WMNav: Integrating Vision-Language Models into World Models for Object Goal Navigation

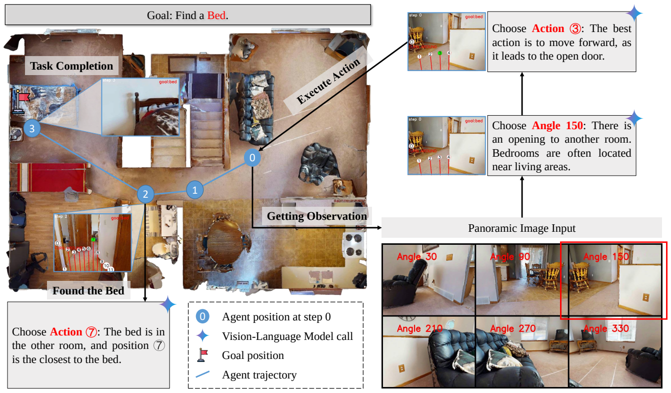
Dujun Nie[1,*], Xianda Guo[2,*,†], Yiqun Duan[3], Ruijun Zhang[1], Long Chen[1,4,5,‡]
[1] Institute of Automation, Chinese Academy of Sciences [2] School of Computer Science, Wuhan University
[3] School of Computer Science, University of Technology Sydney [4] IAIR, Xi'an Jiaotong University [5] Waytous
{niedujun2024,zhangruijun2023,long.chen}@ia.ac.cn;
xianda_guo@163.com; duanyiquncc@gmail.com

***Abstract*— Object Goal Navigation—-requiring an agent to locate a specific object in an unseen environment—-remains a core challenge in embodied AI. Although recent progress in Vision-Language Model (VLM)–-based agents has demonstrated promising perception and decision-making abilities through prompting, none has yet established a fully modular world model design that reduces risky and costly interactions with the environment by predicting the future state of the world. We introduce WMNav, a novel World Model-based Navigation framework powered by Vision-Language Models (VLMs). It predicts possible outcomes of decisions and builds memories to provide feedback to the policy module. To retain the predicted state of the environment, WMNav proposes the online maintained Curiosity Value Map as part of the world model memory to provide dynamic configuration for navigation policy. By decomposing according to a human-like thinking process, WMNav effectively alleviates the impact of model hallucination by making decisions based on the feedback difference between the world model plan and observation. To further boost efficiency, we implement a two-stage action proposer strategy: broad exploration followed by precise localization. Extensive evaluation on HM3D and MP3D validates WMNav surpasses existing zero-shot benchmarks in both success rate and exploration efficiency (absolute improvement: +3.2% SR and +3.2% SPL on HM3D, +13.5% SR and +1.1% SPL on MP3D). Project page: https://b0b8k1ng.github.io/WMNav/.**

I. Introduction

Effective navigation is a fundamental requirement for domestic robots, allowing them to access specific locations and execute assigned operations [1]. Zero-Shot Object Navigation (ZSON) is a critical component of this functionality, which demands that an agent locate and approach a target object from an unseen category through environmental understanding. This navigation capability proves indispensable for performing varied and sophisticated functions in practical home environments. The primary difficulty in ZSON stems from the need to employ broad semantic knowledge to direct movement with optimal efficiency while precisely identifying previously unencountered target objects.

With the recent advancements in visual foundation models, the effectiveness of visual perception has experienced a substantial enhancement, achieving remarkable performance in zero-shot scene understanding. Existing navigation methods can be divided into two categories: network-based navigation [2], [3], [4] and map-based navigation[5], [6], [7], [8], [9]. Network-based methods usually use reinforcement learning or imitation learning, which rely on trainable modules that need large amounts of data to train(such as policy learning)[10], [11], [12], resulting in significant computational resources and time for learning. Map-based methods construct maps to preserve semantic information about the scene and generate frontiers or waypoints for path planning. Their path planning heavily relies on accurate maps, which involve complex and time-consuming map construction processes. These methods fail to fully leverage the capabilities of VLMs trained with egocentric RGB images. Additionally, the selection strategies employed are often inflexible, frequently falling short of identifying optimal positions for decision-making. To address the limitations inherent in map-based planning, VLMNav[13] introduces an action space generation method that leverages navigable regions in egocentric images, providing a more flexible decision space and fully exploiting the capabilities of VLMs. However, due to the limited field of view of egocentric images, capturing environmental information outside the immediate perspective remains a significant challenge.



Choose Action ③: The best action is to move forward, as it leads to the open door.

Choose Angle 150: There is an opening to another room. Bedrooms are often located near living areas.

Choose Action ⑦: The bed is in the other room, and position ⑦ is the closest to the bed.

Fig. 1: **World Model Navigation with VLM.** In object navigation, our model first estimates the goal's presence likelihood in each scene of the panoramic image (e.g., a bed typically resides in a living room, which is likely situated at the corridor's terminus), then plans intermediate subtasks and chooses the most appropriate action to execute.

What's more, most existing methods, whether network-

[*] This author contributed equally to this work.
[†] Project Leader
[‡] Corresponding Author

based methods or map-based methods, require actual interaction with the environment to achieve accurate scene understanding. These approaches can not leverage predictive information about future states and the outcomes of potential actions, limiting their ability to perform anticipatory planning and reasoning effectively in uncertain scenarios. VLFM[14] constructs a value map of the likelihood of each region to lead toward the out-of-view target object. It makes use of prediction information to some extent. Still, it uses BLIP-2[15], which pays more attention to the relevance of image-text pairs and has limited interaction and reasoning capabilities, which makes it difficult to cope with complex planning tasks. When navigating to a specific object, humans can effectively imagine the room layout and anticipate action outcomes based on visual cues and common design patterns (such as walking along the corridor often leads to a new room and going to the bathroom to find a sofa is unwise).

