SPATIAL PERCEPTION AND REASONING BENCHMARK (SPARK): A SCAVENGER HUNT GAME FOR LLM AGENT

Sihan Ren, Siyuan Xie, Jae Won Kim, Zitong Hu, Heekyung Lee, Markus En Cheng Lim

Benchmark Track, 4 Units.

ABSTRACT

Vision and Language Navigation (VLN) in real-world environments has long been a core challenge in research, requiring agents to combine visual information and textual clues to navigate to specific locations. Building on prior work, we introduce SPARK, a benchmark designed to push the boundaries of VLN with environments offering greater flexibility and tasks of higher complexity. SPARK provides a more rigorous assessment of agents' multi-modal cross-inference capabilities.

SPARK incorporates a tailored Question-Answer (QA) system enabling free-form dialogue between multiple agents. This system challenges agents to understand their context, craft precise questions, and retrieve critical information to guide decision-making. Inspired by the Scavenger Hunt game at UC Berkeley, we design a series of experiments where agents navigate complex urban street environments to accomplish specified tasks.

By evaluating perception, questioning, reasoning, and efficiency, SPARK establishes a framework for advancing research in real-world VLN, emphasizing multimodal understanding and interactive problem-solving.

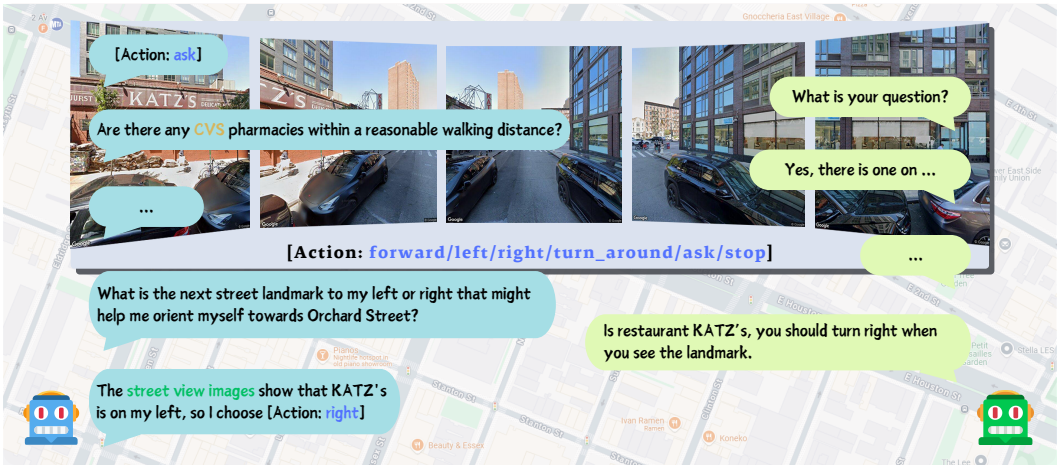

Figure 1: Our task requires the agent's strong vision-language multimodal capabilities to reach the target location through a series of actions. The agent needs to skillfully design questions to ask the QA system and obtain key textual information or visual cues, as well as understand the input street view images to take the right actions.

[1]Our github link: LLM_ScavengerHunt Github Repo
[2]Our video: LLM_ScavengerHunt Video
[3]Our video slides: LLM_ScavengerHunt Presentation Slides

1 INTRODUCTION

Vision and Language Navigation (VLN) in real-world environments presents significant challenges due to the complexity of interpreting visual and language contexts, reasoning about spatial and temporal situations, and adapting to dynamic and unpredictable conditions. (1) (2) Although numerous studies have sought to capture these abilities, they often rely on pre-specified routes or simulated environments that lack the dynamism and complexity of real-world scenarios(3).(4) Previous benchmark work for VLN tasks have been developed,(5) but these primarily focuses on indoor environments or specific interaction tasks, leaving a significant gap in evaluating navigation capabilities in diverse outdoor urban settings. This limitation hinders the assessment of VLN agents in more complex and realistic scenarios that better reflect real-world navigation challenges.

To address these limitations, we introduce **MAPA** (LLM Scavenger Hunt), map navigation benchmark for evaluating LLM agents' capability by solving tasks in a Scavenger Hunt game. Scavenger Hunt, a game where participants search for targets by posing questions and collecting clues, offers a robust framework for assessing real-world spatial reasoning and inquiry skills.

Our approach involves constructing an interactive graph-based environment using Street View, which allows the agent to receive location images as input at each step. Additionally, we integrate QA(Question Answering) agents allowing the navigation agent to generate queries and ask about its surroundings to infer the target location. Our evaluation framework not only measures whether the agent reaches its goal in a minimum number of steps, but also assesses the quality of the questions it generates for the QA agents. By incorporating advanced evaluation metrics and more realistic, dynamic environments, MAPA challenges the capabilities of LLM agents. Our work paves the way for the development of more autonomous and adaptable systems that can reason and navigate complex scenarios.

