

RANGER: A Monocular Zero-Shot Semantic Navigation Framework through Visual Contextual Adaptation

Ming-Ming Yu^{1,2}, Yi Chen³, Börje F. Karlsson^{2†}, and Wenjun Wu^{1,4†}

Abstract—Efficient target localization and autonomous navigation in complex environments are fundamental to real-world embodied applications. While recent advances in multimodal foundation models have enabled zero-shot object goal navigation, allowing robots to search for arbitrary objects without fine-tuning, existing methods face two key limitations: (1) heavy reliance on ground-truth depth and pose information, which restricts applicability in real-world scenarios; and (2) lack of visual in-context learning (VICL) capability to extract geometric and semantic priors from environmental context, as in a short traversal video. To address these challenges, we propose RANGER, a novel zero-shot, open-vocabulary semantic navigation framework that operates using only a monocular camera. Leveraging powerful 3D foundation models, RANGER eliminates the dependency on depth and pose while exhibiting strong VICL capability. By simply observing a short video of the target environment, the system can also significantly improve task efficiency without requiring architectural modifications or task-specific retraining. The framework integrates several key components: keyframe-based 3D reconstruction, semantic point cloud generation, vision-language model (VLM)-driven exploration value estimation, high-level adaptive waypoint selection, and low-level action execution. Experiments on the HM3D benchmark and real-world environments demonstrate that RANGER achieves competitive performance in terms of navigation success rate and exploration efficiency, while showing superior VICL adaptability, with no previous 3D mapping of the environment required.

I. INTRODUCTION

Efficient visual target localization and autonomous navigation in complex environments are core capabilities of embodied agents, with broad applications in service robotics, human-robot interaction, and autonomous monitoring systems. With the recent development of multimodal foundation models, including large language models (LLMs) (*e.g.*, [1]–[6]) and open-vocabulary vision models (*e.g.*, [7]–[9]), the performance of zero-shot object navigation (*e.g.*, [10]–[15]) has significantly improved. Such methods enable robots to be capable of navigating towards arbitrary target objects without task-specific fine-tuning. Specifically, they leverage open-vocabulary perception to recognize arbitrary targets and semantic reasoning to guide navigation. This zero-shot paradigm enables immediate deployment in unseen environments without the need for environment-specific re-training.

However, existing approaches still face major limitations. First, most methods rely heavily on depth sensing and precise pose estimation, usually involving global

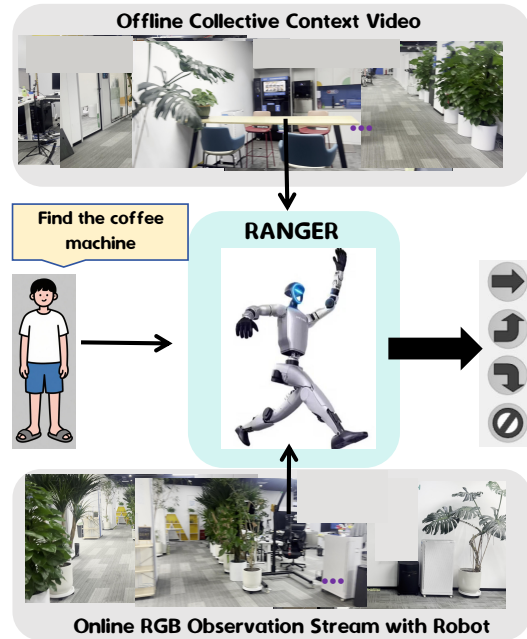


Fig. 1: RANGER Workflow: Given an offline video of a new environment captured with a camera (optionally) and an online RGB observation stream from the robot, RANGER efficiently adapts to the new environment and completes navigation tasks based on human instructions.

mapping/reconstruction, which are typically feasible only in simulation or with specialized hardware. In real-world scenarios, sensor noise, calibration errors, data collection, and hardware costs substantially restrict their practicality, similarly to indoor localization scenarios [16]. Second, current systems have yet to better leverage contextual knowledge available in everyday video streams, which can provide valuable priors for efficient exploration and rapid adaptation to novel environments.

This naturally raises a fundamental question: *Can a navigation framework be built relying solely on RGB input?* Such a framework would be inherently low-cost, easy to deploy, and capable of directly utilizing videos captured from arbitrary devices for rapid adaptation to new environments. Notably, recent data-driven methods such as MAST3R [17], Dust3R [18], Fast3R [19], and VGGT [20] have demonstrated remarkable performance and generalization in 3D reconstruction tasks, opening new opportunities to achieve efficient navigation with RGB-only input.

In this paper, we introduce **RANGER**: a zero-shot, open-vocabulary semantic navigation framework based on monocular RGB input. Our approach requires neither depth sensors

¹Beihang University; ²Beijing Academy of Artificial Intelligence; ³Institute of Automation, Chinese Academy of Sciences; ⁴Hangzhou International Innovation Institute, Beihang University. [†]Corresponding authors. Emails: {mingmingyu, wwj}@buaa.edu.cn, borje@baai.ac.cn

nor precise robot pose estimation, yet enables semantic-driven exploration and navigation in unseen environments. Furthermore, RANGER can adapt to a new environment using only a short offline video to extract rich geometric and semantic priors, significantly reducing exploration costs and enhancing navigation efficiency. Importantly, our method does not require any task-specific fine-tuning during navigation (see Fig. 1). Specifically, RANGER is built upon four key components: (1) online 3D reconstruction and localization from RGB-only input, (2) cross-frame semantic point cloud fusion, (3) high-level waypoint selection guided by vision–language models (VLMs), and (4) low-level action execution. At its core lies a *keyframe-based memory bank*, which encodes geometric, semantic, and exploration-value information of the environment. This memory is continuously curated and optimized in the backend to reduce errors and improve system robustness.

