by natural language instructions in unknown environments, we propose an active search behavior decision-making method. In this work, a large language model (LLM) serves as a search sequence planner to determine the order of exploration. The LLM's output combined with prior probabilistic knowledge and semantic relevance metrics is used to guide the UAV's search directions.

Figure 1 illustrates the overall framework of our approach. Consider a scenario where a UAV searches for a specified target inside a house. The user provides a natural language instruction containing the target name . Based on this instruction, the UAV generates a task plan to find the target: it first constructs an initial search sequence using prior map information and the target name, then autonomously selects search directions by fusing real-time visual observations with semantic guidance from the LLM. Corresponding sub-goal points are generated accordingly, and the UAV iteratively navigates toward them using a path planning algorithm until the target is located. Within this framework, there are two extremely important modules that support the autonomous object-seeking and exploration of UAV.

(1) Observation Module. To enable autonomous navigation in unknown environments, it is crucial to perceive raw environmental observations and extract critical information for downstream reasoning. This necessitates two key perceptual capabilities: scene recognition, which identifies objects within the scene, and mapping, which locates these objects to facilitate path planning.To achieve these capabilities, the framework integrates two locally deployed visual language models:

Recognize-anything Model [21]: This model constructs scene descriptions by identifying object names from RGB images.

GroundingDINO Model [22]: This model detects object names and their corresponding bounding boxes from RGB images and a list of object names.

The module leverages these models to obtain object names and bounding boxes, combining this information with depth images to calculate object positions through coordinate transformation. Its

 doi:10.20944/preprints202512.1129.v1

primary function is to construct a semantic map using RGB and depth images, recording discovered objects and their positions.

(2) Memory Module. This module is designed to store essential information and maintain data acquired within the scene. It manages both static and dynamic data: static information includes prompt templates and prior probability lists, while dynamically updated information encompasses the UAV's historical path data, semantic maps, and records of previously encountered objects.

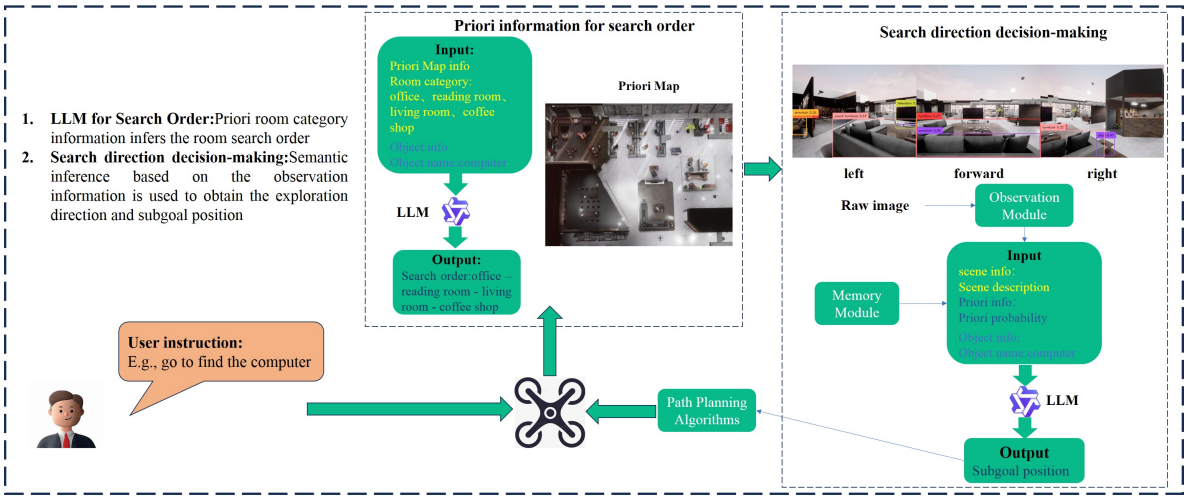

Figure 1. The framwork of the system.

3.2. Priori Information for Search Order

This section focuses on the decision-making problem of search sequence in unknown indoor environments. By leveraging the commonsense reasoning capabilities of large language models (LLMs), we guide the model to infer the most probable regions where the target object may be located and generate a priority-based search list.This approach enables the drone to prioritize searching high-probability regions, avoiding resource wastage from blind exploration. It provides a critical decision-making foundation for determining search directions and optimizing efficiency.

To effectively guide the large language model (LLM) in generating a realistic search sequence tailored to indoor search tasks, a structured prompt was designed. The prompt consists of two key elements: **role description** and **example description**, using a single-interaction mode without requiring a continuous dialogue mechanism. The prompt construction method is as follows:

(1) Role description design

The role description includes **task scenario definition** and **capability boundary** constraints, clearly specifying the drone system's responsibilities and decision-making authority.

1) Task scenario description

The task scenario serves as a critical foundation for prompt construction. By clarifying task requirements and contextual background, it provides a systematic execution framework for subsequent experiments. A comprehensive task scenario definition should include the following core elements:

$$Task\_Scene = \{R_{Des}, A_{Sce}\} \tag{1}$$

where R_{Des} is the role definition, which specifies the function of the large language model in the task, and A_{Sce} is the application scenario, which defines the specific experimental environment.

For the task requirement of generating global search sequences, the role is defined as a sequencing assistant; the application scenario focuses on indoor environments to evaluate the spatial distribution probability of search targets.

2) Capability boundary constraints

 doi:10.20944/preprints202512.1129.v1

These constraints guide the LLM's behavior, mitigate potential risks, and ensure system reliability. A complete capability boundary definition includes:

$$Capability\_Boundaries = \{O_{Per}, O_{Pro}\} \tag{2}$$

where O_{Per} is the permitted operation, which clearly defines the operation behaviors that can be invoked by the large language model when solving the target task, and ensures that all the operations comply with the system safety specification and task requirements. O_{Pro} is the prohibited operation, which strictly restricts the operation behaviors that may cause safety hazards.

For the task of generating a global search order, the permitted operation is limited to the use of predefined region labels in the input list, and the output result must be limited to the set of input region labels and include all the labels; the prohibition operation prohibits the generation of any region labels outside the list, the omission of the predefined region labels, or hypothetical reasoning beyond the description of the environment. This design not only fully utilizes the reasoning advantages of the large language model, but also effectively avoids its potential risk of "illusion".

