**Robotics: Science and Systems 2026**
**Sydney, Australia, July 13–17, 2026**

# MVP-Nav: Multi-layer Value Map Planner Navigator

Wenyuan Xie, Shaokai Wu, Yijin Zhou, Yanbiao Ji, Guodong Zhang,
Bayram Bayramli, Qiuchang Li, Xunchu Zhou, Yue Ding*, and Hongtao Lu*
Shanghai Jiao Tong University
*Corresponding authors: dingyue@sjtu.edu.cn, htlu@sjtu.edu.cn

***Abstract*—Zero-shot Object Goal Navigation (ZSON) with RGB-only perception poses a fundamental challenge for embodied agents, as the absence of explicit depth information introduces severe physical uncertainty and semantic–physical misalignment. Existing approaches either rely on high-level semantic reasoning without geometric grounding or learn end-to-end policies that lack explicit physical constraints, often resulting in semantically plausible but physically unsafe behaviors. In this paper, we propose MVP-Nav, a physical-aware RGB-only navigation framework that aligns perception, planning, and control with the real 3D world. MVP-Nav reconstructs explicit physical occupancy from monocular observations by leveraging 3D foundation models to project 2D semantic instances into 3D oriented bounding boxes, forming a global spatial semantic representation. To unify high-level semantic reasoning and low-level physical constraints, we introduce a Multi-layer Value Map (MVM) that integrates semantic priorities and reconstructed geometry into a shared cost space, enabling physically grounded geometric planning. Extensive experiments on zero-shot object navigation benchmarks demonstrate that MVP-Nav significantly outperforms existing depth-free methods, achieving state-of-the-art performance and validating that structured physical priors can effectively compensate for the absence of active depth sensors. Code could be accessed in https://github.com/BorisXwy/MVP-Nav.git**

## I. Introduction

**Zero-shot Object Goal Navigation** (ZSON) [22] is a fundamental capability for autonomous agents, requiring them to locate unseen objects in new environments based on natural language instructions. While conventional methods relied heavily on active sensors like LiDAR or RGB-D cameras to obtain precise metric information [38], recent research has pivoted towards **RGB-only navigation** [14, 3]. This shift is motivated not only by the desire to reduce hardware costs and enhance the accessibility of consumer-grade robots but also by the goal of challenging agents with higher-level visual reasoning. As noted by Chaplot et al. [6], biological organisms primarily "rely on passive vision and internal priors to perceive 3D structure and navigate"; this paradigm therefore compels robots to move without depth sensors, instead mimicking human-like perception patterns, inferring complex semantic meanings and latent geometric structures from monocular or panoramic visual cues combined with spatial commonsense.

Current research in RGB-only navigation primarily follows two paradigms. Semantic-driven methods [14] utilize Multimodal Large Language Models (MLLMs) for panoramic parsing or heuristic exploration. However, treating space as a collection of discrete semantic labels often leads to distorted topological relationships under perspective changes. End-to-end methods [3, 35, 12] learn spatial mappings through vision-action coupling. However, without explicit geometric modeling, they often fail to accurately perceive physical obstacles, resulting in navigation that is semantically reasonable yet physically unsafe. Collectively, these approaches expose a fundamental dilemma: existing solutions either decouple physical perception from high-level planning or sacrifice geometric constraints at the level of low-level control. As a result, current systems lack a unified representation that aligns discrete semantic reasoning with continuous physical boundaries. These limitations highlight a critical bottleneck in RGB-only navigation: the inability to recover explicit 3D scale from 2D sequences leads to **physical uncertainty** (i.e., the ambiguity of object distance and scale), and further induces **semantic-physical misalignment**, where high-level reasoning becomes disconnected from low-level physical constraints.

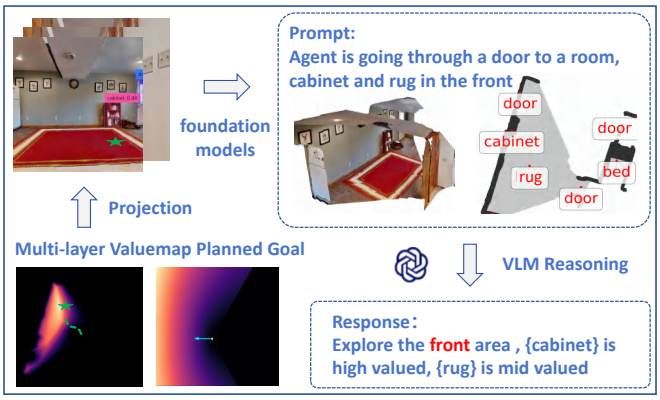



Fig. 1: The overview of our navigation system.

To address these challenges, we argue that the perception and control layers in object-goal navigation should align as closely as possible with the real physical world, while planning should leverage both historical exploration memory and real-time observations. Based on this principle, we propose **MVP-Nav (Multi-layer Value Map Planner Navigator)**. Our framework is motivated by the insight that physical occupancy in monocular vision can be reconstructed via geometric priors rather than explicit depth estimation. To mitigate **physical uncertainty**, we leverage 3D foundation models to project semantic instances from 2D observations into 3D space, representing them as Oriented Bounding Boxes (OBB) [10]. This semantic-to-physical re-projection transforms ambiguous depth cues into deterministic spatial representations and enables the construction of a Global Spatial Semantic List

(GSSL) to maintain the locations and scales of observed instances. Furthermore, to address the **semantic-physical misalignment**, we introduce the Multi-layer Value Map (MVM) mechanism. MVM acts as a unified integration layer that embeds MLLM-generated high-level semantic priorities and low-level physical constraints into a shared cost space. This design enables geometric path planning (e.g., via the Fast Marching Method) that is jointly governed by reconstructed physical occupancy and semantic guidance. By decoupling high-level reasoning from low-level execution, **MVP-Nav** ensures navigation that is both goal-directed and physically grounded.

The contributions of this work are summarized as follows:

- We propose a **physical-aware paradigm** for RGB-only ZSON, which reconstructs explicit 3D spatial occupancy from monocular observations by leveraging 3D foundation models and OBB-based fusion.
- We introduce the **Multi-layer Value Map (MVM)** architecture, providing a bridge to align discrete semantic reasoning with continuous physical constraints in a unified cost space.
- Extensive evaluations demonstrate that MVP-Nav achieves state-of-the-art performance among depth-free methods, proving that well-structured physical priors can effectively compensate for the absence of active depth sensors.