The world model is a computational representation of environment dynamics that equips agents with the ability to simulate interactions. By simulating action choices within the virtual environment, agents can explore potential outcomes safely without directly interacting with the environment. This method not only reduces collision with obstacles but also enhances the agent's efficiency in planning and exploring. However, the true challenge lies in creating a versatile world model that can faithfully capture the landscape of an indoor environment. ATD [16] uses a dual-branch self-guided imagination policy based on large language models with a human-like left-right brain architecture. WebDreamer [17] uses LLMs as world models in web environments for model-based planning of web agents, demonstrating its effectiveness over other baselines. Can VLMs also function as world models? Given that VLMs are trained on vast amounts of first-person human perspective images, we hypothesize that they have acquired sufficient common sense knowledge to simulate the outcomes resulting from human choices and actions. With their extensive pre-trained knowledge spanning indoor layout and general reasoning ability, VLM has great potential to take on this task.

Building on the key insight that VLMs inherently encode comprehensive knowledge about indoor layout and spatial relationships of objects, we propose WMNav as shown in Figure 1, which is a pioneering approach leveraging VLMs in a world model paradigm to enable efficient navigation in complex indoor environments. It receives panoramic scene information and uses VLMs to quantitatively predict future outcomes and finish all the perception, planning, reasoning, and controlling processes without task-specific training, prebuilt maps, or prior knowledge of the surroundings(using detection or segmentation results).

Our contributions can be summarized as follows:

- We introduce a new direction for object goal navigation in a complex, unknown environment using a world model consisting of VLMs and novel modules.
- We design an innovative memory strategy of predicted environmental states that employs an online Curiosity Value Map to quantitatively store the likelihood of the target's presence in various scenarios predicted by the world model.
- We propose a subtask decomposition with feedback and a two-stage action proposer strategy to enhance the reliability of VLM reasoning outcomes and improve exploration efficiency.
- We achieve state-of-the-art results on the Zero-shot Object Navigation task and outperform benchmark methods on HM3D [18] and MP3D [19].

II. Related Work

A. Zero shot Object Goal Navigation

Existing object navigation methods fall into two paradigms: supervised methods and zero-shot methods. Supervised approaches either train visual encoders with RL/IL policies [20], [21], [22], [23] or build semantic maps from training data [24], [25]. While effective in known environments, these methods struggle with unseen objects/rooms due to training dependency. Recent zero-shot works address this via open-vocabulary scene understanding. Image-based methods map target objects to visual embeddings [2], [26], [4], while map-based approaches use frontier-based map [6], [5], [27], [28], [8], [9] or waypoint-based map [7] with LLM/VLM reasoning. Here, we focus on the zero-shot ObjectNav task in unknown environments using a novel world model framework.

B. Foundation Model Guided Navigation

Recent advances in embodied navigation highlight the complementary roles of Vision-Language Models (VLMs) and Large Language Models (LLMs), driven by their distinct strengths in visual grounding and common sense reasoning. LLMs enable zero-shot commonsense reasoning through object-room correlation prediction [5], semantic mapping [27], [7], and chain-of-thought planning [28], [29]. However, LLMs are unable to process rich visual information and do spatial reasoning. VLMs like CLIP [21], [2], [30] address this by directly aligning visual observations with textual goals through multimodal embeddings, offering seamless integration with exploration policies through unified visual features. Compared to LLM-based methods, VLM-based methods [9], [13] may be more efficient and elegant for target-driven navigation by avoiding language-mediated grounding gaps. This is due to their multimodal nature and better spatial reasoning capabilities [31]. Our framework uses pure VLM, and unlike other approaches, we use the VLM to predict the state of the environment caused by actions.

C. World Models

World models originated in classical reinforcement learning with Dyna's simulated experiences [32], evolving into frameworks for predicting state transitions to improve sample efficiency [33], [34]. These models enable policy training in diverse domains and emphasize high-fidelity environment dynamics. Recent work explores large language models (LLMs) as abstract world models, prioritizing task