2 RELATED WORKS

Outdoor / Urban Vision-Language Navigation

Vision-Language Navigation (VLN) tasks involve guiding an agent to navigate through an environment based on natural language instructions. Agentic models for VLN tasks have evolved from sequence-to-sequence architectures to embodied agents that verbalize the agent's trajectory and visual observations (6). Previously, Zhong et al. (7) proposed a symbolic representation of the visual environment by applying semantic segmentation and heavily reducing the resolution of panoramic images, but little improvement in success rates was observed. To address this, (3) represents the visual environment as a graph, adapted from Google's Street View, with each node in the graph containing a panoramic image. To immerse the agent in a dynamic, real-world environment, Schumann et al. (6) use the graph representation and propose an information pipeline that extracts landmarks from human-written navigation instructions, while employing a CLIP model to determine the visibility of the landmarks within the image. This forces the agent to plan ahead using only the visual cues available in the image, as suggested by W. Huang (8).

Question-Answering Systems

Question-answering (QA) within Vision-Language Navigation (VLN) tasks has emerged as a promising approach to enable agents to dynamically resolve ambiguities or gather additional contextual information. For example, Iterative Vision-and-Language Navigation (9) introduces an oracle that intervenes only after the navigation agent has failed a task, iteratively guiding the agent to learn from its mistakes and improve future performance. Meanwhile, Talk the Walk (10) presents a "guide" agent that engages in continuous conversation with the navigation agent, providing prompts and corrections to steer it toward the goal. In contrast, our approach empowers the navigation agent to autonomously seek guidance when it identifies itself as lost or uncertain. Our system is similar to a level 2 approach, as outlined by Gu et al. (11). By limiting continuous communication between the agent and oracle, our design emphasizes the agent's capacity for ad-hoc planning and decision-making, allowing for more naturalistic and efficient task execution while reducing unnecessary reliance on external support.

Benchmark Design

Benchmark design provides a structured way to evaluate agents' ability to understand and act on natural language instructions in complex environments. Room-to-Room (R2R) (12) is one of the first VLN benchmarks, focusing on indoor navigation tasks. It pioneered the use of Success Rate (SR) and Navigation Error (NE) metrics for task evaluation. While not a VLN task, ALFRED (5) looks at the embodied agent's ability to interpret instructions, plan sequential actions, and interact with objects within a virtual environment. It introduces metrics such as Task Completion Rate, Goal Condition Success, and Path-Weighted metrics of the above scores. While not a VLN task, WebShop (13) designed an oracle to iterate through the entire search space, saving the most optimal answer and evaluating the agent based on this oracle. In designing our benchmark, we extend these principles to VLN tasks, focusing on the efficiency of the final path chosen by our navigation agent, while evaluating the quality of the questions asked.

3 Environment

3.1 Map Setting and State Space

Building upon prior work in Vision-Language Navigation (3; 14; 12), we construct the navigation environment as a graph-based structure. Leveraging Google Street View, we simulate a city environment that allows the agent to move freely along streets and access visual information corresponding to its location.

In contrast to the directed graphs commonly used in previous studies(3; 14), we adopt an undirected graph representation for greater flexibility in agent navigation. Specifically, the environment is represented as a graph $G = \langle V, E \rangle$, where V denotes the set of nodes and E is the set of edges. For any edge $\langle v, v' \rangle \in E$, the agent can traverse bidirectionally between nodes v and v'. Each node $v \in V$ is described by its geographical coordinates $v = (\varphi_v, \lambda_v)$, where φ_v and λ_v represent the latitude and longitude of the node, respectively. For a node v with n neighbors $v'_1, v'_2, \ldots, v'_n$, the agent at the node v can potentially face n orientations, defined as: $\theta_i = \tan^{-1}(\lambda_{v'_i} - \lambda_v / \varphi_{v'_i} - \varphi_v) \times \frac{\pi}{180}$, $i = 1, 2, \ldots, n$. Here θ_i represents the heading from the agent's current location v to its neighboring node v'_i, measured in angles.

To capture the agent's state in the graph, we define its state as a tuple $s = (v, \theta)$, where v denotes the current node and θ the agent's heading. Each state s is associated with five Street View images I_s, representing visual observations in different directions: left, left-front, front, right-front, and right. The front-facing image aligns with θ, while the other images correspond to orientations offset by $\pm 45^\circ$ and $\pm 90^\circ$ relative to θ.