We conduct systematic evaluations on the large-scale HM3D dataset, demonstrating that RANGER achieves competitive performance without access to depth sensing, precise poses, or global maps. In addition, we validate the practicality and robustness of our system in real-world environments such as meeting rooms and offices with a humanoid robot deployment. Our contributions are as follows: (1) A zero-shot semantic navigation framework (RANGER) that operates solely on monocular RGB input, enabling rapid adaptation via offline videos. (2) A novel system architecture integrating RGB-based 3D reconstruction with VLM-guided hierarchical planning through a dynamic keyframe-based memory bank. (3) Extensive evaluations on HM3D and real-world humanoid robots, showing competitive performance and superior VICL adaptability.

II. RELATED WORK

Object Goal Navigation. Current object-goal navigation methods include: (1) Learning-based approaches that train navigation policies over visual features extracted by pre-trained encoders via extensive imitation learning (*e.g.*, [21]–[27]). However, these methods often overfit to simulations and are restricted to predefined object categories, limiting open-world generalization. (2) Approaches that focus on Zero-Shot Object Navigation (*e.g.*, [10], [11], [13]–[15], [28], [29]). These methods explicitly construct environmental maps and utilize VLMs [1], [3] or LLMs [4], [5] to select the most valuable frontier or waypoint for exploration. For instance, in Cow [11], the robot explores the nearest frontier point until the target is detected using CLIP features [30] and open-vocabulary object detectors [31]. ESC [32], L3MVN [12], and Voronav leverage LLMs for reasoning and decision-making to enhance performance. VLFM [14] employs a VLM to build semantic values on the map based on first-person observations and textual prompts, selecting the highest-scoring frontier. Their reliance on depth information, precise pose estimation, and task-specific fine-tuning often limits their performance in real-world scenarios, where depth data may be inaccurate or unavailable. These methods also hinder the ability to perform contextual

learning, making it challenging for robots to leverage freely available monocular videos to improve exploration efficiency and adapt to new environments.

Monocular 3D Reconstruction. Monocular 3D reconstruction methods have made notable progress recently in enabling the building of 3D maps from single camera inputs. Techniques such as MAST3R [17], Dust3R [18], Fast3R [19], VGGT [20], MAST3R-SLAM [33], and VGGT-SLAM [34] have contributed significantly to this field. However, these methods often rely on passive reconstruction, where the trajectory is pre-collected and does not involve active perception or exploration. This restricts the system’s ability to adapt to dynamic environments and limits its exploration capabilities. In contrast, our approach tightly integrates incremental monocular 3D reconstruction with active semantic exploration to enable autonomous navigation.

In-Context Learning. In-context learning (ICL) [35] allows models to perform tasks by leveraging a few examples provided in the input context, without requiring task-specific retraining. Recent work has extended this idea to robotics, where robots are trained to perform navigation tasks based on contextual information, such as video or textual prompts. For example, several works have used context from videos or previous maps to train navigation models [36]. However, these methods often require the target object to appear in the context, which severely limits their practical application, especially when the object is not initially visible. In contrast, our approach does not require the target object to appear, even in the collected offline video. Instead, our approach enables robots to adaptively decide whether to continue exploring or exploit the knowledge contained in the offline video, remaining fully functional even without any prior visual context. This adaptive approach makes RANGER more versatile, allowing it to handle a broader range of navigation scenarios.

III. PROBLEM FORMULATION

We consider two semantic navigation tasks under a **zero-shot** setting, where the agent must locate a target object in a novel environment without task-specific training. These tasks differ in the availability of contextual information: the first relies solely on online RGB observations, while the second leverages an offline traversal video as prior context.

RGB Navigation. In the standard *ObjectNav* task with RGB-only inputs, an agent must navigate an unseen environment to approach a target object specified by a human-provided instruction. We denote the set of object categories as $\mathcal{C} = \{c_1, \dots, c_M\}$ and the set of scenes as $\mathcal{S} = \{s_1, \dots, s_N\}$.¹ Each episode is defined by a tuple $\mathcal{T}_i = \langle s_i, c_i, p_i \rangle$, where the agent starts from an initial pose p_i in scene s_i with target category $c_i \in \mathcal{C}$. At each time step t , the agent receives an RGB observation $o_t \in \mathbb{R}^{H \times W \times 3}$ and the target category c_i , without access to depth or pose information. The action space \mathcal{A} consists of six

¹Each scene $s_i \in \mathcal{S}$ represents a unique physical environment (*e.g.*, a specific house layout) used to evaluate the agent’s navigation performance.