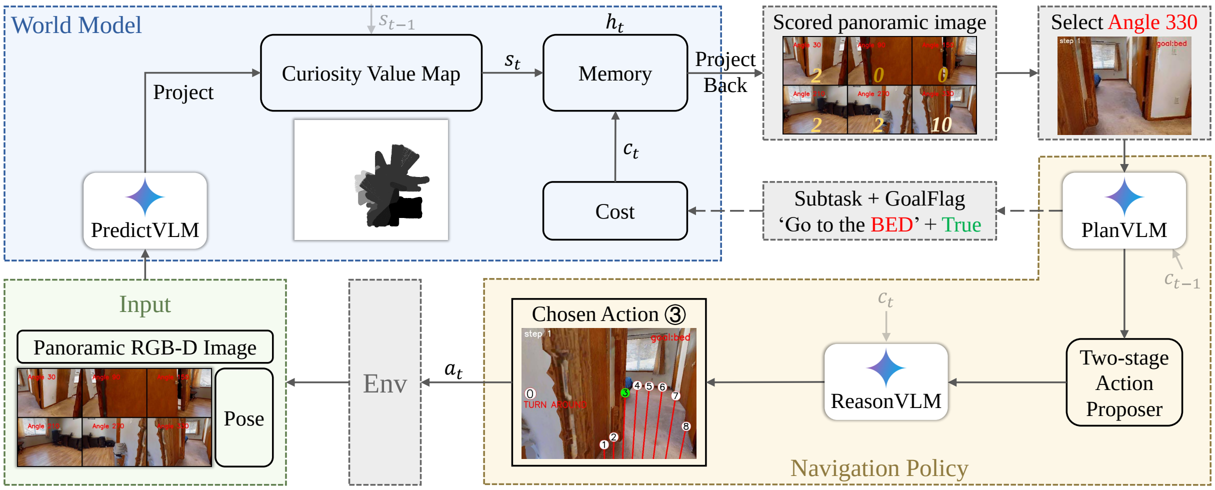


Fig. 2: **The WMNav framework.** After acquiring the RGB-D panoramic image and pose information at step t, the PredictVLM first predicts the state of the world, and the state is merged with the curiosity value map s_{t-1} from the previous step to get the current curiosity value map s_t. After that, the updated map projects the scores of each direction back onto the panoramic image, and the direction with the highest score is selected. Secondly, given the selected direction image, the new subtask and the goal flag are determined by PlanVLM and are stored in memory as cost c_t, and the memory h_t is combined by s_t and c_t. Finally, the two-stage action proposer annotates the action sequence on the selected image and sends it into ReasonVLM to obtain the final polar coordinate vector action a_t for execution. Note that PlanVLM and ReasonVLM are configured by the cost c_{t-1}.

abstraction over precise simulation. Studies like [35] demonstrate LLMs' ability to leverage commonsense knowledge for planning in simple environments, while [36] applies LLMs to web navigation. However, LLMs struggle with visually grounded decision-making. Therefore, it is more advantageous to integrate VLMs into world models in visual navigation tasks. VLMs combine visual grounding with semantic reasoning, enabling multimodal state understanding and adaptation to unseen environments through joint visual-textual prompts. Our approach takes VLM in a world model framework and involves it in all the processes of navigation, including predicting, planning, reasoning, action, etc.

III. WMNav Approach

A. Task Definition

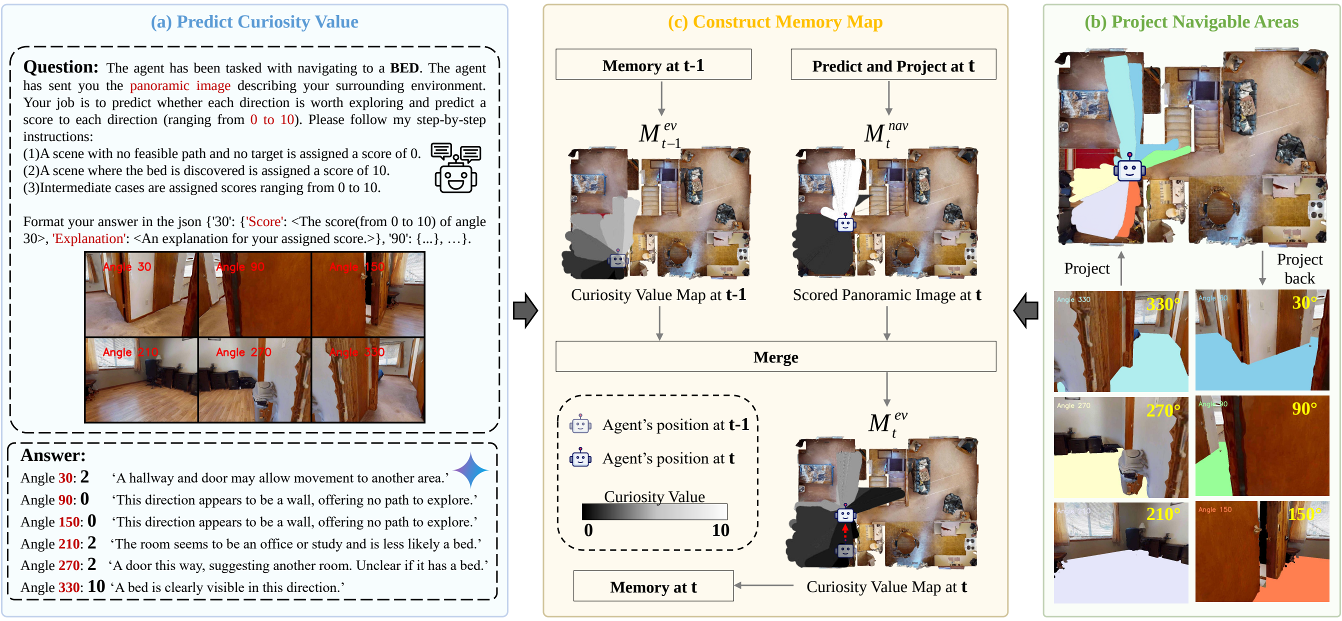
The traditional object goal navigation task requires the agent to explore an unknown indoor environment and navigate to an arbitrary instance i within a given category c (e.g., bed, sofa, toilet). The agent starts from a designated initial position. At each time step t, the agent takes an RGB-D observation O_t of the surroundings and its real-time pose P_t. Then, the agent determines the action a_t to find the goal. The task is deemed successful if the agent stops at a position within a predefined distance threshold d_{thres} from the goal. While most previous works utilize a discrete action space, such as {Stop, Move Forward, Turn Left, Turn Right, Look Up, Look Down}, we adopt polar coordinates (r_t, θ_t) to represent the action a_t, where θ_t denotes the direction of the action and r_t indicates the distance traveled by the action.

B. Overview

The framework of WMNav is depicted in Figure 2. A panoramic understanding is essential to make a comprehensive perception. To this end, the agent undergoes a series of rotations $\mathcal{A} = \{30°, 90°, 150°, 210°, 270°, 330°\}$ and captures six distinct RGB-D images which are converted into a panoramic image I_t^{pan}. In our framework, the world model consists of PredictVLM and the memory constructed by curiosity value map and cost. The world model does not receive any actual reward signals from the environment, which means it is only used to predict and simplify the future state of the environment. PredictVLM quantitatively predicts the likelihood of the goal's presence in each direction and projects the scores from a panoramic image to a top-down map. The map is then merged with the curiosity value map from the previous step and is stored in the memory. After that, the curiosity scores are projected back onto the panoramic image. Then, the direction in the panoramic image with the highest score is selected and sent to the navigation policy module. The navigation policy module has access to the reward information from the environment. For PlanVLM and ReasonVLM in the policy module, the cost (the previous step's subtask and the goal flag) is used to configure their prompts, thus optimizing the action output of the entire policy module without any fine-tuning of the VLMs.