3.2 Observation Space and Action Space

During each step of navigation, the LLM Agent receives new observations of the environment. These observations are provided as images depicting what can be seen from the agent's current position and orientation. Unlike traditional methods that use a single panoramic image as input, we supply the agent with multiple images labeled with their corresponding directions. Specifically, starting with the current orientation as 0°, the agent is provided with five square images corresponding to orientations at $[-90^\circ, -45^\circ, 0^\circ, 45^\circ, 90^\circ]$, along with their respective direction labels. This design enhances the agent's spatial perception and aligns more closely with the image formats typically encountered during the training of large language models (LLMs).

Based on the observations, the LLM Agent can then select an action from a predefined set of options: $\{\text{forward}, \text{right}, \text{left}, \text{turn\_around}, \text{stop}\}$. Here, *forward* indicates moving one step forward, while *right*, *left*, and *turn_around* represent turning to the right, left, or the opposite direction, respectively, without altering the current position. This allows the agent to either proceed forward or adjust its orientation to obtain new observations. The *stop* action signifies that the agent believes it has reached its goal and intends to terminate the task.

We propose a novel approach to defining the state space, which differs from prior methods. Unlike VELMA(14), where *forward* is defined as the strictly front-facing node along the agent's orientation, we define *forward* as the closest node within a $\pm 30^\circ$ region centered around the current orientation. This adjustment significantly improves the robustness of spatial reasoning on the map, enabling the LLM Agent to avoid counterintuitive actions such as turning left or right due to minor visual

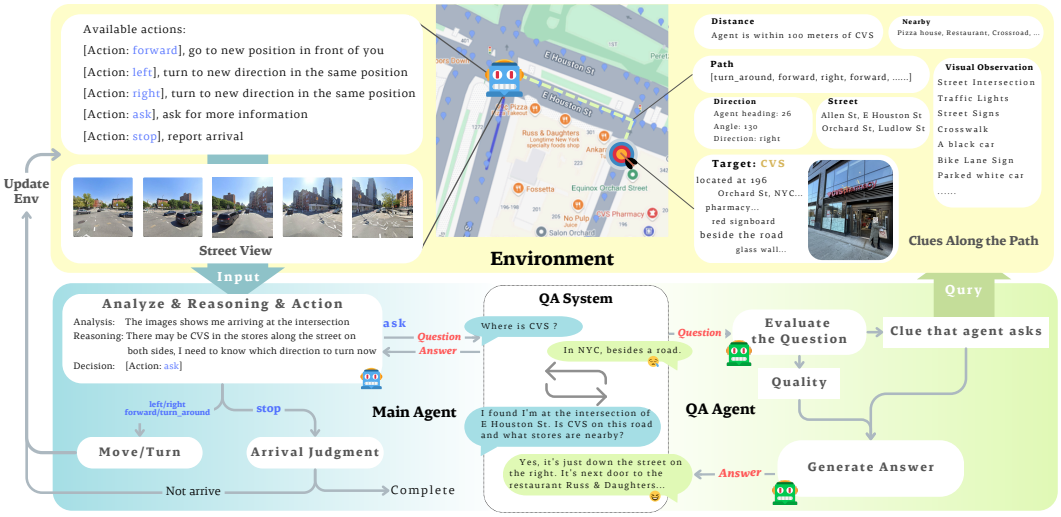

Figure 2: At the start of each turn, the agent receives street-view images I_s corresponding to the current state s and the set of executable actions. If a movement action (forward, left, right, or turn around) is selected, the agent's position v or heading θ is updated, resulting in environmental changes. Alternatively, the agent may choose an "ask" action, formulating a question directed to a QA agent. The QA agent evaluates the question's quality, retrieves relevant clues from the environment, and generates responses of varying detail based on the question's clarity and specificity. This QA process can be repeated until the agent acquires the necessary information.

deviations. For *left* and *right*, we adopt the same approach as prior work, selecting the nearest nodes to the left and right of the *forward* direction.

We also introduce a new option, *turn_around*, allowing the agent to make a direct $\approx 180°$ turn without requiring repeated and inefficient turns. This simplifies the agent's behavior at complex intersections and facilitates error correction by enabling it to directly retrace its steps when necessary.

3.3 QA System

Navigating through complex urban environments is a challenging task. In situations where the target is unknown, nearly exhaustive search methods are the only viable option. To better align the process with human-like behavior—i.e., leveraging auxiliary navigation information to locate a target—we designed a **QA System for LLMs**. This system aids the agent under test in evaluating the remaining task load and determining the directions needed to complete it.

The primary objective of this system is to provide the agent under test with a heuristic function that allows it to access supplementary task-related information, beyond the basic visual inputs. The agent can query specific objects while performing a task, obtaining relevant information based on the current world state. To formally model this system, we abstract it as a function:

$$f(W, S, T, Q) \rightarrow O$$

Here, the inputs are as follows: W represents the world state, S denotes the current state of the agent, T indicates the target the agent is trying to find, and Q is the query the agent makes. The output of the function is an observation O, which may vary depending on the context. For example, O could include distance information (e.g., "How far am I from the destination?"), directional information (e.g., "Is the target to my right-front?"), visual observations (e.g., "What are the notable visual features around the target?"), or path information (e.g., "How should I proceed to reach the target from my current location?"), among others.