(2) Example description design

The example description establishes the interaction paradigm between the user and the LLM. A complete interaction format includes:

$$Interaction\_Protocol = \{I_{Spe}, O_{For}\} \tag{3}$$

where I_{Spe} is the input specification, which explicitly defines the type of requirement contained in the user's instruction and ensures that there is no obvious ambiguity in the instruction, and O_{For} is the output format, which serves the task requirements.

For the task of generating global search sequences, the input specifications clearly define the format and definitions of input elements, enabling the large language model to distinguish between the target object names and search region label lists in the input components. The output format is designed for automated parsing, providing an easily extractable structure to guide the model's outputs while facilitating region label extraction and reducing the likelihood of program errors. An example output format is shown below:

Region Label 1 - Region Label 2 - ... - Region Label n

After generating the search sequence, the drone follows the priority order to scan each region. Within each region, the drone uses its visual perception module to identify the target object. If the target is not found, the drone proceeds to the next higher-priority region until the search is complete.

3.3. Search Direction Decision-Making

Upon obtaining the regional search sequence, the drone needs to determine the optimal moving direction based on current perceptual information until reaching the target location. To achieve this, this paper adopts two complementary methods to evaluate which direction within the current field of view is most likely to lead to the target area: data-driven prior probability analysis and semantic-understanding-based LLM reasoning.

(1) Prior probability analysis: Based on large-scale datasets, this method assigns prior probabilities to each region label through statistical methods, representing the likelihood of the target object appearing in that region. During the drone's search process, by detecting object categories in the current field of view and combining them with pre-calculated regional probability distributions, it can compute regional relevance scores for each direction.

(2) LLM reasoning: To compensate for the limitations of pure probability analysis in complex scenarios, this paper introduces LLM for semantic relevance assessment. By inputting detected object categories (such as "refrigerator", "sofa", "desk", etc.) from various directions, the LLM can determine the most probable corresponding region labels based on its rich semantic knowledge base, and output scores indicating the likelihood of target object presence. This method fully utilizes the

 doi:10.20944/preprints202512.1129.v1

LLM's deep understanding of semantic relationships between objects and scenes, effectively handling special scenarios or complex environments not covered in the dataset. By combining prior probability analysis with LLM reasoning, comprehensive evaluation of search directions can be achieved from both statistical and semantic understanding dimensions.

The pseudo-code of the search direction autonomous decision-making method is shown in the Algorithm 1. The method first determines the search order of the region based on the list of search targets and region names, and then acquires visible light images in different directions. In this paper, we mainly use the Recognize-anything model for target recognition. Based on the recognition results, the scene likelihood ranking and relevance score under the a priori probability and the scene likelihood ranking and relevance score generated by the large language model are calculated respectively. Finally, the results of the a priori probability and the large language model are fused to determine the prioritized search direction, and the subgoal point locations are computed and generated by Equation (4), which provide inputs for the subsequent local planning of the search path.

$$subgoal_p = cur_p + \lambda * sensor_r * \vec{D} \tag{4}$$

where cur_p is the current position, $subgoal_p$ is the subgoal position, λ is a fixed value, $sensor_r$ is the sensor sensing range, and $\vec{D}$ is the search direction vector per unit length.

Algorithm 1 Search Direction Decision-Making

input: $img = [img_0, img_1, img_2]$,$target\_name$,$x_{satrt}$
output: $search\_direction$
initialize p_adj,$area\_list$
$search\_order = llm\_pre\_order(area\_list, target\_name)$
for each img_i in img **do**
  $name\_list = recognize(img_i)$
end for
$p\_score, p\_order = P\_compute(name\_list)$
$l\_order = llm\_order(name\_list)$
$l\_score = llm\_score(name\_list)$
$f\_order, f\_score = fusion\_order(p\_score, p\_order, l\_order, l\_score)$
$search\_direction = direction_decide(f\_order, f\_score, search\_order)$
return $search\_direction$

3.3.1. Prior Probabilities for Search Direction

To quantify the correlation between region categories and target objects, this study measures their association by calculating the prior probability of objects appearing in specific regions. Specifically, based on a public dataset containing 3000 images and their recognition results [23], we constructed a "region category-object name" prior probability list. This list covers most common region names (e.g., kitchen, bedroom, living room) and object names (e.g., refrigerator, bed, sofa), and calculates the prior probability of object a appearing in region b, denoted as p(a | b), using statistical methods.

After obtaining scene descriptions from different directions, to reduce the impact of common objects (e.g., doors and windows) on scene recognition, this study adopts the information entropy filtering method proposed in [20] to calculate the entropy value of each object category. The objects with the highest information content are selected based on their entropy values, which are then used to compute the scene's posterior probability. The entropy calculation formula is as follows:

$$H_{o_i} = -\sum_{t_j \in L_T} p(t_j \mid o_i) \log p(t_j \mid o_i) \tag{5}$$

where $o_i \in L_o$ represents the target object category (e.g., refrigerator, bed), and $t_j \in L_T$ denotes the scene category (e.g., kitchen, bedroom). $p(t_j \mid o_i)$ indicates the conditional probability of target object o_i appearing in scene t_j, obtained through normalization processing based on current scene categories.

 doi:10.20944/preprints202512.1129.v1

As can be seen from Equation (5), objects with greater information content have lower entropy values H_{o_i}. The entropy H_{o_i} measures the distribution uncertainty of target object o_i across different scenes. A lower entropy value indicates higher concentration of the object in specific scenes and stronger scene discriminative power; conversely, higher entropy reflects more uniform distribution across different scenes and weaker scene differentiation.