## II. Related works

### A. Conventional Object Goal Navigation

**Object Goal Navigation (ObjectNav)** requires an agent to locate and approach a specific object category in unseen environments. The development of this field is closely tied to the standardization of benchmarks, particularly the Habitat platform [27]. By providing photorealistic 3D datasets such as Matterport3D and Gibson [33], Habitat established the "Success weighted by Path Length" (SPL) [1] as the primary metric to evaluate both navigation effectiveness and path efficiency.

Historically, ObjectNav has been dominated by Geometry-driven Paradigms that prioritize explicit metric mapping and frontier exploration. Early successful agents primarily relied on active depth sensing (e.g., RGB-D) to build explicit 2D/3D occupancy representations for path planning [7]. A classic representative is Stubborn [21], which effectively leverages geometric baselines to prioritize collision-free frontier exploration, demonstrating that structured physical occupancy is fundamental to reliable navigation. As the field progressed, the focus shifted from pure geometric obstacle avoidance to cognitive process modeling [4]. This paradigm shift is deeply rooted in neuroscientific findings, which suggest that both humans and animals maintain internal representations of environmental geometry—such as grid cells and place cells—to support flexible spatial navigation [42]. Consistent with these theories, recent studies have shown that human-like navigation involves maintaining and dynamically updating fine-grained cognitive states [4]. This implies that while metric geometry provides a physical safety foundation, the integration of high-level environmental understanding is essential for efficient exploration. This evolution is further reflected in graph-based works like VoroNav [32], which utilizes Voronoi diagrams to abstract spatial topology while still leaning on the geometric scaffolding established by earlier depth-based methods.

### B. Recent RGB-only Navigation

The transition toward RGB-only Navigation represents a deliberate move toward simulating higher-level visual cognition, where agents must rely on semantic cues and spatial memory rather than direct distance measurements. Modern depth-free methods have evolved from early reactive policies into two main technical paradigms:

*Modular Reasoning based on Image/Pixel Space:* Early RGB-only attempts like ONN [37]and Target-driven RL [46] utilized visual scene priors or generic feature embeddings to guide exploration. To compensate for the lack of depth, modern works leverage Multi-modal Large Language Models (MLLMs) to perform reasoning directly on observed images or panoramic views [3, 14]. For instance, PixNav [3] bridges foundation models and navigation via pixel-guided goal specification, while ImagineNav [44] and PanoNav [14] use 'scene imagination' and panoramic parsing to predict semantic waypoints. Similarly, SG-Nav [38] prompts LLMs with 3D scene graphs to enhance reasoning. A notable recent work, Mobility VLA [34] , leverages long-context reasoning with topological graph priors to navigate in known environments. However, these methods either depend on environment-specific priors or perform inference in 2D semantic spaces, frequently suffering from a Semantic-Physical Gap. In contrast, MVP-Nav targets zero-shot exploration in entirely unseen environments by reconstructing a 3D substrate for every decision, ensuring the agent's 'mental plan' is consistently grounded in the latent 3D layout. However, because these methods perform inference in 2D image semantics or abstract graph spaces, they frequently suffer from a Semantic-Physical Gap. The agent's 'mental plan' often fails to ground in the environment's latent 3D layout, leading to semantically logical but physically unreachable goal points.

*End-to-End Learning Paradigm:* This paradigm explores mapping raw RGB pixels directly to control signals, with early foundational works such as SAVN [31] focusing on self-adaptive navigation. Modern attempts like OmniNav [35] and the navigation foundation model NavFoM [43] seek to achieve generalizable navigation through large-scale pre-training across diverse tasks and embodiments. While these methods provide efficient, low-latency policies, they typically treat spatial relations as implicit latent features. Without an explicit persistent 3D substrate to ensure **Embodied Spatial Consistency**, these end-to-end models struggle to maintain stable environmental awareness over long horizons, manifesting the **Physical Uncertainty** and **Semantic-Physical Misalignment** that our work aims to address.


Fig. 2: Overview of the **MVP-Nav** framework. Our system employs a recursive architecture that first transforms monocular RGB sequences into a 3D Global Spatial Semantic List (GSSL) using foundation models. A Vision-Language Model (VLM) then reasons over this list to assign semantic scores and determine the navigation mode. These insights are integrated into a Multi-layer Value Map (MVM) to identify an optimal midterm goal $g_{mid}$ within a unified cost space. Finally, a low-level execution loop performs $A^*$ and FMM planning, ensuring physical safety via real-time semantic floor re-projection.

Unlike these paradigms, **MVP-Nav** reconciles high-level reasoning with physical boundaries by grounding VLM decisions within a recovered 3D volume via the Multi-layer Value Map. This transition from image-space imagination to physically-grounded planning ensures both goal-directed efficiency and environmental safety, fundamentally overcoming the bottleneck of depth-free navigation.

## III. Methodology

### A. Problem Statement

Following the standard experimental protocols of Object Goal Navigation (ON) benchmarks [27], we consider an embodied agent operating within a large-scale, *stationary* indoor environment $\mathcal{E}$. The agent is equipped with a reliable onboard localization system that provides its global pose estimate $p_t = (x_t, y_t, \theta_t)$ at each time step $t$. Despite the availability of pose information, the environmental geometry, semantic distribution, and the exact coordinates of the target remain entirely unknown at the start of the mission.

The agent's perception is strictly constrained to monocular vision, where the observation tuple at time $t$ is $O_t = \{I_t, p_t\}$. We adhere to a rigorous *RGB-only constraint*, meaning the system does not rely on any active depth sensors or LiDAR. Consequently, all spatial information must be inferred purely from the image sequence $\mathcal{I}$. The navigation goal is specified by a goal image $I_{goal}$ or a language instruction. The task is deemed successful if the agent executes a `STOP` command at a pose $p_t$ such that $\|p_t - p^*\|_2 < d$, where $p^*$ represents the target position and $d$ is a predefined distance threshold.

### B. Pipeline Overview

In this subsection, we will introduce the overall process of the entire episode, including the start and end of tasks, as well as the multi round navigation mechanism in between. The navigation mechanism as the core algorithm will be explained in detail in the following content, while the start and end of tasks as non core content will only be introduced here.

*1) Episode Start:* The target image is parsed by a VLM into a combination of a main-target and several sub-targets, which will be utilized in subsequent navigation loop. At the start of the episode, the robot first performs a 360-degree in-place rotation to obtain the initial observation before proceeding to the actual navigation.