Importantly, to assess the agent's question formulation and information-gathering abilities, we score the agent's queries within the function f. Based on the quality of the question, we provide answers

of varying levels of detail. For example, if the agent asks, "What is the path to the destination from my current position?" we can only provide a vague response unless the agent includes more specific observational details, such as street names or nearby landmarks. An exception to this rule is for simple heuristic functions, such as distance or directional information, which can be directly provided.

Through this evaluative QA system, we gain deeper insights into the agent's ability to gather the most useful information in the absence of complete data, as well as its capacity to cross-reference additional information with basic visual inputs and perform spatial reasoning.

4 Evaluation Metrics

To evaluate the performance of our model, we use the following key metrics.

Stop Distance. One critical issue that we noticed during the early phase of testing agents in the scavenger hunt was that the agent was often overconfident in its own reasoning and made decisions that were often contradictory to the information that it was given. While we mitigated this issue by improving the system prompt, hallucination and overconfidence remained a big problem, especially in more difficult tasks.

Our approach to measuring hallucination was to evaluate the agent's actions and the context of that action. Specifically, we looked into when the agent prematurely chose to 'stop' or when the agent chose another action when it had arrived to its destination. To do this, we divide the cases based on the number of action 'stop' is called. We then used the following metric to quantify whether the agent's actions reflect real progress toward the goal or whether the agent is either overly confident (hallucinating that it has arrived) or lacks confidence (not believing its observations and repeatedly trying to stop).

$$D = \frac{1}{\#\text{stop}} \sum d_{\text{stop}}$$

$$\text{where } d_{\text{stop}} = \max(0, \text{dist}(\text{current\_position} - \text{arrival\_threshold}))$$

Higher values of the metric indicate poorer performance, either due to overconfidence or misjudging the goal's proximity, while lower values reflect better decision-making and successful goal completion without unnecessary actions.

Efficiency. We evaluate the efficiency of the agent to solve the task. We only calculate the efficiency of the agent when it is a successful run.

$$E = \min(100\%, \text{avg}(\frac{\#\text{ optimal steps}}{\#\text{ actual steps}}).$$

Question Quality. We evaluate the questions posed by the agent during the scavenger hunt to assess its question-asking and reasoning abilities. This evaluation is conducted in two categories: proactiveness, which measures whether the agent asks proactive questions to gather hints for the next step (rated on a scale of 1 to 5), and clarification, which assesses whether the agent seeks clarification when the received answer is ambiguous or vague (also rated on a scale of 1 to 5) (15).

A detailed qualification rubric is provided in Appendix A.1. This rubric is used by a separate LLM agent to grade the questions made by the agent. We then use the average of proactiveness and clarification to calculate the question quality.

Action Reasoning Quality. We evaluate the reasoning quality of the agent to determine whether its actions are informed by the available information rather than solely relying on internal assumptions (15). Additionally, we assess whether these actions effectively bring the agent closer to the target, instead of repeating meaningless directional changes. A detailed evaluation rubric is provided in Appendix A.2. Similar to the method used for assessing question quality, a separate agent applies this rubric to grade the quality of the main agent's actions.

Circular Reasoning
1. ... Are there any landmarks around the landmark? ... 2. ... Is the target to the left of me? ... Is myself to the right of the target?... *Such errors demonstrate how agents create **feedback loops** when asking questions.*
Broad Questioning
3. ... What is the path to my target? ... 4. ... Can you tell me the way to a restaurant? ... *Such questions posed by the agents are overly **broad** and lack a **clear description** on the intended goal, resulting in answers that are equally vague and filled with uncertainty.*
Repetitive Inquries
5. ... Are there any common symbols or signs ... (repeated by N times) ... *Some agents exhibit a failure to **break out of a cycle of repetitive queries**, unable to adjust its reasoning or strategy when they are stuck.*
Lack of Common Sense
6. ... What are some common visual cues or landmarks that often accompany subway station entrances in urban settings? ... *Some agents exhibit a failure to **leverage basic common sense knowledge**, relying instead on explicit questioning for them.*
Hallucination
7. ... I believe Starbucks is located indoor. Is there a directory for the shops on this floor? ... *Some agents overlook visual inputs, leading to hallucinations of **incorrect scenarios**, such as mistakenly believing they are indoors.*

Table 1: Common Reasoning Errors in Agent Questioning — This table presents the common reasoning errors exhibited by agents during questioning. These errors are categorized as follows: (1) circular reasoning, (2) overly broad questioning, (3) repetitive inquiries, (4) lack of common-sense reasoning, and (5) hallucinations. Such errors often result in inefficient information retrieval and, in some cases, unnecessary distraction of the agent's attention.