C. World Model

1) VLM-based State Prediction: The core capability of the world model is to estimate the state of the world, which is not provided by perception, and to predict possible future state changes. We employ a VLM as the predictor in the world model. To guide the VLM to make reasonable predictions about the indoor scene, we design a novel prompting strategy as illustrated in Figure 3 (a). We use a panoramic image as the image prompt. PredictVLM is tasked with



Question: The agent has been tasked with navigating to a **BED**. The agent has sent you the panoramic image describing your surrounding environment. Your job is to predict whether each direction is worth exploring and predict a score to each direction (ranging from 0 to 10). Please follow my step-by-step instructions:
(1)A scene with no feasible path and no target is assigned a score of 0.
(2)A scene where the bed is discovered is assigned a score of 10.
(3)Intermediate cases are assigned scores ranging from 0 to 10.

Format your answer in the json {'30': {'Score': <The score(from 0 to 10) of angle 30>, 'Explanation': <An explanation for your assigned score.>}, '90': {...}, …}.

Answer:
Angle 30: **2** 'A hallway and door may allow movement to another area.'
Angle 90: **0** 'This direction appears to be a wall, offering no path to explore.'
Angle 150: **0** 'This direction appears to be a wall, offering no path to explore.'
Angle 210: **2** 'The room seems to be an office or study and is less likely a bed.'
Angle 270: **2** 'A door this way, suggesting another room. Unclear if it has a bed.'
Angle 330: **10** 'A bed is clearly visible in this direction.'

Fig. 3: **Predict the Likelihood.** (a) The world model predicts the Curiosity Value for each direction in the panoramic image based on the likelihood of the goal's presence. (b) The mutual projection of the navigable area between the ego-centric and top-down view perspectives. (c) Curiosity Value Map construction: The predicted scores from the world model are projected onto the top-down map and then fused with the previous step's Curiosity Value Map.

predicting each viewpoint's curiosity value, which represents the likelihood of the target's presence in each direction and scores from 0 to 10. The VLM outputs scores for each direction denoted as $Score_t(\alpha), (\alpha \in \mathcal{A})$. At each time step t, the panoramic image I^{pan}_t is input to PredictVLM, which outputs scores $Score_t$ for each direction in the current panoramic view:

$$Score_t = PredictVLM(I^{pan}_t) \tag{1}$$

These scores can then be leveraged to construct the curiosity value map.

2) Curiosity Value Map Construction: The curiosity value map M^{cv} has a size of $map\_size \times map\_size \times 1$, where each pixel value represents the curiosity value of the position in the entire scene, ranging from 0 to 10. For regions that have been visited and found to be devoid of the goal object, the curiosity value is set to 0. For example, an observed bedroom that does not contain the goal television would have a curiosity value of 0. Regions where the goal can be directly discovered have a curiosity value of 10. For areas that have not yet revealed the goal but offer potential pathways to the goal, the curiosity value is set between 0 and 10 based on the VLM's imagination.

The construction and update process is illustrated in Figure 3 (c). Initially, all pixels in M^{cv}_0 are set to 10, indicating it is possible to find the target in all regions as the agent does not have any information about the entire scene. To convert the scores from the ego-centric perspective to the top-down view of the curiosity value map, we perform a projection transformation using depth information D_t and pose P_t, similar to [13] [37]. We project the navigable area with scores $Score_t$ from the ego-centric view onto a top-down map M^{nav}_t as in Figure 3 (b):

$$M^{nav}_t = Projection(Score_t) \tag{2}$$

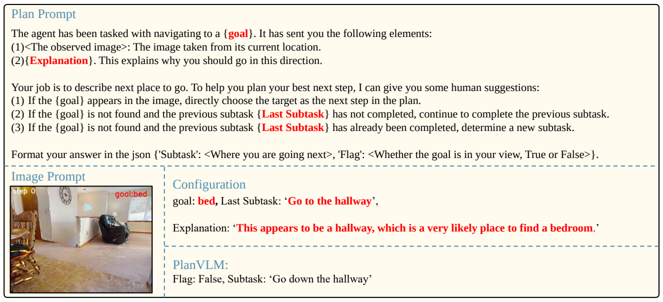
where M^{nav}_t has the same size as M^{cv}. Then M^{cv}_t (s_t in Figure 2) is updated by combining M^{nav}_t with the curiosity value map in the previous step M^{cv}_{t-1} (s_{t-1} in Figure 2):

$$M^{cv}_t(u,v) = \min(M^{cv}_{t-1}(u,v), M^{nav}_t(u,v)) \tag{3}$$

3) Cost: The cost module in the world model is used to provide environmental rewards. In our work, we use the subtask and the goal flag as the cost. For the detailed generation process of subtasks, refer to the Subtask Decomposition section (see Section III-D). The goal flag indicates whether the PlanVLM finds the goal in the selected image and is set to either True or False. The cost is fed into PlanVLM and ReasonVLM as part of their prompts to implicitly optimize the outputs in the navigation policy. PlanVLM receives the subtask from the previous step, and ReasonVLM receives the current subtask and the goal flag to switch between two configuration states.

D. Subtask Decomposition

Planning a route to an unseen goal in a completely unknown environment is a challenging task because it is hard for the model to get dense rewards. Decomposing the final goal and identifying an intermediate subtask at each step is helpful. For instance, when searching for a bed, an efficient approach would be first finding the corridor leading to the bedroom, then reaching the bedroom door, and finally approaching the bed. We adopt a subtask decomposition strategy to obtain more feedback from the environment. After the updated curiosity value map is stored in the memory,



Plan Prompt

The agent has been tasked with navigating to a {**goal**}. It has sent you the following elements:
(1)<The observed image>: The image taken from its current location.
(2){**Explanation**}. This explains why you should go in this direction.