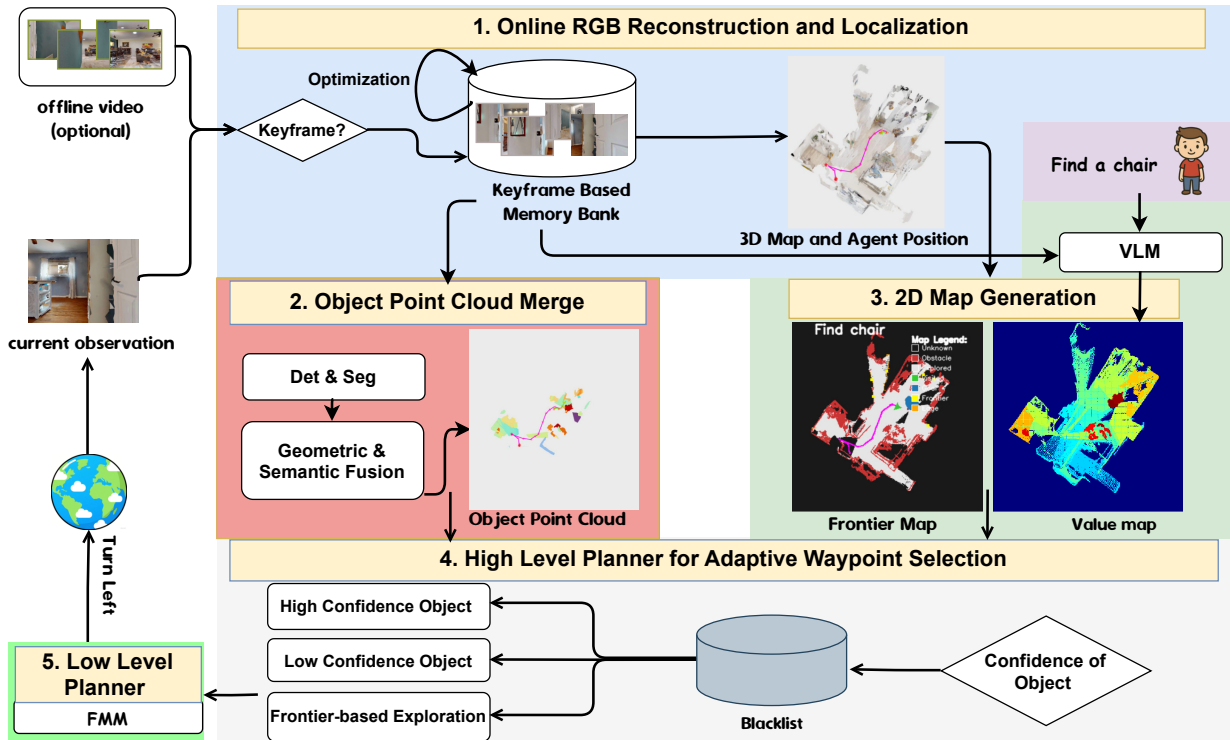


Fig. 2: Overview of RANGER: a unified architecture for zero-shot RGB-only navigation that builds geometric and semantic memory via a keyframe-based memory bank, which subsequently guides the agent’s exploration.

discrete actions: *move_forward*, *turn_left*, *turn_right*, *look_up*, *look_down*, and *stop*. An episode is considered successful if the agent executes a *stop* action when the geodesic distance to the target is below a predefined threshold, with a trajectory length within the step limit.

Contextual RGB Navigation. In the contextual variant, the agent is additionally provided with an offline traversal video V_{s_i} of a human exploring the target environment s_i . The agent must then navigate the same environment s_i to approach the target object by combining prior visual context from V_{s_i} with real-time RGB observations. Compared to the RGB-only setting, this formulation enables the agent to reduce redundant exploration and achieve more efficient semantic navigation in unseen environments.

IV. METHODOLOGY

A. Overview of the Process

As illustrated in Fig. 2, RANGER employs a unified architecture that seamlessly handles both *zero-shot RGB-only navigation* and *video-context-augmented navigation* without requiring any architectural modifications or retraining. The system maintains a dynamic keyframe bank, which serves as a joint geometric and semantic memory of the environment. Upon receiving each new RGB observation, the backend refines the agent’s pose in real-time and performs loop closure by optimizing relative poses against the stored keyframes. This process incrementally constructs a dense 3D map, enabling metric self-localization. Meanwhile, a semantic module aggregates object features via CLIP [30] and GroundingDINO [7] into a semantic point cloud, where

each point is annotated with category probabilities and spatial context. To facilitate efficient navigation, we project the 3D reconstruction into a 2D occupancy map. Based on this representation, a hierarchical planner generates high-level waypoints by jointly evaluating the target’s semantic relevance to the textual goal, along with its spatial and perceptual confidence. This perception-planning-action loop executes continuously until the agent reaches the target.

B. Online RGB Reconstruction and Localization

To enable metric-aware navigation from monocular RGB inputs, we integrate **MASt3R-SLAM** [33], a state-of-the-art real-time dense SLAM system based on the MASt3R two-view 3D reconstruction prior. Its open-source implementation is directly incorporated into RANGER as a plug-and-play geometric backbone.

Keyframe-Based Reconstruction. To build the dense scene representation, the pipeline dynamically registers keyframes when the number of valid matches falls below a specific threshold ω_k . For each frame, the MASt3R network directly predicts dense pointmaps. The system continuously fuses them into a canonical keyframe pointmap using a running weighted average filter. This sequential filtering seamlessly merges multi-view information to maintain a coherent 3D structure.