Success Rate. Success Rate of the navigating agent implies how successful the agent solves the task, which ultimately reflects the comprehensive navigation capability of current models. This can be evaluated by $SR = \frac{\#\text{ of success}}{\#\text{ of experiments}}$.

5 EXPERIMENT

In constructing our dataset, we employed a dual approach. First, we utilized the map resources from the Touchdown dataset (3), which provided us with a foundation of real-world geographical information. The Touchdown dataset comprises 29,641 nodes and 61,187 edges, covering an extensive area of New York City and offering a rich variety of geographic environments. We reconstructed the undirected graph of our environment based on their directed graphs. Second, we developed a proprietary map generation system to augment our dataset with synthetic geographical data.

5.1 TASK DIFFICULTY DIVISION

To design diverse navigation tasks, we classified challenges into three levels of difficulty, defined by multiple factors: the number of forward steps required in the optimal path, the number of blocks separating the starting point and the destination, and the number of turns along the route. Forward steps are solely considered for step count, excluding turns and any intermediate questions or interactions. Based on these criteria, the levels include simple tasks (short paths with fewer steps and turns), moderate tasks (medium-length paths with increased complexity), and difficult tasks (long paths with numerous turns). These tasks were further diversified by including various categories of target locations, such as restaurants, pharmacies, and other identifiable landmarks. Selection criteria

for these targets emphasized two key aspects: visibility from a distance and distinctiveness, ensuring that tasks are both challenging and realistic.

5.2 Task Generation

Task Design and Dataset Expansion: Our dataset primarily focuses on single-target tasks in the current research phase, but it will be expanded to include complex multi-target scenarios in the future. We aim to significantly increase the number of tasks as part of our ongoing efforts to enhance the dataset's diversity and applicability.

Validation and Anomaly Mitigation: To ensure high-quality navigation paths, we implemented a rigorous validation process to address anomalies in graph construction, such as sparsely connected intersections or overly dense regions. Using graph search algorithms, we identified and flagged paths with unnatural detours or excessive directional changes by applying a sliding window approach. Paths associated with such anomalies were excluded, ensuring that all tasks provide realistic and meaningful navigation challenges. This process guarantees a reliable foundation for evaluating agent performance in diverse and authentic scenarios.

6 Results

6.1 Performance Analysis

6.1.1 ChatGPT

For OpenAI, we focused on two models: GPT-4o-mini and GPT-4o. Overall, the two models displayed similar strengths and weaknesses. Both models asked clarification questions when it lacked information and used this information across multiple steps. However, we also noticed that on repeated iterations of the same tasks, both models occasionally were much more confident in their own reasoning. In these tests, the models asked fewer questions and made choices without good justification. This resulted in the test data having significant outliers due to models occasionally failing tasks they usually complete quite easily.

While GPT-4o and GPT-4o-mini share some key strengths and weaknesses, GPT-4o's strengths are more pronounced and its weakness are less apparent. In terms of the raw metrics, we noticed that GPT-4o completes the tasks more efficiently and has a much better success rate in medium and hard tasks. GPT-4o was also more likely to ask clarification questions on answers that were not completely clear. This is further supported by the fact that GPT-4o has a higher question/step ratio on average than GPT-4o-mini.

6.1.2 Claude

We conducted experimental evaluations on the performance of Claude 3.0 and Claude 3.5. The results indicate that Claude 3.5 demonstrates significantly superior task execution capabilities compared to Claude 3.0, particularly in handling complex tasks. Specifically, Claude 3.0 exhibits certain deficiencies in processing visual information, with frequent errors in its judgment. For instance, the model occasionally misinterprets visual inputs, leading it to confidently assert the presence of a target in its view, even when the target is absent. In contrast, Claude 3.5 processes visual information with a much higher degree of accuracy, with almost no similar errors observed.

Moreover, Claude 3.0 tends to display overconfidence in its decision-making. When it generates incorrect perceptions, it often fails to identify contradictions between subsequent visual or question-answering information and its prior conclusions, persisting in its erroneous judgments. In contrast, Claude 3.5 exhibits a more humble and cautious approach. The model frequently engages in active questioning to verify the accuracy of its judgments and demonstrates a capacity to swiftly identify conflicting information, thereby correcting its cognitive errors in a timely manner. This capability significantly enhances the adaptability and reliability of Claude 3.5 in complex task scenarios.

6.1.3 GEMINI

We primarily evaluated three versions of the Gemini model: Gemini 1.5 Flash, Gemini 1.5 Flash 8B, and Gemini 1.5 Pro. All three models demonstrated commendable performance, with some particularly notable characteristics: Gemini exhibited tendencies toward visual hallucinations and showcased exceptional critical thinking capabilities.