*2) MVP-Nav Loop:* The **MVP-Nav** framework adopts a decoupled, recursive architecture designed to bridge the gap between high-level semantic reasoning and low-level physical constraints. As illustrated in Fig. 2 and Algorithm 1, the system alternates between high-level decision stages and low-level execution loops through four logical components:

**Physical Perception**. This module transforms the monocular image sequence $\mathcal{I}$ into a structured Global Spatial Semantic List (GSSL). We utilize 3D foundation models as the geometric backbone for end-to-end 3D reconstruction, recovering pseudo-depth and physical scale. 2D instances from open-

vocabulary models are projected into 3D Oriented Bounding Boxes (OBB) to populate the GSSL.

**VLM-based High-level Reasoning**. The Vision-Language Model (VLM) acts as the system's cognitive core. Leveraging common-sense priors, the VLM processes GSSL entities and the navigation goal to evaluate spatial-semantic relevance. It assigns heuristic weights to entities and determines the navigation strategy (e.g., *Explore*, *Find*, or *Judge*).

**Multi-layer Value Map (MVM) Planning**. This component unifies high-level VLM reasoning with low-level geometric constraints into a shared cost space. By projecting 3D OBBs and VLM-assigned weights onto a 2D grid, the system generates a multi-layer value map integrating semantic attraction, exploration guidance, and traversability. A path search on this map establishes a stable midterm goal ($g_{mid}$).

**Low-level Execution Loop**. Once $g_{mid}$ is established, the agent enters a high-frequency control loop, generating short-term goals ($g_{shorterm}$, simplified as $g_{st}$) via $A^*$ [11] and the Fast Marching Method (FMM) [28]. To ensure safety under the *RGB-only constraint*, we implement a Safety Verification step by back-projecting semantic floor masks to validate real-time traversability. The loop continues until $g_{mid}$ is reached, triggering the next reasoning cycle.

*3) Episode Finish:* As for the termination of the whole episode, if the reasoning stage give the mode as Judge, which means the agent seems find the goal. The robot performs a 360° panoramic scan. We utilize LightGlue to match the goal image with the most relevant historical views. Final task completion is only triggered after a Vision-Language Model (VLM) confirms the target's presence in the immediate vicinity. If a sub goal or main goal is misdetected, the system will give it a low score to stop agent from being attracted again.

**Algorithm 1 MVP-Nav**: Recursive Navigation Loop

**Input:** $I_{goal}, \mathbf{p}_0$; **Init:** $\mathcal{I}_{curr} \leftarrow$ initial 360° scan;
**while** mission not complete **do**
  Update GSSL via Physical Perception using $\mathcal{I}_{curr}$;
  Select NavMode and Direction through GSSL and $I_{goal}$;
  Synthesize Value Map, determine midterm goal $\mathbf{g}^*$;
  $\mathcal{I}_{next} \leftarrow \emptyset$;
  **while** not reached $\mathbf{g}_{mid}$ **do**
    Generate short-term goal $\mathbf{g}_{st}$ toward $\mathbf{g}_{mid}$;
    Execute Obstacle Avoidance using real-time observations and $\mathbf{g}_{st}$;
    Record $I_t$ into $\mathcal{I}_{next}$ and update $\mathbf{p}_t$;
  **end while**
  $\mathcal{I}_{curr} \leftarrow \mathcal{I}_{next}$;
**end while**

### C. Physical Perception for 3D Reconstruction and GSSL

The Physical Perception module serves as the geometric and semantic foundation of **MVP-Nav**. As shown in Fig. 4 (a), this module aims to lift fleeting monocular RGB observations $I_t$ into a persistent 3D spatial memory through four stages. It enables end-to-end 3D reconstruction from uncalibrated sequences by regressing globally consistent point clouds, thereby recovering pseudo-depth and physical scale from monocular observations.

For each navigation stage, the agent utilizes the visual geometry foundation model VGGT [30] to process the image sequence $\mathcal{I}$, regressing the pseudo-depth map $\hat{z}$ and camera parameters $\hat{\mathbf{K}}$. All pixels $\mathbf{q} = [u, v]^\top$ from the original image $I_t$ are back-projected to construct a comprehensive local 3D point cloud $\mathcal{P}_{raw}$:

$$\mathbf{p} = \hat{z} \cdot \hat{\mathbf{K}}^{-1}[u, v, 1]^\top. \tag{1}$$

In each stage, observation history collected in last stage are used as the source image of Physical Perception. To balance perceptual coverage and computational efficiency, we constrain the number of input images to approximately 30 per stage, if agent moves over 30 steps, this stage is stopped early. Empirically, exceeding 50 images significantly increases the risk of Out-of-Memory (OOM) errors during the global point cloud reconstruction phase.

Simultaneously, Grounded-SAM [26, 20, 16] generates three categories of semantic masks by inference 3 times separately , namely target objects ($M_{target}$), general objects ($M_{obj}$), and the floor ($M_{floor}$), which collectively guide the branching of point cloud data. Specifically, the raw point cloud $\mathcal{P}_{raw}$ is projected into the Bird's-Eye View (BEV) space, where non-floor occupancy is extracted and processed via morphological inflation to generate a local occupancy map for $A^*$ and FMM planning (Sec. III-F). Points corresponding to $M_{floor}$ are isolated and passed to the low-level execution loop as a benchmark for real-time semantic re-projection verification. To prevent semantic cross-contamination during spatial fusion, the system explicitly decouples point clusters covered by $M_{target}$ and $M_{obj}$ to fit separate Oriented Bounding Boxes (OBB). This isolation ensures that navigationally critical target entities remain distinct in the GSSL, as Fig. 3, and are not erroneously merged with nearby general obstacles.

For GSSL entity generation and verification, the system implements a robust construction pipeline in the local space. The final entity label $L_{entity}$ is determined by a majority voting mechanism based on the mode of all associated detections. To ensure temporal consistency and filter out sensor noise, an observation frequency constraint is enforced, requiring each entity to be associated across at least three independent frames. Furthermore, OBBs serve as the primary geometric entries, effectively optimizing computational efficiency while maintaining high physical occupancy accuracy.