Your job is to describe next place to go. To help you plan your best next step, I can give you some human suggestions:
(1) If the {goal} appears in the image, directly choose the target as the next step in the plan.
(2) If the {goal} is not found and the previous subtask {**Last Subtask**} has not completed, continue to complete the previous subtask.
(3) If the {goal} is not found and the previous subtask {**Last Subtask**} has already been completed, determine a new subtask.

Format your answer in the json {'Subtask': <Where you are going next>, 'Flag': <Whether the goal is in your view, True or False>}.

Image Prompt

Configuration

goal: **bed,** Last Subtask: '**Go to the hallway**',

Explanation: '**This appears to be a hallway, which is a very likely place to find a bedroom**.'

PlanVLM:

Flag: False, Subtask: 'Go down the hallway'

Fig. 4: **Plan the Route.** Text prompt is configured by the previous step's subtask, the explanation for selecting the highest-scoring image, and the goal. Using the image with the highest curiosity value as the image prompt, the VLM is invoked to plan the agent's new subtask and detect the goal.

it is projected back to the current navigable areas, and the average curiosity value for each direction is calculated to obtain the final curiosity value score $\overline{Score_t}$. We then select the navigable region corresponding to the direction $\overline{\alpha}$ with the highest score. After selecting a direction $\overline{\alpha}$ based on the curiosity value, we perform more specific planning on this image as shown in Figure 4. Specifically, we input the selected image $I_t(\overline{\alpha})$ and the previous subtask $SubTask_{t-1}$ into PlanVLM and it outputs a new subtask $SubTask_t$ and the goal flag $GoalFlag_t$ (c_t in Figure 2):

$$SubTask_t, GoalFlag_t = PlanVLM(I_t(\overline{\alpha}), SubTask_{t-1}) \tag{4}$$

E. Two-stage Action Proposer

We employ an action proposer [13] to prepare action choices for ReasonVLM. The polar coordinate action space is sampled from the navigable area. For the selected navigable area in the image $I_t(\overline{\alpha})$, we first sample K vectors at regular angular intervals $\Delta\theta$ within the navigable area and map them to the agent's coordinate system to generate an initial action sequence $A_t^{init} = \{(r_{t,i}, \theta_{t,i})\}_{i=1}^{K}$. Then, actions falling within explored regions are filtered out based on the exploration state map, and the action sequence is further refined by limiting the movement distance and angular intervals. This results in a final candidate action sequence consisting of K' actions. The candidate action sequence $A_t^{cand} = \{(r_{t,j}, \theta_{t,j})\}_{j=1}^{K'}$ in the agent's coordinate system is then mapped back to the image and annotated to obtain I_t^{ann}, as shown in Figure 5.

In order to make the agent more purposeful when choosing actions and reduce confusion of VLMs in accurately estimating the distance of objects in images, we divide the entire action-decision process into two stages as in Figure 5. The first stage is the exploration stage, where the task is to explore the regions most likely to contain the goal, ultimately discovering and accurately locating its position. The second stage is the goal-approaching stage, where the task is to move as close as possible to the goal location and eventually stop at the goal.

1) Exploration Stage: Given $SubTask_t$ and I_t^{ann}, to satisfy the requirements of the subtask $SubTask_t$, the VLM selects the most appropriate action $\overline{a}_t$ from the action sequence A_t^{cand} in I_t^{ann} for execution:

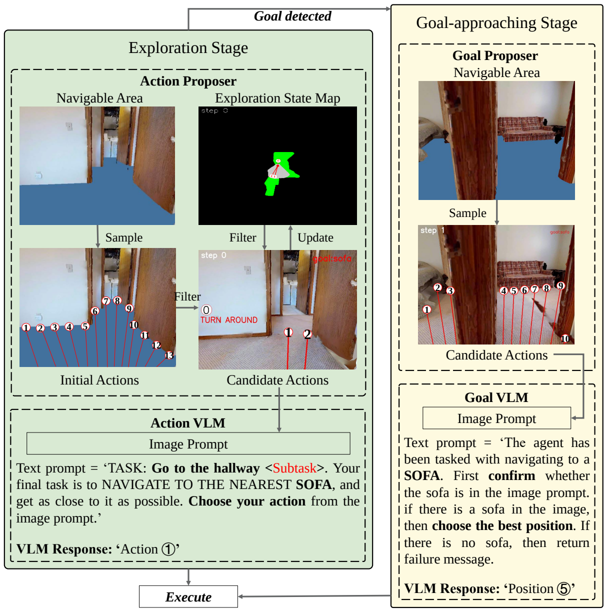
$$\overline{a}_t = ReasonVLM(SubTask_t, Goal_t, I_t^{ann}) \tag{5}$$

The final action satisfies $\overline{a}_t = (\overline{r}_t, \overline{\theta}_t)$.

2) Goal-approaching Stage: Due to the limitations of the existing VLMs' capability, we do not rely on the VLM to estimate the stopping condition directly from the observed image. Instead, we employ a strategy similar to the action proposer to determine the precise location of the goal when the goal appears in the current observation to make the stopping condition more reliable. The length constraint on the polar coordinate vectors is removed, and sampling in the navigable regions is made denser to ensure the presence of vectors that lead to the goal's ground location at the end. This strategy allows for accurate localization of the target. The agent's stopping condition is set as follows:

$$StopFlag = \begin{cases} True, & if\ DistanceToGoal < d_{thres} \\ False, & else. \end{cases} \tag{6}$$

where $DistanceToGoal$ is the Euclidean distance between the current position and the goal position.