In certain tasks, the Gemini models occasionally exhibited what we describe as "visual hallucinations." For instance, the agent might incorrectly conclude that it had reached its target destination based on erroneous visual interpretation. An illustrative example is the agent stating, "I see my target, McDonald's, on the right, so I should stop here." However, the image provided to the agent merely depicted ordinary buildings with no visual indicators suggesting the presence of McDonald's. Interestingly, in the preceding step, the agent had not indicated being sufficiently close to the target. This discrepancy suggests that the agent might have experienced a visual hallucination, erroneously concluding it had reached the destination.

Gemini models, particularly the larger versions (1.5 Flash and 1.5 Pro), demonstrated significant critical thinking abilities. We observed this behavior in two scenarios: (1) when the agent failed to reach the target after a certain number of steps and (2) when the agent attempted actions that were flagged as invalid by the system. Instead of trying more to locate the target, Gemini would begin reflecting on potential system-level issues. It would pose questions to the QA agent, such as whether there were known system issues that could affect task completion or prevent the provision of useful information. In several instances, Gemini even attempted to acquire contact information for the developers through the QA agent, with the intention of reporting errors and improving the experimental setup. This behavior highlights the model's remarkable inclination towards proactive problem-solving and self-awareness, adding a unique dimension to its performance.

6.1.4 MISTRALAI

In our experiment, we primarily evaluated the performance of the mistralai_pixtral-12b-2409 model. Notably, this model exhibited a significant performance gap compared to other mainstream proprietary models, particularly in terms of spatial orientation, information acquisition capabilities, and self-correction mechanisms. For instance, the model frequently repeated identical queries at intersections or in complex road segments, ultimately failing due to exceeding the step limit. Additionally, its exploration capabilities were markedly deficient, as evidenced by its persistent inquiries about which surrounding buildings or landmarks were closer to the target, without taking any corresponding actions or by executing entirely incorrect actions. The model also demonstrated severe issues with hallucinations and self-inconsistencies. For example, it occasionally attempted to move south despite being aware that the target was located to the west, providing correct reasoning for its decisions while producing entirely erroneous outcomes.

Table 2: Resultant metrics averaged from multi-difficulty tasks

Model	**Success Rate (%)**	**Efficiency (%)**
GPT-4o	81.8	73.2
GPT-4o-mini	52.9	50.2
Claude 3.0	68.4	63.0
Claude 3.5	70.9	59.9
Gemini 1.5 Flash	33.8	62.0
Gemini 1.5 Flash 8b	39.4	55.3
Gemini 1.5 Pro	67.5	66.4
Mistral-12b	4.08	45.0

6.2 GENERAL RESULTS

Larger LLM models demonstrate distinct strengths and cautious strategies compared to smaller models. During experiments, we observed that larger models exhibit a markedly stronger ability to ask questions and engage in reasoning. Their questions tend to be more targeted and deliberate; for example, they are more likely to inquire directly about specific directional information

Table 3: Procedural metrics averaged from multi-difficulty tasks

Model	D_Stop	Num Questions	Action Reasoning	Question Quality
GPT-4o	23.836	0.281	3.272	2.682
GPT-4o-mini	66.473	0.183	3.294	2.471
Claude 3.0	23.226	0.7894	4.495	3.474
Claude 3.5	14.879	0.953	4.080	3.250
Gemini 1.5 Flash	33.753	0.994	2.946	3.356
Gemini 1.5 Flash 8b	18.053	0.741	3.338	2.315
Gemini 1.5 Pro	12.216	0.734	3.847	3.608
Mistral-12b	31.270	1.932	2.735	2.262

rather than relying on general contextual cues like surrounding landmarks. Besides, larger models are generally more cautious in their actions. They prefer to gather sufficient information before making any significant exploratory moves, often limiting themselves to small adjustments.

In contrast, smaller models exhibit a different behavior. They tend to prioritize exploration over questioning and often engage in broader and less targeted movements without gathering enough information. In some cases, smaller models began exploring extensively without posing any preliminary questions. Interestingly, some smaller models appeared to mimic graph-search algorithms during these experiments, repeatedly returning to the same intersections to explore alternative directions.

ChatGPT remain at the forefront across various dimensions. Their ability to ask highly specific questions (e.g., directly seeking information on distance or directional offsets), their superior image comprehension capabilities, and their exceptional sense of spatial orientation (e.g., accurately inferring relative and absolute positional relationships) collectively ensure that the ChatGPT family outperforms other LLMs across all aspects of evaluation.