Based on the above method, this paper selects several key objects with the lowest information entropy (i.e., strongest discriminative power) and calculates the likelihood scores for each scene category through their conditional probability distributions, namely the posterior probability:

$$P(t_j) = \prod_{k=1}^{K} p(t_j|o_k) \tag{6}$$

where $\{o_1, o_2, ..., o_K\}$ represents the set of key objects selected through information entropy filtering. Since the number of objects detected from different viewpoints may vary, the posterior probability $P(t_j)$ needs to be normalized using Equation (7) to obtain the final likelihood score $S(t_j)$. Based on these likelihood scores, the scene probability rankings for the three viewpoints are determined as po_1, po_2, and po_3.

$$S(t_j) = \frac{P(t_j)}{\sum_{t_i \in L_T} P(t_i)} \tag{7}$$

3.3.2. LLM for Search Direction

After obtaining the regional search order, the corresponding region labels for each direction are determined through scene matching and ranked based on matching scores. If the region labels differ, the next search direction is determined according to the predefined order; if they are the same, it is considered that the target region has been reached, and further distinction between directions is made using relevance scores (measuring the probability of the target object's presence). This process involves two key tasks: **region label matching score ranking** and **relevance score calculation**, leveraging the semantic understanding capabilities of large language models. Similar to Section 3.2, the prompt design consists solely of role descriptions and example descriptions.

(1) **Region label matching score ranking**

1) Role description design

(a) Task scenario description

The task scenario consists of two elements: role definition and application scenario. In this paper, the role definition is a sorting assistant; the application scenario is oriented toward scenario matching, evaluating the matching degree between scenario descriptions and regional tags.

(b) Capability boundary constraints

The capability boundary includes two elements: permitted operations and prohibited operations. In this paper, permitted operations are limited to using predefined regional tags from the input list, and the output results must be confined to the input set of regional tags while including all tags; prohibited operations forbid generating any regional tags outside the list or omitting predefined regional tags.

2) Example description design

The example description consists of two elements: input specifications and output format, clarifying the interaction form with the large language model. In this paper, the input specifications enhance the definition of input elements, enabling the large language model to distinguish between the scene description list and the region label list in the input elements of this method. The output format serves the automated parsing of downstream tasks, providing an output format that facilitates extraction. An example of the output format is as follows:

Direction n: Region Label 1 - Region Label 2 - ..... - Region Label n.

(2) **Relevance score calculation**

1) Role description design

 doi:10.20944/preprints202512.1129.v1

(a) Task scenario description

The task scenario consists of two elements: character definition and application scenario. The character definition is a scoring assistant; the application scenario is to measure the relevance level of the searched target to the current scenario. After the drone arrives at the area, when the region labels in each direction are determined to be consistent, this score is used to determine the drone's next search direction.

(b) Capability boundary constraints

The capability boundary includes two elements: permitted operations and prohibited operations. In this paper, permitted operations are limited to measurement by scoring only; prohibited operations forbid the use of any form of measurement other than scoring.

2) Example description design

The example description consists of two elements: input specifications and output format, which clarify the interaction form with the large language model. The input specifications enable the large language model to distinguish between the target object name and scene descriptions in different directions within the input elements of this method. The output format serves automated parsing, measured by a score between [0, 10]. An example of the output format is as follows:

Direction $n : x, x \in [0, 10]$.

From an implementation perspective, the two aforementioned issues can be combined into a single prompt to guide the large language model's reasoning. However, in practice, this approach presents significant problems. Firstly, the merged prompt may make it difficult for the large language model to handle both tasks simultaneously, often resulting in outputs that fail to meet requirements—either due to irregular formatting or incomplete answers. Secondly, complex prompts increase the model's reasoning burden, reducing the accuracy and consistency of its outputs. Therefore, it is preferable to address these two issues separately in the interaction. This step-by-step approach not only ensures that the large language model focuses on a single task, producing more standardized results, but also enhances the efficiency and reliability of the interaction. By sequentially obtaining match rankings and relevance scores, this information can be integrated more precisely, thereby providing more dependable support for subsequent decision-making.

3.3.3. Fusing Prior Probabilities with LLM for Search Direction

Both methods exhibit notable limitations when used independently. The prior probability-based approach has two key drawbacks: (1) Its predefined "region-object" probability coverage is limited, struggling with unfamiliar environments or uncertain targets; (2) It fails to distinguish between identical region labels, providing only categorical probabilities without direct object associations. The LLM-based method, while demonstrating superior semantic understanding, lacks quantitative discrimination for identical labels. Its relevance scores offer limited practical value due to weak correlation with region labels.

To address these limitations, we propose a fusion strategy: the prior probability method provides quantitative baselines, while the LLM method supplements judgment for uncovered scenarios. Crucially, this hybrid enables relevance-based navigation decisions. The fused scoring formula is:

$$s = a * area\_score\_p + b * area\_score\_l \tag{8}$$

where s is the fusion probability score, where a and b are constants, $area\_score\_p$ is the probability score $S(t_j)$ from Section 3.3.1, and $area\_score\_l$ is the ranking-based score from Section 3.3.2.

Algorithm 2 outlines the fusion strategy, which generates labeled regions and relevance scores for each direction. Inputs include prior probabilities and LLM-derived rankings with scores, while outputs are fused rankings and scores. If the prior and LLM labels match, they are adopted directly; otherwise, the highest-scoring fused label is selected. Since the LLM lacks explicit matching scores, higher ranks are assigned higher values. The fused results guide the search direction, with sub-goal points generated based on sensor range, advancing the search through local path planning.

 doi:10.20944/preprints202512.1129.v1

Algorithm 2 Fusion Strategy

input: $p\_score = [ps_0, ps_1, ps_2], p\_order = [po_0, po_1, po_2], l\_order = [lo_0, lo_1, lo_2], l\_score = [ls_0, ls_1, ls_2]$
output: $f\_order = [fo_0, fo_1, fo_2], f\_score = [fs_0, fs_1, fs_2]$
initialize $area\_list, area\_score\_p, area\_score\_l, area\_score\_f, a, b$
$area\_score\_p = index(area\_list, p\_score)$
$area\_score\_l = assign\_score(area\_list, l\_order)$
for $i = 0 : 2$ **do**
  if $same(po_{i0}, lo_{i0})$ **then**
    $fo_i = po_{i0}$
    $fs_i = lo_{i0}$
  else
    for each s_i in $area\_score\_f$ **do**
      $s_i = a * area\_score\_p_i + b * area\_score\_l_i$
    end for
    $fo_i = max\_score(area\_list, area\_score\_f)$
    $fs_i = indx(fo_i, lo_i)$
  end if
end for
return f_order, f_score

4. Experiments

To evaluate the visual-language navigation method based on large language models proposed in this study, we initially validated the feasibility of our approach through simulation experiments conducted in the AirSim environment. Subsequently, we constructed a physical flight scenario and performed further experiments using an actual drone.