Text prompt = 'TASK: **Go to the hallway** <Subtask>. Your final task is to NAVIGATE TO THE NEAREST **SOFA**, and get as close to it as possible. **Choose your action** from the image prompt.'

VLM Response: 'Action ①'

Text prompt = 'The agent has been tasked with navigating to a **SOFA**. First **confirm** whether the sofa is in the image prompt. if there is a sofa in the image, then **choose the best position**. If there is no sofa, then return failure message.

VLM Response: 'Position ⑤'

Fig. 5: **Reason the Action.** In the exploration stage, the agent uses the action proposer to filter sampled actions. ActionVLM(obtained by configuring ReasonVLM) selects the most appropriate action for execution from the image with a labeled candidate action sequence, continuing until the target is found and the agent shifts to the next stage. The next stage is the goal-approaching stage. The agent uses the Goal Proposer to densely sample actions from the image. The GoalVLM(also obtained by configuring ReasonVLM) then selects the action that best represents the goal location.

TABLE I: **Zero-shot object navigation results** on HM3D v0.1 [38] and MP3D [19] benchmarks. TF refers to training-free, and ZS refers to zero-shot.

Model	**TF**	**ZS**	**HM3D**		**MP3D**	
			SR(%)↑	SPL(%)↑	SR(%)↑	SPL(%)↑
Habitat-Web [22]	✗	✗	41.5	16.0	31.6	8.5
OVRL [10]	✗	✗	-	-	28.6	7.4
OVRL-V2 [11]	✗	✗	64.7	28.1	-	-
ZSON [2]	✗	✓	25.5	12.6	15.3	4.8
PSL [39]	✗	✓	42.4	19.2	18.9	6.4
PixNav [12]	✗	✓	37.9	20.5	-	-
SGM [40]	✗	✓	60.2	30.8	37.7	14.7
VLFM [14]	✗	✓	52.5	30.4	36.4	17.5
CoW [4]	✓	✓	-	-	9.2	4.9
ESC [5]	✓	✓	39.2	22.3	28.7	14.2
L3MVN [27]	✓	✓	50.4	23.1	-	-
VoroNav [7]	✓	✓	42.0	26.0	-	-
OpenFMNav [8]	✓	✓	54.9	24.4	-	-
TopV-Nav [9]	✓	✓	45.9	28.0	31.9	16.1
WMNav(Ours)	✓	✓	**58.1**	**31.2**	**45.4**	**17.2**

IV. Experiments

A. Datasets and Evaluation Metrics

Datasets The HM3D v0.1 [38] is used in the Habitat 2022 ObjectNav challenge, providing 2000 validation episodes on 20 validation environments with 6 goal object categories. The HM3D v0.2 [38] is the new version of HM3D with higher quality. It improves geometry and semantic labels with typo fixes, painting error corrections, and has 1000 validation episodes. MP3D [19] contains 11 high-fidelity scenes and 2195 episodes for validation, with 21 categories of object goals.

Metrics We adopt Success Rate (SR) and Success Rate Weighted by Inverse Path Length (SPL) as the evaluation metrics. SR represents the percentage of episodes that are completed. SPL quantifies the agent's navigation efficiency by calculating the inverse ratio of the actual path length traversed to the optimal path length weighted by success rate.

B. Implementation Details

We set 40 as the agent's maximal navigation steps. The agent adopts a cylindrical body of radius 0.18m and height 0.88m. We equip the agent with an egocentric RGB-D camera with resolution 640×480 and an HFoV of 79°. The camera is tilted down with a pitch of $14°$, which helps determine navigable areas. d_{thres} is set to 1.0, which means if the agent stops when the distance to the goal is less than 0.1m, the episode is considered successful. We mainly use Gemini VLM for our experiments, given its low cost and high effectiveness.

C. Comparison with SOTA Methods

In this section, we compare our WMNav method with the representative methods for object navigation on the MP3D [19] and HM3D [38] benchmarks. As shown in Table I, our method outperforms all the state-of-the-art zero-shot methods (+3.2% SR and +3.2% SPL on HM3D, +13.5% SR and +1.1% SPL on MP3D). Compared to all methods, including supervised methods, our approach also achieves the optimal SR on MP3D and the best SPL on HM3D, demonstrating the effectiveness of our method.

TABLE II: Ablation study of different modules and memory strategies on HM3D v0.2 [38]. SD refers to the subtask decomposition, TAP refers to the Two-stage Action Proposer strategy, No refers to No Memory, Text-Image refers to Text-Image Memory, and CVM refers to Curiosity Value Map.

	Memory	SD	TAP	SR(%)↑	SPL(%)↑
a	No	✗	✗	65.8	25.8
b	No	✓	✗	67.4	33.1
c	Text-Image	✓	✗	62.0	29.6
d	CVM(Ours)	✗	✗	69.5	**34.9**
e	CVM(Ours)	✓	✗	70.1	34.7
f	CVM(Ours)	✓	✓	**72.2**	33.3

Most existing zero-shot methods utilize carefully constructed semantic maps to obtain spatial layout information and utilize Foundation Models (LLMs or VLMs) for commonsense reasoning. Both ESC [5] and L3MVN[27] adopt the frontier map exploration strategy and leverage an LLM to select appropriate frontier points. ESC [5] converts the image into text scene information for LLM to reason. But textual information cannot accurately describe the spatial relationships in the scene, and it is difficult for LLM to make good spatial decisions. OpenFMNav [8] uses detectors (Grounding DINO[41] and SAM[42]) and VLMs to process the image to obtain an informative semantic map. Yet, building a fine map with the help of detectors is too complicated and does not fully stimulate the VLM's ability to understand the scene. TopV-Nav [9] directly takes the top-view map as the VLM's input to utilize the complete spatial information. However, since VLM is trained on egocentric image data, it does not take advantage of VLM's powerful egocentric reasoning ability. Similar to the frontier map, our simple and online maintained Curiosity Value Map, without prior information from other detectors, makes full use of the scene semantic understanding and predicting ability of VLM to implicitly encode the spatial layout of the scene and predict the results, thus enhancing the navigation efficiency in the unknown environment.