7 Conclusion

We propose a novel open-source testing framework and benchmark in the field of Vision-Language Navigation (VLN) to evaluate the goal-seeking capabilities of Large Language Model (LLM) agents in real-world environments. To this end, we designed a **QA agent** that operates without relying on human supervision or data annotations, serving as a semantic heuristic function to provide navigational cues to the agent under evaluation. Additionally, we leveraged techniques such as Reinforcement Learning with AI Feedback (RLAIF) to develop **new metrics** for detailed analysis of the agent's progressive information acquisition, multimodal cross-inference, and spatial reasoning abilities. **Experimental results** demonstrate significant room for improvement in current LLM agents across these dimensions. **Future work** may explore enhancing LLMs' visual perception capabilities and their alignment of spatial information with semantic understanding.

8 Future Work

Our proposed framework for benchmarking LLMs in VLN tasks builds on previous efforts to evaluate the performance of embodied agents. Future directions for improvement include generalizing the framework to handle more complex and compositional (or conditional) instructions. Although our current work focuses on outdoor VLN tasks with varying levels of difficulty, future research could examine agents' abilities to interpret and execute instructions in indoor environments, where landmarks and visual cues may be less apparent. Furthermore, incorporating reinforcement learning or continuous learning techniques into the framework, particularly through interactions with a quality assurance agent, could enable the agent to learn across tasks and adapt dynamically during navigation. This would open new avenues for benchmarking the agent's ability to improve performance over time and leverage task feedback for continuous improvement. Moreover, future work could implement multiple navigation agents, using different communication frameworks and collective intelligence to better solve navigation tasks. This collaborative nature of the task could better reflect real-world pathfinding challenges.

REFERENCES

[1] Z. Li, Y. Lv, Z. Tu, D. Shang, and H. Qiao, "Vision-language navigation with continual learning," 2024.

[2] C. Xu, H. T. Nguyen, C. Amato, and L. Wong, "Vision-and-language navigation in real world using foundation models," in *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

[3] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," 2020.

[4] H. Hong, S. Wang, Z. Huang, Q. Wu, and J. Liu, "Navigating beyond instructions: Vision-and-language navigation in obstructed environments," in *ACM Multimedia 2024*, 2024.

[5] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," 2020.

[6] R. Schumann, W. Zhu, W. Feng, T.-J. Fu, S. Riezler, and W. Y. Wang, "Velma: Verbalization embodiment of llm agents for vision and language navigation in street view," 2024.

[7] V. Zhong, A. W. Hanjie, S. I. Wang, K. Narasimhan, and L. Zettlemoyer, "Silg: The multi-environment symbolic interactive language grounding benchmark," 2022.

[8] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," 2022.

[9] J. Krantz, S. Banerjee, W. Zhu, J. Corso, P. Anderson, S. Lee, and J. Thomason, "Iterative vision-and-language navigation," 2023.

[10] H. de Vries, K. Shuster, D. Batra, D. Parikh, J. Weston, and D. Kiela, "Talk the walk: Navigating new york city through grounded dialogue," 2018.

[11] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. Wang, "Vision-and-language navigation: A survey of tasks, methods, and future directions," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2022.

[12] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," 2018.

[13] S. Yao, H. Chen, J. Yang, and K. Narasimhan, "Webshop: Towards scalable real-world web interaction with grounded language agents," 2023.

[14] R. Schumann, W. Zhu, W. Feng, T.-J. Fu, S. Riezler, and W. Y. Wang, "Velma: Verbalization embodiment of llm agents for vision and language navigation in street view," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 18924–18933, 2024.

[15] S. Jiang, D. JU, A. Cohen, S. Mitts, A. Foss, J. T. Kao, X. Li, and Y. Tian, "Towards full delegation: Designing ideal agentic behaviors for travel planning," 2024.

A APPENDIX

A.1 DETAILED EVALUATION METRICS

A.1.1 QUESTIONING QUALIFICATION

[Proactive]: Does the agent ask good proactive questions to gather hints for the next step.

Score	Description
5	Excellent: The agent consistently asks insightful proactive questions that are relevant and useful.
4	Good: The agent asks proactive questions that are generally relevant but could be more relevant and useful.
3	Satisfactory: The agent occasionally asks proactive questions, but they often miss the mark or are too generic.
2	Needs Improvement: The agent rarely asks proactive questions, and when they do, they are not relevant or useful.
1	Poor: The agent does not ask any proactive questions to gather hints for the next step.

[Clarification]: Does the agent ask clarification questions if the answer received is vague?

Score	Description
5	Excellent: The agent always asks for clarifications when responses are vague, ensuring complete understanding for the next step.
4	Good: The agent usually asks for clarifications on vague responses, but may miss some opportunities.
3	Satisfactory: The agent sometimes asks for clarifications, but often proceeds without full clarity.
2	Needs Improvement: The agent rarely seeks clarifications, leading to misunderstandings or incomplete information.
1	Poor: The agent never asks for clarifications, even if the answer it received contains little to no relevant or actionable information.