Localization. To continuously estimate the camera pose, the system first establishes fast pixel correspondences between the incoming frame and the active keyframe using an efficient iterative projective matching scheme. Based on these matches, the relative pose is then solved by minimizing the directional ray error. By formulating the problem purely

around ray geometry, this approach inherently bypasses the need for specific camera models. Consequently, it remains highly robust to structural inconsistencies and seamlessly handles uncalibrated video sequences, making it ideal for real-world deployment.

Global Consistency via Backend Optimization. To mitigate drift, MAST3R-SLAM constructs a pose graph connected by sequential tracking and loop-closure edges. Loop candidates are efficiently retrieved via the ASMK framework [37] on encoded visual features and validated through geometric matching. The system then enforces long-term consistency across all poses and dense geometry by jointly minimizing the directional ray errors across the entire graph.

Real-Scale Reconstruction. Since monocular reconstruction lacks absolute scale, we utilize the robot’s camera height h_{real} as a geometric prior. We select the lowest 20% of points along the vertical axis as candidate ground points and use RANSAC to fit a ground plane $\mathbf{n}^\top \mathbf{x} + d = 0$, where $\mathbf{n} = [a, b, c]^\top$ is the plane normal. The estimated camera height in the reconstructed space is computed as $h_{\text{est}} = \frac{|d|}{\|\mathbf{n}\|}$. Finally, we derive the global scale factor $s = h_{\text{real}}/h_{\text{est}}$ to align the 3D map with the physical environment.

C. Object Point Cloud Merge

We first employ Grounding DINO as an open-vocabulary detector and combine it with Mobile SAM [8] to extract object masks within the detection boxes of each keyframe stored in the memory bank. Then, using the keyframe confidence masks provided by MAST3R-SLAM, we filter these object regions to obtain high-confidence object point cloud representations in 3D space. For each semantic mask, we further crop the corresponding image region and extract visual features with CLIP. This process ultimately generates a semantic point cloud along with class confidence scores and target category information.

Object Association. For each newly detected object j , we compute both geometric and semantic similarity to existing objects. Geometric similarity S_{geo} is based on the Intersection over Union (IoU) of the point clouds, while visual similarity S_{vis} is determined by comparing the CLIP semantic features of the newly detected object with those of existing objects. The overall similarity score is a weighted sum:

$$s(i, j) = w_1 S_{\text{vis}}(i, j) + w_2 S_{\text{geo}}(i, j), \quad (1)$$

where w_1 and w_2 are weights. If the similarity score falls below a predefined threshold τ , a new object is initialized. Otherwise, the new detection is associated with the existing object exhibiting the highest similarity.

Object Fusion. For associated objects, their point clouds are merged, and their features are updated via a weighted average based on detection frequency. Additionally, the final confidence of the fused object is determined by the maximum confidence score across all its observations.

D. 2D Map Generation

Frontier Map. After constructing a unified 3D point cloud, we project the points onto a 2D plane to create

obstacle and exploration maps. Points above the floor are projected onto the obstacle map, while all points contribute to the exploration map. Frontiers are generated by detecting the boundaries between explored and unexplored regions. As the robot explores, the locations of frontiers dynamically change until the environment is fully explored.

Value Map. To guide the agent’s exploration, we construct a 2D value map by integrating both object-level and image-level semantic relevance. For the fused objects and each keyframe in the memory bank, we compute the cosine similarity between their CLIP embeddings \mathbf{f} and the target instruction embedding \mathbf{g} , i.e., $\frac{\mathbf{f} \cdot \mathbf{g}}{\|\mathbf{f}\| \|\mathbf{g}\|}$, denoted as matching scores s_{obj} and s_{img} respectively. The associated high-confidence 3D point clouds are then projected onto a 2D grid. The final value at each grid cell (i, j) is determined by the maximum relevance score from both sources, denoted as $V(i, j) = \max(s_{\text{obj}}, s_{\text{img}})$.

E. High-Level Planner for Adaptive Waypoint Selection

To ensure robust navigation under perceptual uncertainty, we design a high-level planner that dynamically selects waypoints by integrating *semantic relevance*, *traversability*, and *candidate confidence*. Operating on dense 2D maps and the current pose T_t provided by MAST3R-SLAM, the planner evaluates a candidate set $\mathcal{W}_t = \mathcal{O}_t \cup \mathcal{F}_t$, where \mathcal{O}_t represents object proposals and \mathcal{F}_t denotes frontier points. The selection follows a *hierarchical priority*: (1) objects exceeding σ_{obj} are final targets; (2) low-confidence objects serve as temporary waypoints for guided exploration; (3) otherwise, the frontier in \mathcal{F}_t with the highest 2D value map response is selected. To ensure *geometric feasibility*, all candidates must be navigable via the obstacle map. Crucially, a dynamic blacklist \mathcal{B}_t records two failure modes to prevent repetitive errors: (i) *unreachable locations* that remain inaccessible beyond the maximum step limit, and (ii) *perceptual false positives* where low-confidence candidates fail to meet the verification criteria despite exceeding τ_{obs} cumulative observations.

F. Low-Level Planner for Path Planning

We use the Fast Marching Method (FMM) [38] as the local planner to generate a path toward the goal. At each step, the robot executes motion commands toward a local subgoal, while the map and target are updated in real time, enabling adaptive, closed-loop navigation in partially observed environments.