What's more, methods based on LLM or VLM suffer from hallucination. Therefore, we decompose the task into multiple subtasks and send the previous subtask to VLMs as a cost to reduce misleading hallucinations without fine-tuning the VLM. Our two-stage action proposer strategy avoids reliance on local policy and learning-based policy while utilizing the powerful spatial reasoning ability of VLM. So, the agent only needs a VLM base to complete all the processes without any policy modules to train.

TABLE III: Ablation study of VLMs on HM3D v0.1 [38].

VLM	SR(%)↑	SPL(%)↑
Qwen2.5-VL-3B	29.7	15.8
Qwen2.5-VL-7B	46.1	20.7
Gemini 1.5 Flash	53.5	27.0
Gemini 2.0 Flash	57.9	30.7
Gemini 1.5 Pro	**58.1**	**31.2**

D. Ablation Study

Effect of Different Modules. To manifest the contribution of each module, we compare three ablation models on HM3D V0.2 as it is more representative. Removing TAP implies using only the Action Proposer without computing the goal location after the goal is detected and relying on a stoppingVLM to directly determine the stopping condition according to the observation. As indicated in Table II, a and b, a and d, e and f respectively show the effectiveness of modules SubTask Decomposition, Curiosity Value Map, and Two-stage Action Proposer for improving navigation performance.

Effect of different VLMs. We further evaluate the abilities of different VLMs in navigation as shown in Table III, including open-source models and proprietary models. For each row of the table, all the VLMs used in the framework are guaranteed to be of the same model. The comparison between the 3B and 7B models of Qwen2.5-VL [43] shows that an increase in the model's scale can effectively enhance the agent's capabilities. Comparison between Qwen2.5-VL and the Gemini series models reveals that current open-source multimodal models still lag behind advanced proprietary models in terms of ZSON. Gemini 1.5 Pro demonstrates superior capability in the task. Note that when using the smaller Gemini 1.5 Flash VLM, our approach still achieves competitive performance compared to other methods on HM3D, which means our framework is practical on its own rather than relying solely on the capabilities of VLM. What's more, With the evolution of open-source and proprietary VLMs, the ability of each module of our model can be significantly enhanced, and as a whole, it has the potential to show better performance.

Effect of different memory strategies. As shown in the Table II, we also explore the influence of different memory strategies. No Memory represents that no map memory is used. The Text-Image Memory strategy first uses the VLM to generate textual descriptions of the observation. It constructs a top-down trajectory map and then inputs it as a prompt to the VLM for planning. Curiosity Value Map is our method, which uses a curiosity value map to function as memory. These three strategies all employ SD but do not employ TAP (corresponding to b, c, and e, respectively). The text-image memory strategy performs even worse than the no-map memory strategy. This is because directly feeding the VLM with a text-image combination can easily induce hallucinations, leading to erroneous memory information. Our method shows improvements in both SR and SPL metrics, as the quantitative construction of the Curiosity Value Map forces the VLM to produce output as rigorously as possible and guarantee the accuracy of each call, ensuring reliable memory.

V. Conclusion

We have introduced WMNav, which finds a novel direction for object goal navigation in unknown environments by leveraging VLMs in a world model framework and significantly enhancing ZSON performance. Our method addresses serious inefficiency problems of back-and-forth redundant movement by employing an online Curiosity Value Map to quantitatively predict the likelihood of the target's presence. The subtask decomposition module provides a denser reward for the prompt-based optimization of the policy module. Moreover, the two-stage action proposer strategy leads to more purposeful navigation and efficient exploration. By constructing the world model architecture based on VLMs, concise memory map building, and task breakdown, WMNav indicates a new optimization direction for the ZSON task and opens up new pathways for embodied robots to interact with environments.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 62373356 and the Joint Funds of the National Natural Science Foundation of China under U24B20162.

References

[1] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv preprint arXiv:2006.13171*, 2020.

[2] A. Majumdar, G. Aggarwal, B. S. Devnani, J. Hoffman, and D. Batra, "ZSON: Zero-shot object-goal navigation using multimodal goal embeddings," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=VY1dqOF2RjC

[3] H. Chen, R. Xu, S. Cheng, P. A. Vela, and D. Xu, "Zero-shot object searching using large-scale object relationship prior," 2023.

[4] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 171–23 181.

[5] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," *arXiv preprint arXiv:2301.13166*, 2023.

[6] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, "How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers," 2023.

[7] P. Wu, Y. Mu, B. Wu, Y. Hou, J. Ma, S. Zhang, and C. Liu, "Voronav: Voronoi-based zero-shot object navigation with large language model," *arXiv preprint arXiv:2401.02695*, 2024.

[8] Y. Kuang, H. Lin, and M. Jiang, "Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models," *arXiv preprint arXiv:2402.10670*, 2024.

[9] L. Zhong, C. Gao, Z. Ding, Y. Liao, and S. Liu, "Topv-nav: Unlocking the top-view spatial reasoning potential of mllm for zero-shot object navigation," *arXiv preprint arXiv:2411.16425*, 2024.

[10] K. Yadav, R. Ramrakhya, A. Majumdar, V.-P. Berges, S. Kuhar, D. Batra, A. Baevski, and O. Maksymets, "Offline visual representation learning for embodied navigation," in *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[11] K. Yadav, A. Majumdar, R. Ramrakhya, N. Yokoyama, A. Baevski, Z. Kira, O. Maksymets, and D. Batra, “Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav,” *arXiv preprint arXiv:2303.07798*, 2023.