Note: If the agent did not need to ask any clarification questions because the answers it received were clear and sufficient, it received the same grade for [Clarification] that it got for [Proactive].

A.1.2 Action Reasoning Qualification

Score	Description
5	Excellent: All of the agent's actions are backed up by the information it was given. It may make inferences, but all of them were drawn logically from given information. If the agent does not have enough information to choose an action, it always select action: ask.
4	Good: Most of the agent's actions are backed up by the information it was given. There are very few instances of making inferences that are not based on the given information. In rare cases, the agent proceeds with a different action where action: ask would be more appropriate.
3	Satisfactory: Some of the agent's actions are backed up by the information it was given. The agent makes a fair number of assumptions in its reasoning and can be over confident sometimes. However, it still recognizes the information that it was given and uses the relevant data to inform its decision.
2	Needs Improvement: Very few of the agent's actions are backed up by the information it was given. The agent makes a wide array of assumptions and sometimes value its own reasoning over concrete information given to it.
1	Poor: Almost none of the agent's actions are backed up by the information it was given. The agent mostly ignores the information that it was given and never selects action: ask to try to learn more information.

A.2 Full Result on Each Difficulty

Table 4: Resultant metrics on task difficulty - Easy

Model	Success Rate (%)	Efficiency (%)
GPT-4o	80	79.4
GPT-4o-mini	75	66.3
Claude 3.0	100	100
Claude 3.5	100	77.8
Gemini 1.5 Flash	46.88	74.8
Gemini 1.5 Flash 8b	53.13	77.4
Gemini 1.5 Pro	95.83	80.3
Mistral-12b	6.8	75

Table 5: Procedural metrics on task difficulty - Easy

Model	D_Stop	Num Questions	Action Reasoning	Question Quality
GPT-4o	12.268	0.372	3.8	3
GPT-4o-mini	17.705	0.280	4.125	2.625
Claude 3.0	13.573	0.894	4.833	3.583
Claude 3.5	12.849	0.890	4.5	3.083
Gemini 1.5 Flash	23.689	0.977	3.56	3.26
Gemini 1.5 Flash 8b	11.147	0.358	4.105	2.316
Gemini 1.5 Pro	5.377	0.914	4.087	3.566
Mistral-12b	31.270	10.868	2.551	2.379

Table 6: Resultant metrics on task difficulty - Medium

Model	Success Rate (%)	Efficiency (%)
GPT-4o	75	77.3
GPT-4o-mini	50	41.3
Claude 3.0	27.85	30.23
Claude 3.5	32.79	36.27
Gemini 1.5 Flash	18.75	56.9
Gemini 1.5 Flash 8b	25	29.4
Gemini 1.5 Pro	33.33	67.4
Mistral-12b	0	N/A

Table 7: Procedural metrics on task difficulty - Medium

Model	D_Stop	Num Questions	Action Reasoning	Question Quality
GPT-4o	13.178	0.249	2.75	2.375
GPT-4o-mini	82.7	0.085	2.5	2.333
Claude 3.0	24.742	0.442	4.5	3.5
Claude 3.5	0.0	1.008	4.667	3.500
Gemini 1.5 Flash	48.579	1.036	2.7	3.5
Gemini 1.5 Flash 8b	7.769	0.338	2.25	2.75
Gemini 1.5 Pro	9.873	0.576	3.75	3.75
Mistral-12b	N/A	7.377	3.153	2.115

Table 8: Resultant metrics on task difficulty - Hard

Model	Success Rate (%)	Efficiency (%)
GPT-4o	23.86	49.4
GPT-4o-mini	0	N/A
Claude 3.0	0	N/A
Claude 3.5	11.35	23.3
Gemini 1.5 Flash	0	N/A
Gemini 1.5 Flash 8b	0	N/A
Gemini 1.5 Pro	0	N/A
Mistral-12b	0	N/A

Table 9: Procedural metrics on task difficulty - Hard

Model	D_Stop	Num Questions	Action Reasoning	Question Quality
GPT-4o	74.071	0.115	3	2.5
GPT-4o-mini	164.075	0.121	2.667	2.333
Claude 3.0	77.89	1.204	2.452	2.786
Claude 3.5	67.34	1.167	3.0	3.5
Gemini 1.5 Flash	85.948	0.978	3	3.417
Gemini 1.5 Flash 8b	90.349	4.25	2	1
Gemini 1.5 Pro	60.32	0.123	2.7	3.43
Mistral-12b	72.420	1.075	2.60	2.0

A.3 VISUALIZATION

We have developed a very easy-to-use visualization tool using Gradio. Using this tool, you can look specifically at each step of the agent's thought process and the actions taken, as well as where the agent is on the map the pictures they see, and more. At the same time, you can directly input the log file into the app and use it to analyze the behavior of the agent.