Building upon these locally consistent entities, the system must then integrate them into a globally metric-consistent framework. To this end, scale ambiguity is resolved by correlating the predicted trajectory $\Gamma_{vggt}$ and the metric trajectory $\Gamma_{loc}$. The similarity transform $\mathcal{T}(\mathbf{x}) = \sigma\mathbf{R}\mathbf{x} + \mathbf{t}$ is derived by

minimizing the residual:

$$\min_{\sigma,\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} \|\mathbf{p}_{loc,i} - (\sigma \mathbf{R}\mathbf{p}_{vggt,i} + \mathbf{t})\|^2. \quad (2)$$

Once the scales are unified, the GSSL can be dynamically updated via 3D Intersection-over-Union (IoU). In this stage, if semantics are consistent and the spatial overlap exceeds a predefined threshold, a weighted fusion is triggered to refine the entity properties. The preliminary category decoupling mentioned above ensures that target entities remain semantically salient throughout this fusion process, ultimately providing a high-fidelity memory for VLM-based reasoning.


```
[   {
    "id": "node_11",
    "caption": "doorframe",
    "position": "directly behind, 1.18m away",
    "explore": false,
    "exploration_score": 0.2,
    "cls_type": "node"
    },
    {
    "id": "node_12",
    "caption": "cabinet",
    "position": "to the back-left, 2.99m away",
    "explore": false,
    "exploration_score": 0.4,
    "cls_type": "node"
    },
    {
    "id": "sub_8",
    "caption": "shelf",
    "position": "directly behind, 3.04m away",
    "explore": false,
    "exploration_score": 0.8,
    "cls_type": "sub"
    },
    ...]
```


Fig. 3: An example of a part of GSSL.

### D. VLM-based High-level Reasoning

The VLM reasoning module serves as the cognitive core, taking the GSSL and navigation goal $\mathcal{G}$ as inputs to output semantic importance scores $\alpha_i$ for each entity and a navigation mode to guide spatial planning. To evaluate the exploration value, the Vision-Language Model (VLM, e.g., GPT-4o) leverages common-sense priors to assess the relevance of environmental entities to the goal. In order to optimize computational efficiency, the VLM performs scoring exclusively on newly detected entities within the GSSL, denoted as having an explored status of False. For each novel entity $\omega_i$, the VLM assigns an exploration value score $\alpha_i \in [0, 1]$ based on its likelihood of being the target or a significant semantic cue, which is formally defined as:

$$\alpha_i = \text{VLM}(\omega_i, \mathcal{G} \mid \text{Common-sense Priors}). \quad (3)$$

These scores are stored in the GSSL and serve as weights for evaluating spatial attraction during the cost-map generation stage.

Beyond scoring, the VLM acts as a high-level arbitrator that dynamically determines the navigation mode based on the distribution of exploration values. When no high-value entities are identified in the GSSL, the system operates in an explore mode, where the VLM provides coarse directional suggestions such as left-front, front, or right-front. The system then enhances the exploration rewards of the corresponding regions in the cost map to guide the agent toward unmapped areas. Once the target or a strongly correlated cue, such as a pillow when searching for a bed, is detected, the system transitions to a find mode and plans a trajectory directly toward the spatial center of the entity. Finally, when the agent approaches the main target, it enters a judge mode to collect multi-view observations of the object. These visual features are compared with the goal image $I_{goal}$ by the VLM to verify whether the current entity fulfills the task requirements before terminating the search.

### E. Multi-layer Value Map Planning

As shown in Fig. 4 (b), the MVP module integrates the GSSL, VLM-derived scores $\alpha_i$, and the local occupancy map to synthesize a cost space. By synthesizing semantic, directional, and geometric constraints, the system generates a total value map $\Phi_{total}$ to determine the optimal midterm goal $g_{mid}$. Specifically, the planning module constructs three independent value layers in the 2D grid space to represent different planning constraints:

- Semantic Value Map ($\Phi_{sem}$): This layer is generated by aggregating the spatial contributions of GSSL entities. To avoid over-saturation, the sum is clipped at 1.0:

$$\Phi_{sem}(\mathbf{p}) = \min\left(\sum_{i \in \text{GSSL}} \alpha_i \eta_i \cdot e^{-\frac{\|\mathbf{p}-\mathbf{c}_i\|^2}{2\sigma_i^2}}, 1.0\right), \quad (4)$$

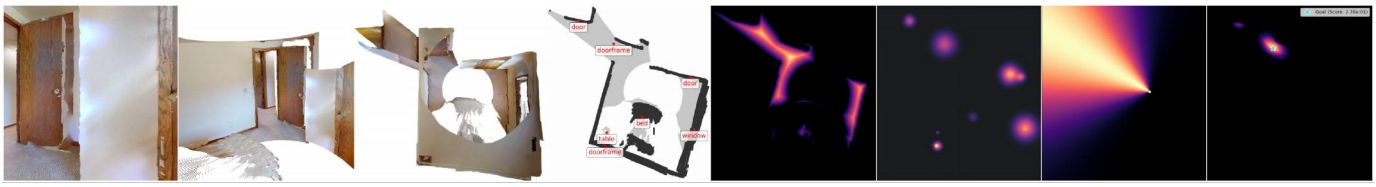
  where $\mathbf{c}_i$ denotes the projected center, $\sigma_i$ is the standard deviation derived from the entity's physical extent, and $\eta_i$ represents whether the object $\mathbf{o}_i$ has been explored or not: $\eta_i$ = 1 for newly-detected, and $\eta_i$ = 0.5 for already detected.
- Directional Value Map ($\Phi_{dir}$): This layer biases movement toward the VLM-suggested orientation. Letting $\Delta\theta = \theta_{\mathbf{p}} - \theta_{target}$ represent the angular deviation, the map is formulated as:

$$\Phi_{dir}(\mathbf{p}) = \exp\left(-\frac{\Delta\theta^2}{2\sigma_\theta^2}\right) \cdot \mathbb{I}\left(|\Delta\theta| \leq \frac{\pi}{2}\right), \quad (5)$$

  where $\sigma_\theta$ is the directional variance controlling the angular spread of semantic guidance, and $\mathbb{I}(\cdot)$ is an indicator function that restricts the exploration gain to the forward 180° semi-circular sector.
- Traversability Value Map ($\Phi_{trav}$): This layer is derived from the geometric occupancy map. We compute the distance $d_{min}(\mathbf{p})$ to the nearest obstacle for each cell to ensure safe navigation:

$$\Phi_{trav}(\mathbf{p}) = 1 - \exp(-k \cdot d_{min}(\mathbf{p})). \quad (6)$$

  where $k$ is the distance penalty weight: $k = 1.0$ for obstacles, and $k = 0.5$ for unknown regions




(a) (b)

Fig. 4: Construction of the Multi-layer Value Map (MVM). (a) The framework reconstructs 3D oriented bounding boxes (OBBs) from monocular RGB sequences to establish physical occupancy and spatial semantics. (b) These representations are integrated into the MVM, where semantic priorities and geometric constraints are fused into a unified cost space to determine the optimal navigation goal.