[12] W. Cai, S. Huang, G. Cheng, Y. Long, P. Gao, C. Sun, and H. Dong, “Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5228–5234.

[13] D. Goetting, H. G. Singh, and A. Loquercio, “End-to-end navigation with vision language models: Transforming spatial reasoning into question-answering,” *arXiv preprint arXiv:2411.05755*, 2024.

[14] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, “Vlfm: Vision-language frontier maps for zero-shot semantic navigation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 42–48.

[15] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[16] P. Zhang, Y. Su, P. Wu, D. An, L. Zhang, Z. Wang, D. Wang, Y. Ding, B. Zhao, and X. Li, “Cross from left to right brain: Adaptive text dreamer for vision-and-language navigation,” *arXiv preprint arXiv:2505.20897*, 2025.

[17] Y. Gu, B. Zheng, B. Gou, K. Zhang, C. Chang, S. Srivastava, Y. Xie, P. Qi, H. Sun, and Y. Su, “Is your llm secretly a world model of the internet? model-based planning for web agents,” *arXiv preprint arXiv:2411.06559*, 2024.

[18] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang *et al.*, “Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai,” *arXiv preprint arXiv:2109.08238*, 2021.

[19] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3D: Learning from RGB-D data in indoor environments,” *International Conference on 3D Vision (3DV)*, 2017.

[20] O. Maksymets, V. Cartillier, A. Gokaslan, E. Wijmans, W. Galuba, S. Lee, and D. Batra, “Thda: Treasure hunt data augmentation for semantic navigation,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 354–15 363.

[21] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, “Simple but effective: Clip embeddings for embodied ai,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[22] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, “Habitat-web: Learning embodied object-search strategies from human demonstrations at scale,” in *CVPR*, 2022.

[23] P. Chen, D. Ji, K. Lin, W. Hu, W. Huang, T. H. Li, M. Tan, and C. Gan, “Learning active camera for multi-object navigation,” in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=iH4eyI5A7o

[24] S. Y. Min, D. S. Chaplot, P. K. Ravikumar, Y. Bisk, and R. Salakhutdinov, “FILM: Following instructions in language with modular methods,” in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=qI4542Y2s1D

[25] K. Zheng, K. Zhou, J. Gu, Y. Fan, J. Wang, Z. Li, X. He, and X. E. Wang, “Jarvis: A neuro-symbolic commonsense reasoning framework for conversational embodied agents,” *arXiv preprint arXiv:2208.13266*, 2022.

[26] Z. Al-Halah, S. K. Ramakrishnan, and K. Grauman, “Zero experience required: Plug & play modular transfer learning for semantic visual navigation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 031–17 041.

[27] B. Yu, H. Kasaei, and M. Cao, “L3mvn: Leveraging large language models for visual target navigation,” *arXiv preprint arXiv:2304.05501*, 2023.

[28] D. Shah, M. Equi, B. Osinski, F. Xia, B. Ichter, and S. Levine, “Navigation with large language models: Semantic guesswork as a heuristic for planning,” in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=PsV65r0itpo

[29] W. Cai, S. Huang, G. Cheng, Y. Long, P. Gao, C. Sun, and H. Dong, “Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill,” *arXiv preprint arXiv:2309.10309*, 2023.

[30] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang *et al.*, “Grounded language-image pre-training,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 965–10 975.

[31] X. Guo, R. Zhang, Y. Duan, Y. He, D. Nie, W. Huang, C. Zhang, S. Liu, H. Zhao, and L. Chen, “Surds: Benchmarking spatial understanding and reasoning in driving scenarios with vision language models,” *arXiv preprint arXiv:2411.13112*, 2024.

[32] R. S. Sutton, “Dyna, an integrated architecture for learning, planning, and reacting,” *ACM Sigart Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.

[33] D. Ha and J. Schmidhuber, “World models,” *ArXiv preprint*, vol. abs/1803.10122, 2018. [Online]. Available: https://arxiv.org/abs/1803.10122

[34] T. M. Moerland, J. Broekens, A. Plaat, C. M. Jonker *et al.*, “Model-based reinforcement learning: A survey,” *Foundations and Trends® in Machine Learning*, vol. 16, no. 1, pp. 1–118, 2023.

[35] D. Kim, J. Lee, J. Park, and M. Seo, “Cognitive map for language models: Optimal planning via verbally representing the world model,” *ArXiv preprint*, vol. abs/2406.15275, 2024. [Online]. Available: https://arxiv.org/abs/2406.15275

[36] H. Chae, N. Kim, K. T.-i. Ong, M. Gwak, G. Song, J. Kim, S. Kim, D. Lee, and J. Yeo, “Web agents with world models: Learning and leveraging environment dynamics in web navigation,” *arXiv preprint arXiv:2410.13232*, 2024.

[37] P. Zhang, X. Gao, Y. Wu, K. Liu, D. Wang, Z. Wang, B. Zhao, Y. Ding, and X. Li, “Moma-kitchen: A 100k+ benchmark for affordance-grounded last-mile navigation in mobile manipulation,” *arXiv preprint arXiv:2503.11081*, 2025.

[38] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra, “Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI,” in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [Online]. Available: https://openreview.net/forum?id=-v4OuqNs5P

[39] X. Sun, L. Liu, H. Zhi, R. Qiu, and J. Liang, “Prioritized semantic learning for zero-shot instance navigation,” in *European Conference on Computer Vision*. Springer, 2024, pp. 161–178.

[40] S. Zhang, X. Yu, X. Song, X. Wang, and S. Jiang, “Imagine before go: Self-supervised generative map for object goal navigation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 414–16 425.

[41] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” *arXiv preprint arXiv:2303.05499*, 2023.

[42] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[43] Q. Team, “Qwen2 technical report,” *arXiv preprint arXiv:2407.10671*, 2024.