4.1. Airsim-Based Simulation Experiments

We conducted simulations using AirSim, an open-source, cross-platform simulator based on Unreal Engine. Supporting both software- and hardware-in-the-loop simulations with PX4/ArduPilot, AirSim excels in physical and visual simulations for robots like drones, leveraging game-engine rendering for superior visual effects. Our study employs a simulated drone, as shown in Figure 2.

Figure 2. Main view and top view of the simulation system UAV. The left figure is the main view and the right figure is the top view.

Thanks to the extensive environmental resource library of Unreal Engine, AirSim offers a variety of simulation scenarios. For this paper, we selected an indoor simulation environment, as illustrated in Figure 3.

 doi:10.20944/preprints202512.1129.v1

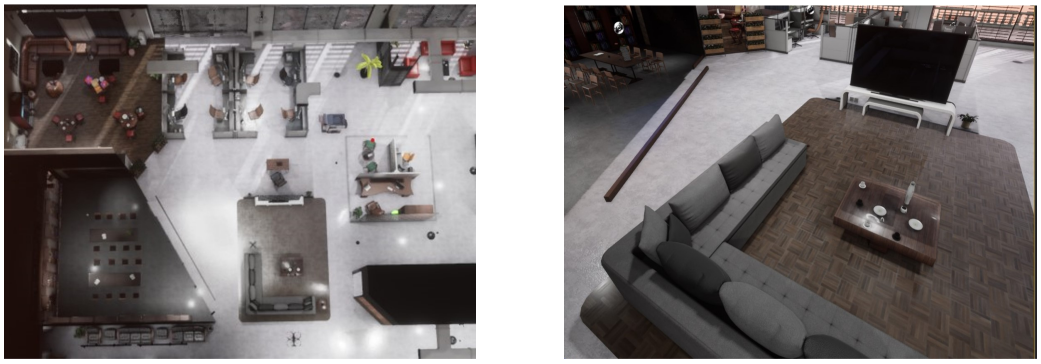

Figure 3. Simulation environment.

The Figure 4 shows an example of searching for a computer without location hint.Based on prior room category information, a pre-determined search order is generated.Then, based on visual observations, the degree of matching with room categories is used to generate sub-goal points.Through path planning algorithms, the UAV gradually approaches and finally find the target.

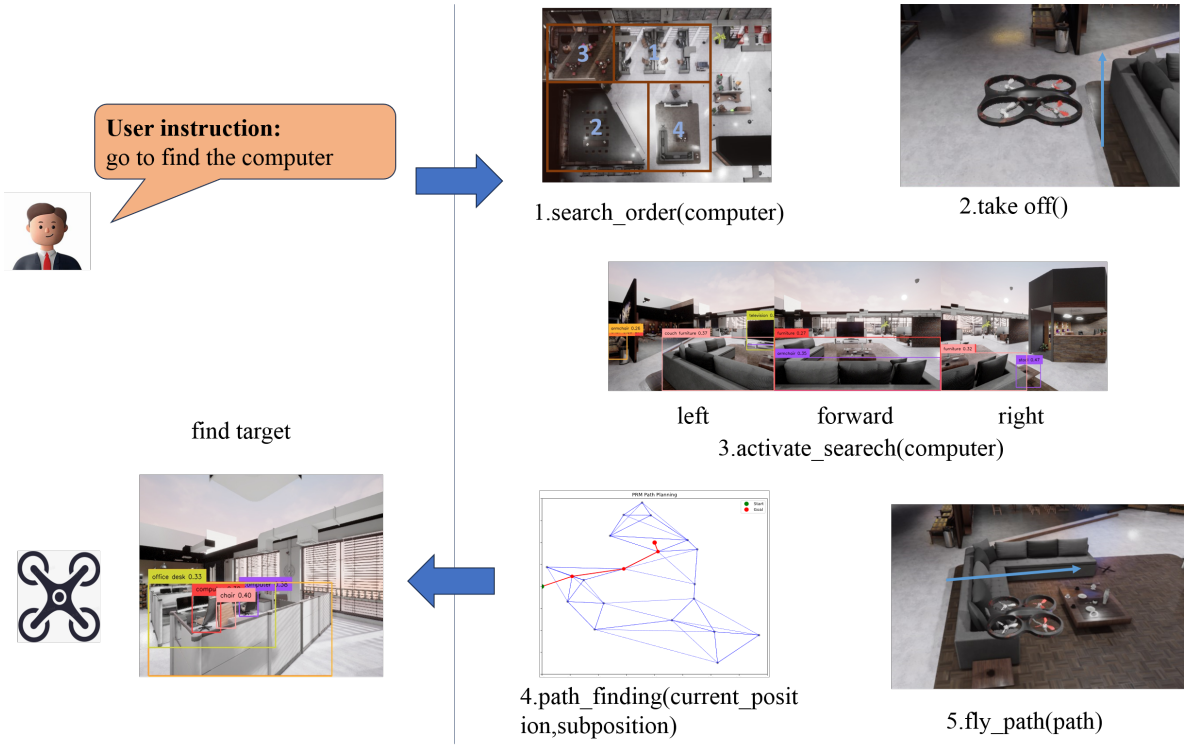

Figure 4. Example of searching for cumputer.

Due to the occasional instability in the output of LLM, which can lead to undefined or unexecutable generated code, and an increasing error rate as task complexity increases, we evaluate performance using both code executability and task success rates.To more comprehensively evaluate task performance, we have introduced two new evaluation metrics: Success weighted by Path Length (SPL) and Distance to Goal (DTG) at the end of the task. The definition of SPL is as follows [24]:

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

where N is the total number of experiments, S_i indicates whether the i-th experiment was successful (1 for success, 0 for failure), l_i is the Euclidean distance between the starting position and the success position in the i-th experiment, and p_i is the actual trajectory length of the i-th experiment. It is

 doi:10.20944/preprints202512.1129.v1

important to note that we have adjusted the definition of l_i in this paper. In the original definition, l_i was the shortest trajectory length between the starting position and the success position. However, in real-world scenarios, obstacles may exist between the two points, making it difficult to calculate the shortest path. Therefore, we use the Euclidean distance as a substitute. The DTG is calculated as the average Euclidean distance between the drone and the target at the end of the task:

$$\frac{1}{N}\sum_{i=1}^{N} D_i \tag{10}$$

where N is the total number of successful experiments, and D_i denotes the Euclidean distance between the UAV and the target position at the end of the i-th experiment.