The final total value map $\Phi_{total}$ is synthesized through an element-wise product of these constituent layers, such that $\Phi_{total} = \Phi_{sem} \odot \Phi_{dir} \odot \Phi_{trav}$. The optimal midterm goal $g_{mid}$ is then determined by identifying the global maximum within this fused space, calculated as $g_{mid} = \arg\max_{\mathbf{p}} \Phi_{total}(\mathbf{p})$. In this framework, $g_{mid}$ serves as the fixed reference point for the current navigation stage, guiding the agent's local execution until a new high-level decision or an environmental update triggers a re-evaluation.

Notably, this multiplicative fusion mechanism distinguishes MVM from traditional Artificial Potential Fields (APF)[18]. While APF typically generates control gradients by summing attractive and repulsive forces at the execution level, MVM operates at the planning level by performing element-wise multiplication. This ensures that the selected $g_{mid}$ strictly satisfies all semantic and physical constraints simultaneously—any area marked as non-traversable in $\Phi_{trav}$ (value of 0) will result in a zero total value, effectively eliminating candidates that are semantically attractive but physically unreachable.

### F. Low-level Execution Loop

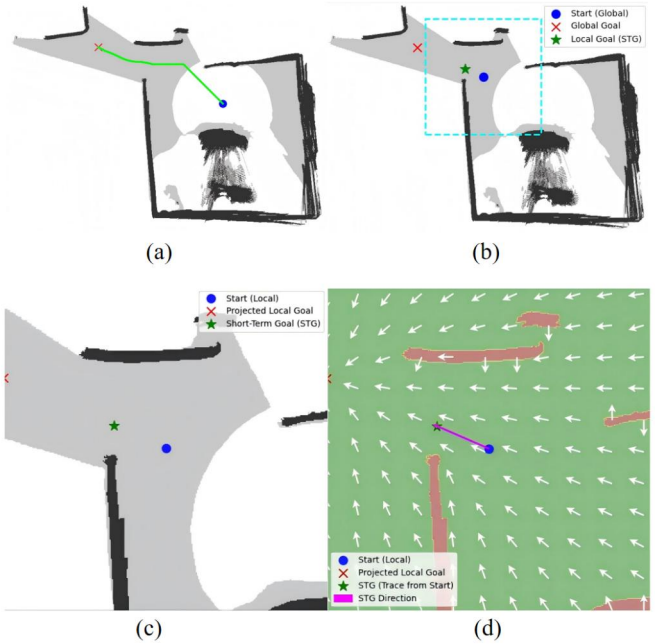
The execution layer translates $g_{mid}$ into motor commands by combining map-based pathfinding with reactive semantic verification. The system performs an $A^*$ search on the local occupancy map to establish a topological trajectory. To ensure high-frequency reactivity, as Fig. 5, a sliding-window Fast Marching Method (FMM) identifies the intersection of the $A^*$ path and the window boundary as a projection target, generating a refined local goal $g_{st}$. While A* identifies the optimal discrete waypoint sequence, FMM solves the Eikonal equation on the MVM to generate a continuous potential field. This field acts as a safety-aware tracker, guiding the robot with smooth velocity commands while maintaining distance from identified hazards.

To compensate for map discretization, a safety layer uses the floor mask $M_{floor}$ for reactive avoidance. Before moving `FORWARD`, $g_{st}$ is re-projected onto the image plane:

$$\mathbf{q}_{local} = \hat{\mathbf{K}} \left( \mathbf{R} \cdot g_{st} + \mathbf{t} \right). \tag{7}$$

If $\mathbf{q}_{local} \notin M_{floor}$, the agent halts and rotates until the goal aligns with the traversable region. This dual-layered strategy maintains global consistency alongside real-time constraints.

Furthermore, **MVP-Nav** employs a parallelized pipeline to eliminate transition latency. When $\|p_t - g_{mid}\| < \epsilon$, high-level reasoning for the next stage is triggered during active low-level control. Overlapping VLM inference with physical movement enables a fluid transition to $g_{mid,next}$ without requiring the agent to stop.




(a) (b) (c) (d)

Fig. 5: Visualization of the local goal $g_{st}$ generation. (a) The $A^*$ path from start position to the midterm goal. (b) The silding window of current position. (c) The intersection of sliding window and the $A^*$ path, which is the projected goal. (d) The System generates a short-term goal (STG) in the local occupancy map with FMM Planner.

## IV. Experiments

### A. Experimental Setup

*Benchmarks:* We evaluate the performance of **MVP-Nav** across three widely-adopted indoor object-goal navigation benchmarks, namely HM3D [23], MP3D [5], and RoboTHOR [8]. Specifically, HM3D consists of 20 high-fidelity indoor scene reconstructions encompassing 2K validation episodes across 6 target object categories, while MP3D provides 11 real-world indoor environments with 1.8K validation episodes covering 20 categories. Additionally, we utilize RoboTHOR, which includes 1.8K validation episodes across 15 indoor environments involving 12 goal categories. In each episode, the goal image is defined by episode file in the dataset.

*Evaluation Metrics:* To quantify navigation performance, we employ two standard metrics: Success Rate (SR) and Success weighted by Path Length (SPL). Specifically, SR represents the ratio of episodes where the agent successfully