V. EXPERIMENTS

We evaluate RANGER on object goal navigation tasks across both simulated and real-world environments, testing its navigation performance in completely unseen settings, as well as when provided with a contextual video.

A. Simulation-based Experiment Setup

Navigation in unseen environments. We evaluate our approach on the Habitat HM3Dv2 dataset (HM3D-Semantics-v0.2 [39], released in the 2023 Habitat Challenge), which includes six goal categories. For our experiments, we select 10

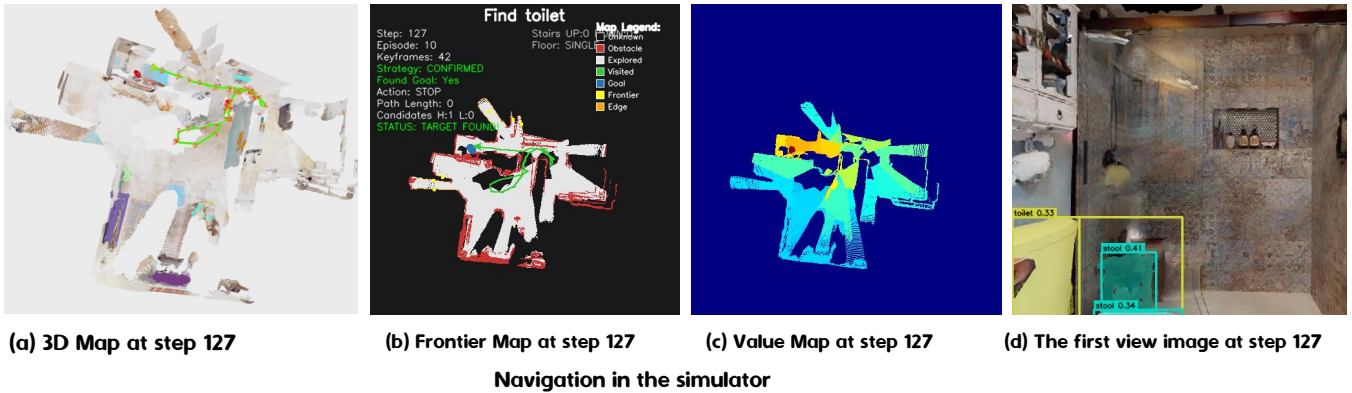


Fig. 3: Visualization of the navigation process in a simulated environment with task instruction: "Find the toilet". (a) Reconstructed 3D map and the agent's trajectory at step 127. The red cones represent keyframe locations, while the green cone indicates the agent's current position. The green line traces the agent's movement, and the red point cloud represents the toilet object. (b) The Frontier Map showing the robot's exploration trajectory. The blue area highlights the target object region, red areas indicate obstacles, and yellow dots represent frontier points. (c) The Value Map, where red regions correspond to high-value areas. (d) First-person observation.

single-floor scenes comprising 279 episodes in total. In each episode, the agent's starting position is randomly sampled to assess navigation performance in unseen environments. The *move forward* action moves the agent by 25 cm, while *turn* actions adjust orientation by 30°. Considering the trade-off between system storage and performance, we set the maximum episode length to 300 steps.

Navigation with contextual video. We use the same 10 scenes and manually collect one video trajectory per scene by teleoperating the robot in the simulator. For evaluation, the agent is initialized near the endpoint of the collected video to ensure partial overlap between the agent's initial observation and the video frames, which facilitates re-localization. For each scene, we randomly sample 2 or 3 starting positions near the video endpoint, resulting in 100 episodes in total across 6 object categories.

B. Metrics

We use navigation success rate (SR) and success rate weighted by navigation path length (SPL) as evaluation metrics. SR represents the percentage of successful episodes out of the total episodes. SPL measures the efficiency of reaching the goal in addition to the success rate.

C. Implementation Details

In Habitat, the agent uses a 640×480 RGB camera mounted at 0.88m height. For semantic segmentation, we leverage Grounding DINO [7] to generate class-agnostic object proposals and MobileSAM [8] to produce pixel-accurate masks. Semantic confidence scores for detected objects are computed using CLIP [30], enabling zero-shot semantic grounding. Our system employs consistent hyper-parameters: keyframe selection threshold $\omega_k = 0.333$, loop-closure detection threshold $\omega_l = 0.1$, and robustness regularization weight $\omega_r = 0.005$. To ensure reliable relocalization, we enforce a stricter geometric consistency check when attaching the current frame to the pose graph: the ratio

of inlier feature matches must exceed 0.3. In addition, the confidence threshold for the point cloud is set to 1.9.

VI. RESULTS

A. Navigation in Unseen Environments

Previously, zero-shot object navigation heavily relied on accurate pose and depth information provided by simulators or auxiliary hardware sensors. To evaluate our approach in more realistic settings, we compare our approach in more realistic settings, we compare with L3MVN [12], a representative work that utilizes LLMs for efficient frontier selection and requires privileged inputs (depth and pose). As detailed in Table I, L3MVN (300) and L3MVN (500) represent performance under step limits of 300 and 500, respectively, both leveraging RGB, depth, and pose data. In contrast, RANGER operates solely on RGB input, without any privileged sensing.

TABLE I: Navigation performance in unseen environments.