The task instruction was "go to find the X," where X represents an object in the experimental scene. Each method was tested 25 times, and the results are shown in Table 1. The fused method achieved approximately a 20% higher success rate compared to the large language model (LLM)-based and prior probability-based approaches, demonstrating its effectiveness. Additionally, the SPL and DTG metrics indicate that the fused method provides better search-direction decisions, resulting in shorter paths upon task success and closer proximity to the target.The results of the method from [25] were similar to those of the LLM-based approach, primarily because both rely on semantic reasoning via LLMs to estimate the likelihood of the target object appearing in a given scene. The key difference lies in prompt construction, leading to only marginal variations in experimental outcomes.

Table 1. Performance comparison of the fusion method.

Method	Success rate	SPL	DTG(m)
Qwen2+Prior probabilities(Proposed)	**60%**	**0.3816**	**2.4618**
Qwen2	44%	0.2750	2.9901
Prior probabilities	36%	0.2196	2.7473
Ref [25]	40%	0.2752	2.8949

To investigate the impact of large language models on the method, this study compared two different models: Alibaba's Qwen2.5-VL and Baidu's ERNIE-3.5. Note that since ERNIE-3.5 is primarily designed for Chinese applications, Chinese prompts were used during interaction. Accordingly, all prompt content was translated into Chinese, and the output of the perception module was also adapted to Chinese. Each method was tested 25 times, with the experimental results shown in Table 2.

Table 2. Success Rates with Different Language Models.

Model	Success Rate	SPL	DTG (m)
Qwen2	60%	0.3816	2.4618
Qwen2.5-VL	**64%**	**0.3859**	**2.2545**
ERNIE-3.5	52%	0.3064	2.6239

The results show that Qwen2 and Qwen2.5-VL performed similarly, with the latter slightly outperforming the former. In comparison, ERNIE-3.5's results were slightly worse. Analysis revealed that this difference stems from variations in the ranking outputs between the Qwen series and ERNIE-3.5, leading to different subgoal point generation under the same conditions. Since individual decisions cannot be directly evaluated, only the overall task results could be analyzed. Thus, when using Chinese prompts, ERNIE-3.5 performed slightly worse than the Qwen series with English prompts. This discrepancy may arise from semantic deviations during translation, affecting decision accuracy, or from differences in model capabilities. The semantic deviations mainly originate from variations in

 doi:10.20944/preprints202512.1129.v1

object name translations between Chinese and English. For example, the Chinese word is a general term, while English distinguishes between "desk" and "table," leading to inconsistent scene inferences and thus affecting decision outcomes.

4.2. Physical Flight Experiments

4.2.1. Customized Indoor Scenes

The actual UAV built for the experiments in this paper is shown in Figure 5, with a wheelbase of 250 mm in length, a height of 298 mm, and a weight of 0.984 kg (without battery).

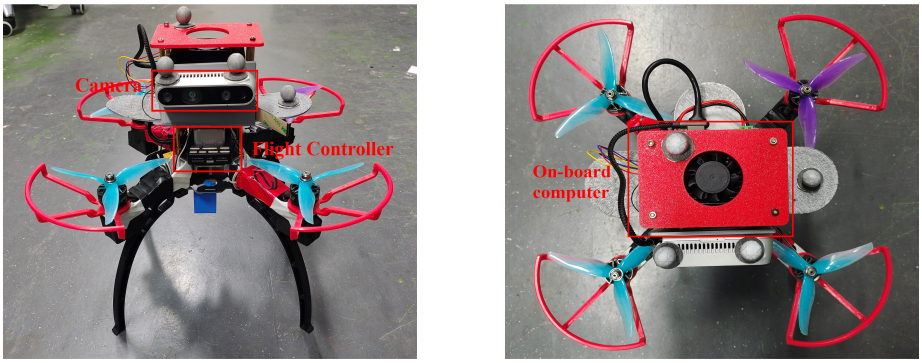

Figure 5. Physical drone hardware configuration.

The UAV is equipped with CUAV's V6X flight control module, offering high stability and scalability, along with a high-performance processor and precision sensors. The algorithm module runs on a Jetson Orin NX host (RTSO-3002 carrier board), an ARM-based system known for its low power consumption, compact design, high computational power, and developer-friendly features. Prior to use, the host requires system flashing with Jetpack 5.1.2, compatible with Ubuntu 20.04 and Python 3.8, meeting the minimum requirements for large model operations. For imaging, the Intel Realsense D435i camera is employed, providing high-precision depth data and high-resolution color images in a compact, energy-efficient design. It features a USB-C 3.1 interface for fast data transfer, fulfilling all experimental needs.

In the real experiments, the Nokov system was used to achieve the positioning of the UAV. Due to the use of a motion capture system for positioning, the flight experiment area must remain within the camera's field of view. Consequently, various indoor scenarios were constructed within the motion capture system's experimental area. Indoor environments often contain numerous objects, some of which are difficult to incorporate due to their complexity. To address this, virtual background images were used as substitutes, enhancing the richness of the physical experiment scenarios and providing additional contextual information. For this study, four indoor scenarios were created, representing a kitchen, a reading room, an office, and a bathroom. Small physical objects and virtual background images were used to construct these environments, with walls (1.8m high and 0.6m thick) installed to separate the scenes. Two layouts were designed for the physical experiments, as shown in Figure 6, arranging the four rooms in a cross layout and a T-shaped layout, while maintaining consistent interior arrangements across all rooms.

The experiment utilized Alibaba's Qwen2 model for human-machine interaction. Due to the model's substantial size making local deployment impractical, we established a local area network via hotspot to enable remote API access from the onboard computer for Qwen2 interaction.The workflow, shown in Figure 7, involves launching two terminals for text instruction reception and LLM interaction. A task instruction, such as "go to find the coffee cup", is entered and published to the ROS topic $/input\_stream$. The LLM interaction module subscribes to this topic, generates a prompt, and sends it to Qwen-2. Upon receiving a response, it publishes long-term control commands to $/control\_cmd$. The task management module executes these commands, prioritizing areas based on the likelihood of containing the target item and using a path planning algorithm to search each area. For example, the

 doi:10.20944/preprints202512.1129.v1

drone first searches the kitchen, then moves to the office, where it successfully locates the coffee cup, ending the task.