| Method | Navigation Setting | | | MP3D | | HM3D [23] | | RoboTHOR | |
|---|---|---|---|---|---|---|---|---|---|
| | Perception | Execution | Training-free | SR (%) | SPL (%) | SR (%) | SPL (%) | SR (%) | SPL (%) |
| SemExp [7] | RGB-D | RGB-D | × | 36.0 | 14.4 | 37.9 | 18.8 | — | — |
| PONI [24] | RGB-D | RGB-D | × | 31.8 | 12.1 | — | — | — | — |
| Habitat-Web [25] | RGB-D | RGB-D | × | — | — | 41.5 | 16.0 | — | — |
| OVRL [36] | RGB-D | RGB-D | × | — | — | 62.0 | 26.8 | — | — |
| ESC [45] | RGB-D | RGB-D | ✓ | 28.7 | 14.2 | 39.2 | 22.3 | 38.1 | 22.2 |
| VoroNav [32] | RGB-D | RGB-D | ✓ | — | — | 42.0 | 26.0 | — | — |
| L3MVN [41] | RGB-D | RGB-D | ✓ | 34.9 | 14.5 | 48.7 | 23.0 | 41.2 | 22.5 |
| OpenFMNav [17] | RGB-D | RGB-D | ✓ | 37.2 | 15.7 | 52.5 | 24.1 | 44.1 | 23.3 |
| VLFM [40] | RGB-D | RGB-D | ✓ | 36.2 | 15.9 | 52.4 | 30.3 | 42.3 | 23.0 |
| SG-Nav [38] | RGB-D | RGB-D | ✓ | 40.2 | 16.0 | 54.0 | 24.9 | 47.5 | 24.0 |
| Unigoal [39] | RGB-D | RGB-D | ✓ | 41.6 | 16.4 | 54.5 | 25.1 | 48.0 | 24.2 |
| ZSON [22] | RGB | RGB | × | 15.3 | 4.8 | 25.5 | 12.6 | — | — |
| PixNav [3] | RGB | RGB | × | — | — | 37.9 | 20.5 | — | — |
| PanoNav [14] | Pano-RGB | Pano-RGB | ✓ | — | — | 43.5 | 23.7 | — | — |
| ImagineNav [44] | RGB | RGB-D | ✓ | — | — | 53.0 | 23.8 | — | — |
| **MVP-Nav (Ours)** | **RGB** | **RGB** | ✓ | **50.4** | **18.1** | **65.4** | **27.9** | **57.5** | **26.8** |

TABLE I: Comprehensive comparison with state-of-the-art methods on Object-Goal Navigation benchmarks. We report Success Rate (SR) and SPL (%) across three major datasets. Missing metrics are indicated by a dash (—).


Fig. 6: Examples of successful navigation episodes in the HM3D dataset. The trajectories illustrate how **MVP-Nav** effectively handles diverse indoor layouts.

reaches the target object within a predefined distance threshold, while SPL evaluates navigation efficiency by normalizing the success rate with the ratio of the shortest path length to the actual distance traversed by the agent.

*Implementation Details:* Experiments are conducted in the Habitat simulator with $640 \times 480$ RGB observations. The agent executes discrete actions (0.25m per step, 30° rotations). We use VGGT for geometric perception and GPT-4o-mini for high-level reasoning. To assess deployment feasibility, all benchmarks and computational analyses are performed on a workstation with an NVIDIA RTX 6000 Ada GPU and an Intel Xeon Platinum 8352V CPU.

### B. Results

As summarized in Table I, **MVP-Nav** demonstrates strong competitiveness across all benchmarks. Compared to the primary RGB-only baseline PanoNav, **MVP-Nav** exhibits an overwhelming performance gain; specifically, on the HM3D dataset, we significantly improve the Success Rate (SR) from 43.5% to 65.4% and the SPL from 23.7% to 27.9%. These results demonstrate that our multi-view geometric perception framework constructs spatial representations far more effectively than traditional monocular schemes.

Notably, despite relying solely on RGB input, **MVP-Nav** achieves performance levels on par with, or even superior to, several state-of-the-art RGB-D systems equipped with physical depth sensors. For instance, our SR outperforms SG-Nav [38] (54.0%) and UniGoal [39] (54.5%), both of which utilize offline-trained scene graphs. This suggests that semantic planning based on physical constraints enables the system to reach a capability level close to that of active depth sensing. Fig. 6 shows two successful episodes in the HM3D dataset.

### C. Ablation Study and Analysis

*Ablation Study:* As summarized in Table II, each core component in **MVP-Nav** is critical for balancing geometric fidelity and navigation intelligence.

**Ablation Analysis.** The failure of the "Full-Sequence Reconstruction" underscores our recursive framework's necessity. Maintaining global maps from long sequences causes severe cumulative drift, geometric warping, and Out-of-Memory (OOM) failures. In contrast, our recursive strategy reconstructs at smaller spatial scales, ensuring high-fidelity, drift-free local

maps while preserving essential context via the Spatial Semantic List.

The semantic re-projection module is vital for operational safety. Removing it (w/o Sem. Re-proj.) marginally increases SPL ($27.9\% \rightarrow 30.5\%$) but significantly reduces Success Rate (SR). Without re-projection, agents adopt riskier, direct trajectories without verifying floor traversability; frequent collisions justify this efficiency trade-off for robust physical safety.

Furthermore, exploration mechanisms are critical for zero-shot navigation. Disabling exploration memory (w/o explore memory) causes SR to plummet to $38.2\%$ as agents become trapped in repetitive search. Similarly, removing LLM-based rating (w/o LLM rating) drops SR to $45.7\%$, demonstrating that without LLM-driven common-sense reasoning to prioritize candidate frontiers (e.g., identifying hallways as paths to kitchens), exploration becomes suboptimal, confirming the value of semantic anchoring.

| Module | SR (%) ↑ | SPL (%) ↑ |
|---|---|---|
| Full-Sequence Recon. | OOM (Out of Memory) | |
| w/o Sem. Re-proj. | 53.2 | **30.5** |
| w/o explore memory | 38.2 | 13.4 |
| w/o LLM rating | 45.7 | 20.1 |
| **full MVP-Nav** | **65.4** | 27.9 |

TABLE II: Ablation Results of **MVP-Nav** on the HM3D Dataset.

*Component Selection Comparation:* We screened various foundation models to determine the optimal configuration for perception and reasoning, as detailed in Table III. Regarding 3D perception, while *Depth-Anything V3* achieves a minimum latency of 2.56 s, its insufficient accuracy fails to provide reliable geometric constraints, leading to a 7.6% drop in SR. Conversely, although *Map-Anything* can generate dense maps, its VRAM consumption (28.78 GB) is prohibitive for real-time applications. The VGGT scheme provides the best balance, maintaining the highest SR with moderate resource usage. For the reasoning backend, GPT-4o-mini stands out with the lowest end-to-end latency (5.59 s) and superior logical robustness in following spatial semantic instructions compared to local alternatives like *Llama3.2-vision*.

| Module | SR (%) | SPL (%) | Latency | VRAM |
|---|---|---|---|---|
| *Physical Perception* | | | | |
| Depth-Anything V3 [19] | 57.8 | 21.5 | **2.56** | 18.61 |
| Map-Anything [15] | 62.4 | 23.7 | 14.83 | 28.78 |
| **VGGT** [30] | **65.4** | **27.9** | 5.31 | **15.72** |
| *VLM Reasoning* | | | | |
| Llama3.2-vision (Local) [9] | 58.3 | 22.4 | **4.33** | – |
| Gemini-3-flash (API) [29] | 63.8 | 26.5 | 8.64 | – |
| **GPT-4o-mini (API)** [13] | **65.4** | **27.9** | 5.59 | – |

TABLE III: Performance analysis of each modules in **MVP-Nav**.