Methods	Sensors	SR	SPL	Avg Steps
L3MVN (300)	RGB, Depth, Pose	39.4	15.7	147.62
L3MVN (500)	RGB, Depth, Pose	42.3	16.3	171.06
RANGER (300)	RGB	42.7	17.8	172.80

As presented in Table I, RANGER achieves an SR of 42.7% and an SPL of 17.8% under the 300-step limit. This performance outperforms L3MVN (300) by 3.3 percentage points in SR. Notably, RANGER even surpasses L3MVN (500), which benefits from both an extended step limit and privileged sensor inputs. These compelling results underscore that integrating contextual video priors enables competitive navigation performance in previously unseen environments using only standard RGB input. Figure 3 visually depicts a navigation instance: the agent successfully reaches the *toilet* at step 127, with 42 frames stored in its keyframe-based memory bank. The highlighted region in Figure 3 (c) further confirms the strong spatial consistency between the agent's



Adapting navigation to new environments using only an offline video.

Fig. 4: Contextual adaptation for navigation in a new environment. RANGER can leverage an offline video to adapt and locate the target object "bed". (a-d) Initial state with offline memory; (e-h) Final navigation state. The system successfully found the target in 38 steps.

memory and the observed environment, illustrating effective visual grounding.

B. Navigation with a Contextual Video

TABLE II: Performance w/ & w/o contextual video.

Methods	SR	SPL	DTG	Avg Steps
w/o video	44.0	16.4	2.933	171.36
w/ video	58.0	30.2	2.381	110.22

Table II presents the navigation performance with and without contextual video. As shown in Table II, incorporating an offline traversal video significantly improves performance, boosting SR (44.0% \rightarrow 58.0%) and SPL (16.4 \rightarrow 30.2), while reducing average steps (171.4 \rightarrow 110.2). These results confirm that contextual videos effectively reduce redundant exploration and guide the agent toward more efficient navigation. In our framework, offline videos serve as an *environmental preview*. By constructing a keyframe-based memory bank, the system acquires prior knowledge of the scene before navigation, thereby reducing redundant exploration. This mechanism mimics the way humans navigate more efficiently in a new environment after watching a walkthrough video. In practice, we observe that providing such contextual information enables the agent to significantly reduce unnecessary exploration and complete target-finding tasks more efficiently. As illustrated in Fig. 4, when searching for a *bed*, RANGER first builds a keyframe memory bank from the offline video and applies semantic retrieval to identify the target object within the video. The object is then transformed into a point cloud representation, which is integrated with

both the offline video and real-time observations to construct an environment map. Finally, the system plans a collision-free path that directly leads the agent to the target location.

C. Ablation Studies

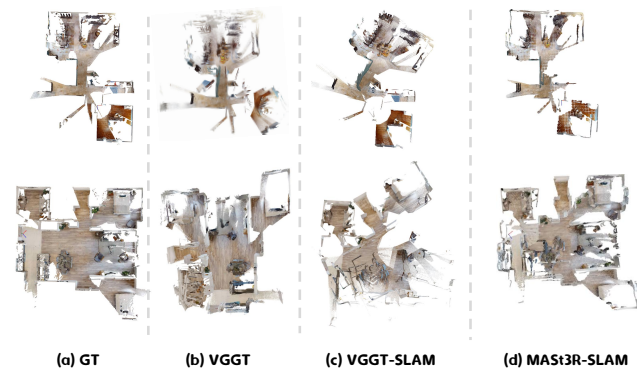


Fig. 5: Comparison of 3D maps reconstructed using different methods. Figure (a) shows the result using real depth and pose data. Figures (b), (c), and (d) present the reconstruction results of VGGT, VGGT-SLAM, and MAST3R-SLAM, respectively, all relying solely on RGB images.

Reconstruction and Localization Accuracy As shown in Fig. 5, we compare the latest 3D reconstruction methods (e.g., VGGT, VGGT-SLAM, and MAST3R-SLAM) with the ground truth. The video trajectories are collected by a manually teleoperated agent navigating within the environment. The results demonstrate that VGGT suffers from insufficient detail modeling and, due to its requirement to

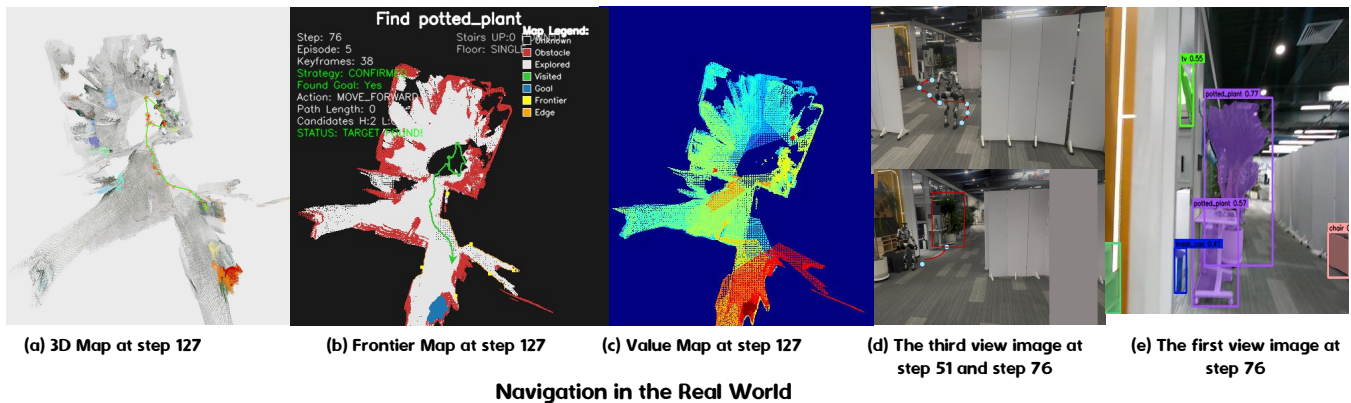


Fig. 6: Real-world navigation with RANGER on a Unitree G1 humanoid robot. Task: "Find potted plant".