Figure 6. Physical experiment scene layout. Cross layout on the left, T-shaped layout on the right.

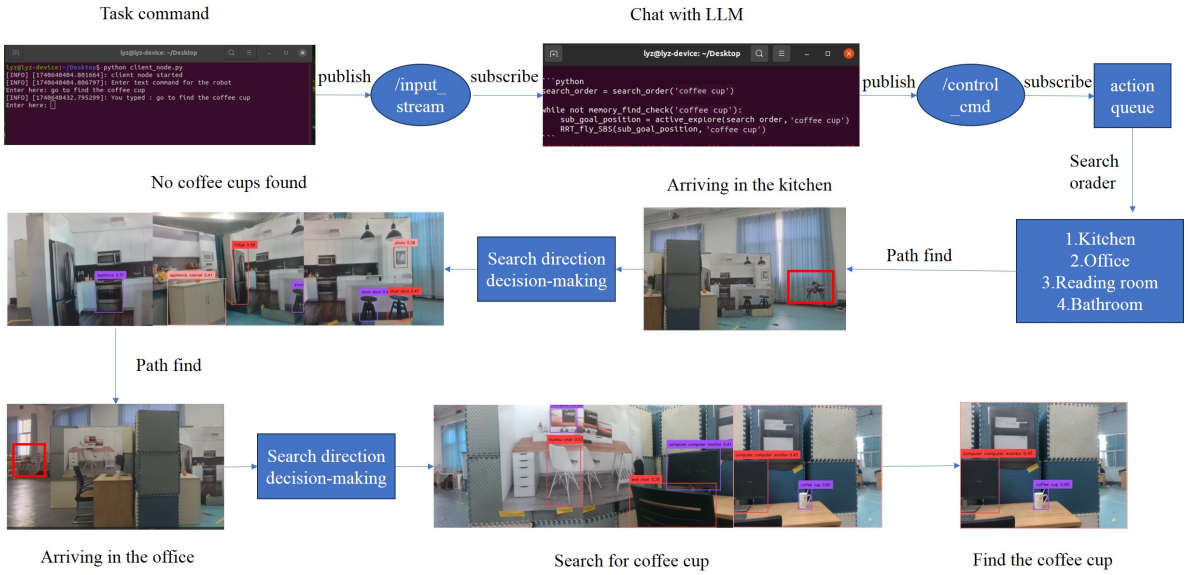

Figure 7. Schematic of finding a coffee cup in the cross layout scenario.

Following the outlined experimental flight procedure, this study conducted 25 target search trials for each of the two layout configurations, using common indoor objects including coffee cups, oranges, and computers as search targets. As depicted in Figure 8, the schematic diagrams present the coffee cup search experiments for both layouts. Each schematic provides comprehensive flight path visualization, explicitly indicating the actual flight trajectory, reference path planning, and target object location. Additionally, the diagrams incorporate critical decision-making points throughout the search process, categorized into three primary components: 1) generation of regional search sequence, 2) autonomous determination of search direction, and 3) termination criteria evaluation.

Based on the aforementioned experimental procedure, multiple flight experiments were conducted to search for different items within the scenarios. This study evaluates the flight experiments using three metrics: success rate, Success weighted by Path Length (SPL), and Distance to Goal (DTG). The experimental results are presented in Table 3. The success rate is determined by whether the target object is identified during the flight. SPL, which is strongly correlated with the success rate, is also influenced by the flight distance. DTG, independent of success, measures the distance between the drone and the target object when the task execution stops. In our flight experiments, the success rate is affected by the accuracy of object recognition and the ability to avoid obstacles. Instances of collisions with walls or unstable drone posture during recognition, which degrade image quality, can lead to task interruptions or failure to identify the target. SPL, on the other hand, is more influenced by the

 doi:10.20944/preprints202512.1129.v1

search order. For example, as shown in Figure 7, certain search orders can result in longer flight paths. Both layouts achieve a certain level of success rate, but compared to the cross layout, the T-shaped layout shows a noticeable advantage in SPL, primarily due to its shorter flight paths in most cases.

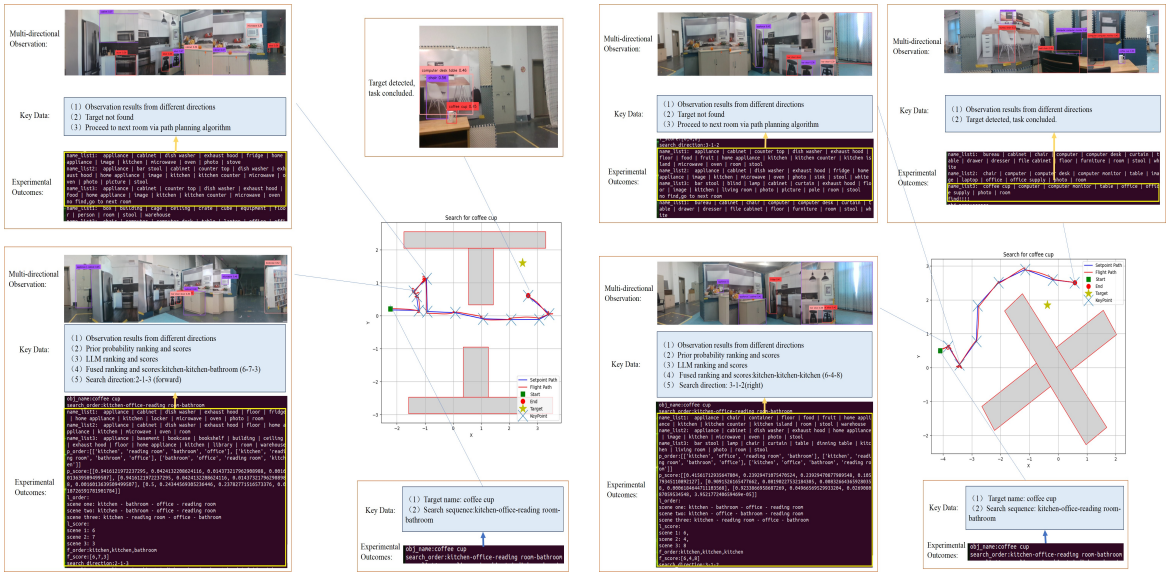

Figure 8. The schematic diagrams of the coffee cup in cross layout and T-shaped layout.