*Computational Efficiency and Latency Analysis:* We evaluate the computational overhead by measuring the end-to-end latency of a single navigation loop. As reported in Table IV, the Physical Perception and VLM Reasoning module serves as the primary bottleneck, requiring 10.90 s per stage.

To prevent this latency from causing motion interruptions, **MVP-Nav** utilizes a parallel pipeline mechanism that overlaps high-level reasoning with physical execution. Specifically, we initiate the planning for the next stage when the agent is within a Euclidean distance threshold of $\epsilon = 3\,\text{m}$ from the current midterm goal.

In each stage, the average step count remaining after reaching the distance $d = \epsilon$ is 10.3. Given that the robot moves $0.25\,\text{m}$ per step and the inference time is approximately $1.67\,\text{s}$, the buffer time of $10.3 \times 1.67\,\text{s} = 17.20\,\text{s}$ theoretically eliminates latency. In practice, however, the number of steps varies per stage, resulting in an actual average delay of $1.6\,\text{s}$. This is negligible compared to the average stage duration of $50.31\,\text{s}$. This minimal overhead demonstrates that our hierarchical decoupling successfully reconciles the high computational demands of MLLMs with the requirements for fluid indoor navigation.

| Module | Device | Latency (s) | Ratio (%) |
|---|---|---|---|
| Physical Perception | GPU | 5.31 | 40.26% |
| MLLM Semantic Reasoning | API-Call | 5.59 | 42.38% |
| Spatial Semantic List Update | CPU | 2.12 | 16.18% |
| MVM & Path Planning | CPU | 0.17 | 1.29% |
| **Total** | - | **13.19** | **100%** |
| Low-level Execution | costs 1.67s per step | | |

TABLE IV: Latency of different modules of **MVP-Nav**.

### D. Real-World Experiments

To bridge the gap between simulation and reality, we deploy **MVP-Nav** on a physical robotic platform to evaluate its navigational robustness in unstructured indoor environments.

*Hardware and Sensor Configuration:* We utilize the **Agibot G1**, a wheeled robot, for our real-world deployment. The hardware and sensor configurations are detailed as follows:

- **Vision System**: Navigation is performed using only the RGB stream from the single head-mounted camera. To balance the field of view (FOV) between low-profile floor obstacles and eye-level semantic instances, the robot's head is fixed at a pitch angle of $-20^\circ$.
- **Motion Tracking**: The G1 is equipped with high-precision wheel encoders and a 9-axis IMU. The onboard odometry system provides real-time pose feedback at a frequency of 1 kHz, which is critical for maintaining trajectory consistency during our recursive map updates.
- **Computing Infrastructure**: All heavy computations are offloaded to a dedicated server via a high-speed wireless link. The server is equipped with dual NVIDIA RTX 2080 Ti GPUs.

*Experimental Environment and Setup:* The real-world evaluation is conducted in a university library corridor, a demanding environment characterized by long-range vistas, repetitive visual patterns, and varying lighting conditions. Specifically, we define two navigation tasks: (1) locating a **fire extinguisher box** situated in the left corridor after exiting the starting room, and (2) reaching a **large green plant** positioned along the wall of the right corridor. Each task was

Fig. 7: Real-world experimental results. We evaluated **MVP-Nav** on an Agibot G1 [2] wheeled robot in a library corridor. The results (right) show that the system can successfully navigate to target objects over distances exceeding 15 meters, proving that our framework effectively translates from simulation to physical hardware.

Fig. 8: Real-world deployment on a wheeled robot platform. We present sequential frames captured from a follow-cam perspective to visualize the robot's physical execution.

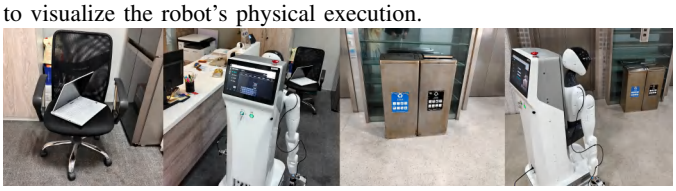


Fig. 9: Real-world deployment in another scenario.

| Target | test times | success | Ratio (%) |
|---|---|---|---|
| Fire extinguisher box | 20 | 7 | 35.0 |
| Large green plant | 20 | 11 | 55.0 |
| **Total** | 40 | 18 | **45.0** |
| Computer on the chair | 20 | 12 | 60.0 |
| Trash bin | 20 | 7 | 35.0 |
| **Total** | 40 | 19 | **47.5** |

TABLE V: Test results in two different real scenarios.

executed for 20 trials. These paths exceed 15 meters in length, with the left corridor stretching approximately 60 meters. To ensure stability, the robot's height was set to 130 cm. Another experiment scenario is an office with its many complex objects and its small range outdoor environment.In this scenario, the goal is a **computer on the chair** and and **trash bin** in front of the elevator

*Results and Observations:* Despite the total absence of active depth sensors, **MVP-Nav** works successfully in the physical world, achieving an overall success rate of $45.0\%$ $(18/40)$. In the right corridor task, the robot reached the green plant with a $55.0\%$ success rate $(11/20)$. Performance in the left corridor was lower, with 7 successful trials out of 20 $(35.0\%)$. Analysis suggests that the primary failure mode in the 60-meter left corridor is the inherent difficulty of **long-range monocular depth estimation**, which occasionally leads to point cloud drifting and map distortion at extreme distances.

It is worth noting that while **MVP-Nav** was originally optimized for indoor benchmarks, these experiments were conducted in a challenging **semi-outdoor** setting due to library environment constraints. The fact that the system maintains a reasonable success rate under such conditions—where depth ambiguity is maximized—underscores its robustness. Our observations highlight two key strengths:

- **Drift-Resistant Execution**: By leveraging the recursive stage-wise update mechanism, the cumulative drift from monocular vision input is effectively reduced. The robot maintains precise alignment between its $A^*$ path and the physical corridor boundaries.
- **Semantic Safety under Monocular Failure**: In the presence of glass partitions and textureless walls where monocular depth estimation frequently fails, **MVP-Nav** prevents collisions by identifying the semantic floor mask. The re-projection module successfully triggers proactive recovery whenever the projected goal falls outside the traversable floor region.
- **Semantic Safety under Monocular Failure**: In the presence of glass partitions and textureless walls where monocular depth estimation frequently fails, **MVP-Nav** prevents collisions by identifying the semantic floor mask. The re-projection module successfully triggers proactive recovery whenever the projected goal falls outside the traversable floor region.