process the entire video sequence at once, fails to meet the real-time demands of navigation tasks. In contrast, VGGT-SLAM constructs dense 3D maps and performs localization by incrementally aligning submaps generated by VGGT. However, in long-sequence scenarios, the submap merging process often introduces significant alignment errors, limiting its effectiveness. MAST3R-SLAM exhibits stronger robustness under the same conditions. It consistently achieves higher reconstruction quality and global consistency across trajectories of varying lengths, effectively mitigating drift and distortion issues commonly observed in long-sequence stitching. Moreover, it supports real-time operation, making it suitable for navigation applications.

Component Analysis. Table III summarizes the ablation results for two key components: the high-level planner and the object detector. When the high-level planner is removed, the SR decreases from 42.7% to 39.8%, and the average number of steps increases from 172.80 to 184.43. This confirms the critical role of the planner in efficiently guiding the agent toward the target. For the detector comparison, YOLO-World [9] and DINO yield relatively close results, with DINO showing a slight advantage (SR = 42.7%, SPL = 17.8% vs. 42.3%, 17.1%). This suggests that while replacing the detector has only a modest impact on overall accuracy, stronger visual grounding from DINO provides more consistent navigation performance.

TABLE III: Impact of model components.

component	SR	SPL	DTG	Avg Steps
w/o High Level Planner	39.8	17.6	3.552	184.43
YOLO-World	42.3	17.1	3.573	178.06
DINO	42.7	17.8	3.366	172.80

Effect of Step Limits. As the step limit increases, the memory bank must maintain a growing number of keyframe point clouds, poses, semantic segmentation, and their associated values, potentially leading to substantial storage and computation overhead. Based on this consideration, we evaluate RANGER under different maximum step limits. As shown in Table IV, the success rate (SR) improves steadily with larger step budgets, rising from 6.1% at 50 steps to 42.7% at 300 steps. This indicates that a longer execution

horizon provides the agent with more opportunities for exploration. However, the improvement becomes marginal beyond 250 steps; therefore, we set the maximum step limit to 300 in our experiments.

TABLE IV: SR under different maximum step limits.

Max Steps	50	100	200	250	300
SR	6.1	22.2	36.6	40.1	42.7

D. Navigation in the Real World

Real-World Experiment Setup. We validate the practicality of RANGER through real-world experiments on a Unitree G1 humanoid robot, equipped with an Intel RealSense D455 RGB-D camera mounted at a height of 1.29 m on a custom 2 DoF head. Importantly, *only the RGB stream is used* for sensing and navigation, consistent with our depth-free assumption. All system components—including MAST3R-SLAM for online 3D reconstruction, CLIP, and Grounding DINO for semantic grounding, and MobileSAM for instance-aware segmentation—are deployed on a mobile workstation with an NVIDIA RTX 4090 Laptop GPU and an Intel Core i9-13900HX CPU, achieving real-time inference at approximately 1 Hz. The experiments were carried out in diverse indoor environments, including laboratories and meeting rooms, where the robot was instructed to navigate toward semantically defined targets.

Visualization. Representative qualitative results are shown in Fig. 6. In the example task, the user instructs the robot to locate a *plant*. After completing exploration of the first room (step 51), the robot autonomously selects a high-value frontier region and transitions to an adjacent room, successfully detecting the target at step 76. Notably, the high-value region highlighted in Fig. 6 (c) aligns closely with the actual target location shown in Fig. 6 (b), demonstrating the effectiveness of semantic information in guiding exploration.

VII. CONCLUSIONS AND LIMITATIONS

In this work, we introduce RANGER, a novel zero-shot visual target navigation framework relying solely on RGB input. RANGER significantly enhances navigation efficiency without task-specific fine-tuning through its keyframe-based

memory bank, back-end optimization, keyframe-driven reconstruction, and hierarchical planning strategy. The system also demonstrates rapid adaptation and substantial performance gains with only a short video sequence of a new environment, offering a new perspective on zero-shot navigation. Ultimately, this work presents a practical pathway toward real-world autonomous navigation and opens new directions for research focused purely on visual input.

Limitations. Despite the strong performance and contextual learning capability of RANGER, several limitations remain. First, the continuous accumulation of keyframes increases storage overhead and may degrade system efficiency over time, highlighting the need for more effective memory management strategies. Second, reconstruction and localization remain possible bottlenecks for purely RGB-based navigation, as their robustness is still insufficient to handle overly complex or dynamic environments. Enhancing the reliability of these modules will be critical for advancing the practical deployment of such systems.