Table 3. Physical Flight Experiment Result.

Layout	Success Rate	SPL	DTG
Cross	**56%**	0.3069	1.7741
T-shaped	48%	**0.3782**	**1.6327**

The success rate in our flight experiments was constrained by the following factors:

(1)Decision Program Anomalies: The large language model occasionally generated abnormal navigation decisions beyond predefined area labels.

(2)Small-Target Detection Limits: When distant and small targets (e.g., oranges) were undetected by the perception module, premature region-completion decisions misled the drone to the next area.

(3)Localization System Failures: Near walls, motion capture system occlusion caused signal degradation or loss, risking collisions.

Additionally, Success-weighted Path Length (SPL) was more sensitive to search sequence. As seen in Figure 8, suboptimal search decisions led to longer paths. The T-layout generally achieved better SPL than the cross-layout, primarily due to its inherently shorter flight paths.

4.2.2. Indoor Structured Environment

This paper conducts additional physical flight experiments in an indoor structured environment, which excludes virtual backgrounds and consists entirely of real objects. The actual scene and its schematic diagram are shown in the Figure 9.The experimental environment comprises two distinct zones: a laboratory area and an office area. The laboratory area primarily contains various tools for assembling drone hardware platforms, while the office area is furnished with typical work supplies such as computers and keyboards.

Since the indoor environment lacks a motion capture system for positioning, we integrated a MID-360 LiDAR as the primary sensor for localization data acquisition. Figure 10 shows the modified UAV configuration with the added LiDAR module.For localization, we employed the Fast-LIO algorithm [26], which was specifically configured for compatibility with this LiDAR system.

 doi:10.20944/preprints202512.1129.v1

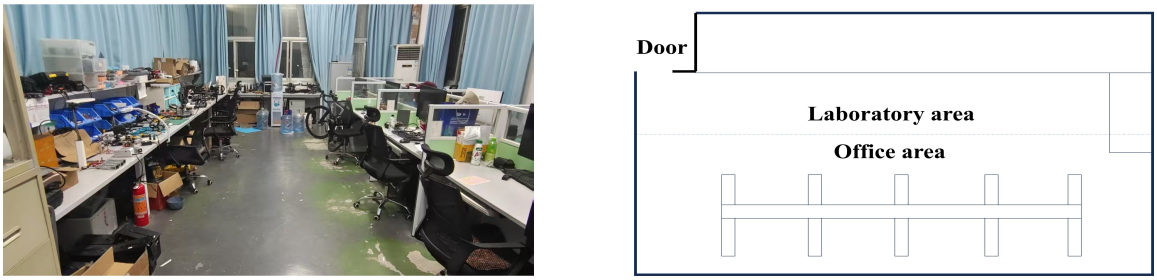

Figure 9. Indoor structured scene layout.

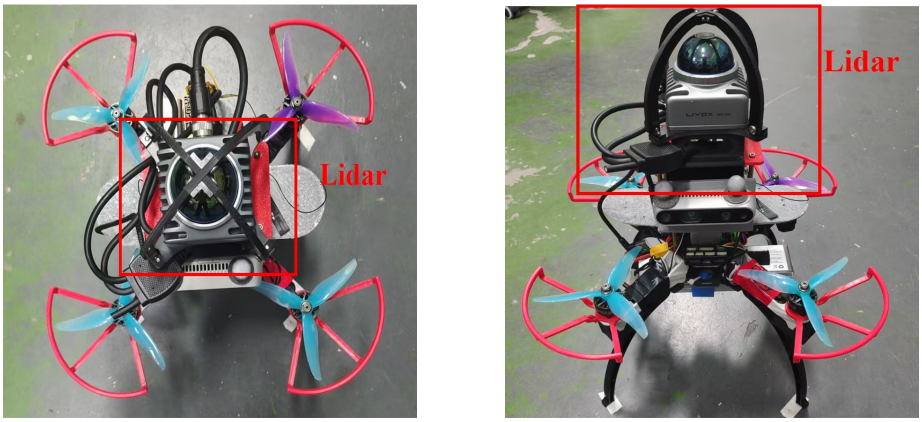

Figure 10. Quadrotor UAV test platform with integrated LiDAR module.

The flight procedure for this task is essentially consistent with that used in the indoor customized environment. Based on this experimental setup, flight tests were conducted to locate a keyboard in the office area and a drilling machine in the testing area. The figures respectively illustrate schematic diagrams of the search operations for the keyboard and drilling machine in the structured indoor environment. Elements in Figure 11 maintain consistency with those in the indoor customized scenario.