## V. Conclusion

In this paper, we presented **MVP-Nav**, a training-free and RGB-only navigation framework that bridges the gap between high-level semantic reasoning and low-level physical occupancy. To address the challenge of achieving safe and consistent navigation without active depth sensing, we developed a 3D-driven OBB generation and computation algorithm that extracts structured geometric priors directly from 3D foundation models. These reconstructed OBBs are unified with MLLM-based semantic reasoning into a Multi-layer Value Map (MVM), enabling a tri-objective goal selection policy that respects both semantic relevance and physical constraints. Extensive experiments across multiple benchmarks and real-world deployment on a wheeled robot platform demonstrate that **MVP-Nav** generalizes robustly across diverse environments and instructions. We also identify several technical insights for future improvement, including fine-tuning 3D foundation models with navigation-specific priors to further enhance reconstruction fidelity, and designing hybrid memory structures to balance implicit neural features with explicit and explainable geometric outputs. Furthermore, exploring a tighter coupling between physical features and the execution layer remains a key direction to further minimize latency and enhance the agility of sensor-minimalist embodied systems.

## References


[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents, 2018. URL https://arxiv.org/abs/1807.06757.

[2] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[3] Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5228–5234. IEEE, 2024.

[4] Yihan Cao, Jiazhao Zhang, Zhinan Yu, Shuzhen Liu, Zheng Qin, Qin Zou, Bo Du, and Kai Xu. Cognav: Cognitive process modeling for object goal navigation with llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9550–9560, 2025.

[5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[6] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations (ICLR)*, 2020.

[7] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33: 4247–4258, 2020.

[8] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3164–3174, 2020.

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[10] Stefan Aric Gottschalk. *Collision queries using oriented bounding boxes*. The University of North Carolina at Chapel Hill, 2000.

[11] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[12] Junjun Hu, Jintao Chen, Haochen Bai, Minghua Luo, Shichao Xie, Ziyi Chen, Fei Liu, Zedong Chu, Xinda Xue, Botao Ren, et al. Astranav-world: World model for foresight control and consistency. *arXiv preprint arXiv:2512.21714*, 2025.

[13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[14] Qunchao Jin, Yilin Wu, and Changhao Chen. Panonav: Mapless zero-shot object navigation with panoramic scene parsing and dynamic memory, 2025. URL https://arxiv.org/abs/2511.06840.

[15] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*. IEEE, 2026.

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[17] Yuxuan Kuang, Hai Lin, and Meng Jiang. Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models. *arXiv preprint arXiv:2402.10670*, 2024.

[18] Min Cheol Lee and Min Gyu Park. Artificial potential field based path planning for mobile robots using a virtual obstacle concept. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, volume 2, pages 735–740 vol.2, 2003. doi: 10.1109/AIM.2003.1225434.

[19] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[20] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv:2303.05499*, 2023.

[21] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A strong baseline for indoor object navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3287–3293, 2022. doi: 10.1109/IROS47612.2022.9981646.

[22] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:

32340–32352, 2022.

[23] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv 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*, 2021. URL https://arxiv.org/abs/2109.08238.

[24] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning, 2022. URL https://arxiv.org/abs/2201.10029.

[25] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *CVPR*, 2022.

[26] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. URL https://arxiv.org/abs/2401.14159.

[27] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *ICCV*, 2019.

[28] James A Sethian. A fast marching level set method for monotonically advancing fronts. *proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[29] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[30] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

[31] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning, 2019. URL https://arxiv.org/abs/1812.00971.

[32] Pengying Wu, Yao Mu, Bingxian Wu, Yi Hou, Ji Ma, Shanghang Zhang, and Chang Liu. Voronav: Voronoi-based zero-shot object navigation with large language model. *arXiv preprint arXiv:2401.02695*, 2024.

[33] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[34] Zhuo Xu, Hao-Tien Lewis Chiang, Zipeng Fu, Mithun George Jacob, Tingnan Zhang, Tsang-Wei Edward Lee, Wenhao Yu, Connor Schenck, David Rendleman, Dhruv Shah, Fei Xia, Jasmine Hsu, Jonathan Hoech, Pete Florence, Sean Kirmani, Sumeet Singh, Vikas Sindhwani, Carolina Parada, Chelsea Finn, Peng Xu, Sergey Levine, and Jie Tan. Mobility vla: Multimodal instruction navigation with long-context vlms and topological graphs. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3866–3887. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/xu25b.html.

[35] Xinda Xue, Junjun Hu, Minghua Luo, Xie Shichao, Jintao Chen, Zixun Xie, Quan Kuichen, Guo Wei, Mu Xu, and Zedong Chu. Omninav: A unified framework for prospective exploration and visual-language navigation. *arXiv preprint arXiv:2509.25687*, 2025.

[36] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline visual representation learning for embodied navigation. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[37] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors, 2018. URL https://arxiv.org/abs/1810.06543.

[38] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *Advances in neural information processing systems*, 37:5285–5307, 2024.

[39] Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19057–19066, 2025.

[40] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[41] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3554–3560, 2023. doi: 10.1109/IROS55552.2023.10342512.

[42] Taiping Zeng, Bailu Si, and Jianfeng Feng. A theory of geometry representations for spatial navigation. *Progress in Neurobiology*, 211:102228, 2022.

[43] Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, Yuxin Fan, Wenjun Li, Zhibo Chen, Fei Gao, Qi Wu, Zhizheng Zhang, and He Wang. Embodied navigation foundation model, 2025. URL

https://arxiv.org/abs/2509.12129.

[44] Xinxin Zhao, Wenzhe Cai, Likun Tang, and Teng Wang. Imaginenav: Prompting vision-language models as embodied navigator through scene imagination. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 94387–94401, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/eb261df4322a8bd0a73093c4d8a0d02d-Paper-Conference.pdf.

[45] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. Esc: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning*, pages 42829–42842. PMLR, 2023.

[46] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning, 2016. URL https://arxiv.org/abs/1609.05143.