VIII. ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 62441617).

REFERENCES

- [1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [2] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *NeurIPS*, vol. 36, pp. 34 892–34 916, 2023.
- [3] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan *et al.*, “Qwen-vl: A frontier large vision-language model with versatile abilities,” *arXiv preprint arXiv:2308.12966*, vol. 1, no. 2, p. 3, 2023.
- [4] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.
- [5] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, “Gemini: a family of highly capable multimodal models,” *arXiv preprint arXiv:2312.11805*, 2023.
- [6] Y. Chen, J. Xu, X.-Y. Zhang, W.-Z. Liu, Y.-Y. Liu, and C.-L. Liu, “Recoverable compression: A multimodal vision token recovery mechanism guided by text information,” in *AAAI*, 2025, pp. 2293–2301.
- [7] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” in *ECCV*, 2024, pp. 38–55.
- [8] 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 *ICCV*, 2023, pp. 4015–4026.
- [9] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, “Yolo-world: Real-time open-vocabulary object detection,” in *CVPR*, 2024, pp. 16 901–16 911.
- [10] Y. Kuang, H. Lin, and M. Jiang, “Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models,” in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 338–351.
- [11] 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 *CVPR*, 2023, pp. 23 171–23 181.
- [12] B. Yu, H. Kasaei, and M. Cao, “L3mvm: Leveraging large language models for visual target navigation,” in *IROS*, 2023, pp. 3554–3560.
- [13] 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.
- [14] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, “Vlfn: Vision-language frontier maps for zero-shot semantic navigation,” in *ICRA*, 2024, pp. 42–48.
- [15] H. Yin, X. Xu, L. Zhao, Z. Wang, J. Zhou, and J. Lu, “Unigoal: Towards universal zero-shot goal-oriented navigation,” *CVPR*, 2025.
- [16] Y. Shu, Z. Li, B. Karlsson, Y. Lin, T. Moscibroda, and K. Shin, “Incrementally-deployable indoor navigation with automatic trace generation,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019, pp. 2395–2403.
- [17] V. Leroy, Y. Cabon, and J. Revaud, “Grounding image matching in 3d with mast3r,” in *ECCV*, 2024, pp. 71–91.
- [18] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, “Dust3r: Geometric 3d vision made easy,” in *CVPR*, 2024, pp. 20 697–20 709.
- [19] J. Yang, A. Sax, K. J. Liang, M. Henaff, H. Tang, A. Cao, J. Chai, F. Meier, and M. Feiszli, “Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass,” in *CVPR*, 2025, pp. 21 924–21 935.
- [20] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, “VGGT: Visual geometry grounded transformer,” in *CVPR*, 2025, pp. 5294–5306.
- [21] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, “Pirlnav: Pretraining with imitation and rl finetuning for objectnav,” in *CVPR*, 2023, pp. 17 896–17 906.
- [22] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, “Navid: Video-based vlm plans the next step for vision-and-language navigation,” *RSS*, 2024.
- [23] 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.
- [24] K. Ehsani, T. Gupta, R. Hendrix, J. Salvador, L. Weihs *et al.*, “Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world,” in *CVPR*, 2024, pp. 16 238–16 250.
- [25] S. Chen, T. Chabal, I. Laptev, and C. Schmid, “Object goal navigation with recursive implicit maps,” in *IROS*, 2023, pp. 7089–7096.
- [26] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, “An embodied generalist agent in 3d world,” *arXiv preprint arXiv:2311.12871*, 2023.
- [27] M.-M. Yu, F. Zhu, W. Liu, Y. Yang, Q. Wang, W. Wu, and J. Liu, “Cnav: Towards self-evolving continual object navigation in open world,” *arXiv preprint arXiv:2510.20685*, 2025.
- [28] H. Yuan, Y. Bai, Y. Fu, B. Zhou, Y. Feng, X. Xu *et al.*, “Being-0: A humanoid robotic agent with vision-language models and modular skills,” *arXiv preprint arXiv:2503.12533*, 2025.
- [29] B. Yu, Y. Liu, L. Han, H. Kasaei, T. Li, and M. Cao, “Vln-game: Vision-language equilibrium search for zero-shot semantic navigation,” *arXiv preprint arXiv:2411.11609*, 2024.
- [30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *ICML*, 2021, pp. 8748–8763.
- [31] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *ICCV*, 2021, pp. 9650–9660.
- [32] 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,” in *ICML*, 2023, pp. 42 829–42 842.
- [33] R. Murai, E. Dexheimer, and A. J. Davison, “MAS3R-SLAM: Real-time dense SLAM with 3D reconstruction priors,” in *CVPR*, 2025, pp. 16 695–16 705.
- [34] D. Maggio, H. Lim, and L. Carlone, “VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold,” *arXiv preprint arXiv:2505.12549*, 2025.
- [35] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *NeurIPS*, vol. 33, pp. 1877–1901, 2020.
- [36] B. Zhou, Z. Zhang, J. Wang, and Z. Lu, “Nolo: Navigate only look once,” in *IROS*, 2025, pp. 17 162–17 169.
- [37] G. Toliás, Y. Avrithis, and H. Jégou, “To aggregate or not to aggregate: Selective match kernels for image search,” in *ICCV*, 2013, pp. 1401–1408.
- [38] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts,” *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.
- [39] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre *et al.*, “Habitat-matterport 3d semantics dataset,” in *CVPR*, 2023, pp. 4927–4936.