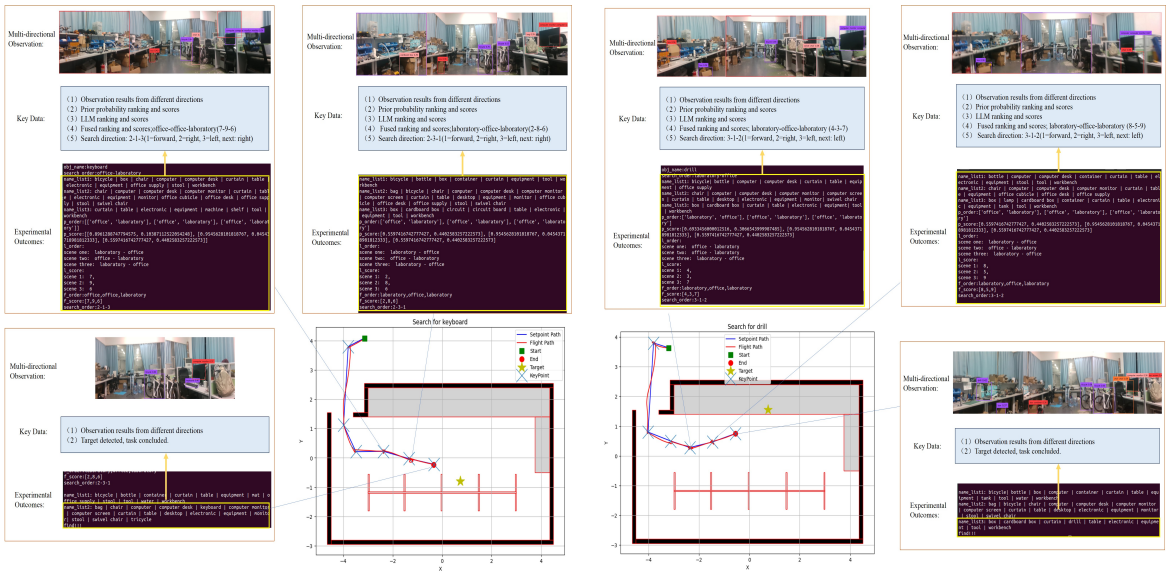

Figure 11. Flight path diagrams in cross layout and T-shaped layout.

5. Conclusions

This paper presents a novel framework that integrates large language models (LLMs) with visual target navigation tasks. Specifically, our research investigates how LLMs can serve as pre-planners in

 doi:10.20944/preprints202512.1129.v1

scenarios lacking explicit positional cues, demonstrating their capability to accurately interpret task contexts and generate predefined search sequences for exploration guidance.The proposed method dynamically generates sub-target waypoints by synergistically combining real-time visual observations with LLM-derived prior probabilities and relevance metrics, enabling autonomous decision-making for search direction selection in unknown environments. Extensive experimental validation confirms the superior performance of our approach in addressing visual target navigation challenges, with additional verification conducted on physical robotic platforms.For future work, we identify three key improvement directions: (1) optimizing semantic map construction, (2) enhancing spatial information representation to strengthen semantic relevance measurements, and (3) improving obstacle avoidance capabilities through advanced perception algorithms.

References

1. Chaplot, D.S.; Gandhi, D.P.; Gupta, A.; Salakhutdinov, R.R. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems* **2020**, *33*, 4247–4258.
2. Gervet, T.; Chintala, S.; Batra, D.; Malik, J.; Chaplot, D.S. Navigating to objects in the real world. *Science Robotics* **2023**, *8*, eadf6991.
3. Dharmadhikari, M.; Dang, T.; Solanka, L.; Loje, J.; Nguyen, H.; Khedekar, N.; Alexis, K. Motion primitives-based path planning for fast and agile exploration using aerial robots. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 179–185.
4. Yang, W.; Wang, X.; Farhadi, A.; Gupta, A.; Mottaghi, R. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543* **2018**.
5. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, 2019, pp. 4171–4186.
6. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.
7. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* **2023**.
8. Liu, B.; Jiang, Y.; Zhang, X.; Liu, Q.; Zhang, S.; Biswas, J.; Stone, P. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477* **2023**.
9. Silver, T.; Hariprasad, V.; Shuttleworth, R.S.; Kumar, N.; Lozano-Pérez, T.; Kaelbling, L.P. PDDL planning with pretrained large language models. In Proceedings of the NeurIPS 2022 foundation models for decision making workshop, 2022.
10. Xie, Y.; Yu, C.; Zhu, T.; Bai, J.; Gong, Z.; Soh, H. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128* **2023**.
11. Kim, B.; Kim, J.; Kim, Y.; Min, C.; Choi, J. Context-aware planning and environment-aware memory for instruction following embodied agents. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 10936–10946.
12. Hu, Y.; Lin, F.; Zhang, T.; Yi, L.; Gao, Y. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842* **2023**.
13. Zhu, M.; Zhu, Y.; Li, J.; Wen, J.; Xu, Z.; Che, Z.; Shen, C.; Peng, Y.; Liu, D.; Feng, F.; et al. Language-conditioned robotic manipulation with fast and slow thinking. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 4333–4339.
14. Yang, Y.; Neary, C.; Topcu, U. Multimodal Pretrained Models for Verifiable Sequential Decision-Making: Planning, Grounding, and Perception. *arXiv preprint arXiv:2308.05295* **2023**.
15. Zhou, Z.; Song, J.; Yao, K.; Shu, Z.; Ma, L. Isr-llm: Iterative self-refined large language model for long-horizon sequential task planning. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 2081–2088.
16. Majumdar, A.; Shrivastava, A.; Lee, S.; Anderson, P.; Parikh, D.; Batra, D. Improving vision-and-language navigation with image-text pairs from the web. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. Springer, 2020, pp. 259–274.

 doi:10.20944/preprints202512.1129.v1

17. Shah, D.; Osiński, B.; Levine, S.; et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In Proceedings of the Conference on robot learning. PMLR, 2023, pp. 492–504.
18. Xie, Q.; Zhang, T.; Xu, K.; Johnson-Roberson, M.; Bisk, Y. Reasoning about the unseen for efficient outdoor object navigation. *arXiv preprint arXiv:2309.10103* **2023**.
19. Chen, W.; Hu, S.; Talak, R.; Carlone, L. Leveraging large (visual) language models for robot 3D scene understanding. *arXiv preprint arXiv:2209.05629* **2022**.
20. Yu, B.; Kasaei, H.; Cao, M. L3mvn: Leveraging large language models for visual target navigation. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2023, pp. 3554–3560.
21. Zhang, Y.; Huang, X.; Ma, J.; Li, Z.; Luo, Z.; Xie, Y.; Qin, Y.; Luo, T.; Li, Y.; Liu, S.; et al. Recognize anything: A strong image tagging model. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 1724–1732.
22. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In Proceedings of the European Conference on Computer Vision. Springer, 2025, pp. 38–55.
23. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ade20k dataset. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 633–641.
24. Anderson, P.; Chang, A.; Chaplot, D.S.; Dosovitskiy, A.; Gupta, S.; Koltun, V.; Kosecka, J.; Malik, J.; Mottaghi, R.; Savva, M.; et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757* **2018**.
25. He, Y.; Zhou, K.; Tian, T.L. Multi-modal scene graph inspired policy for visual navigation. *The Journal of Supercomputing* **2025**, *81*, 1–22.
26. Xu, W.; Zhang, F. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters* **2021**, *6*, 3317–3324.