2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

STRIVE: Structured Representation Integrating VLM Reasoning for Efficient Object Navigation

Haokun Zhu[1*], **Zongtai Li**[1*], **Zhixuan Liu**[1],
Wenshan Wang[1], **Ji Zhang**[1], **Jonathan Francis**[1,2], **Jean Oh**[1,3]
[1]Carnegie Mellon University [2]Bosch Center for AI [3]Lavoro AI Research

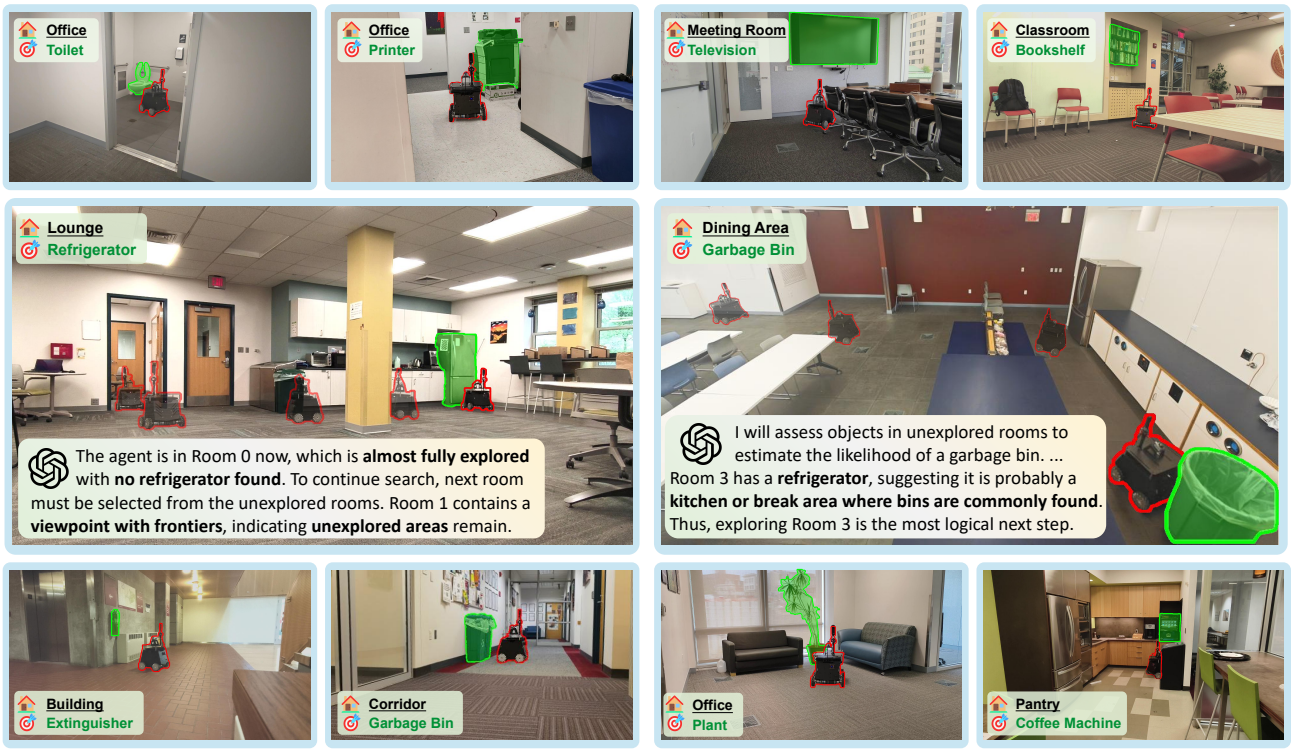

Fig. 1: STRIVE can conduct zero-shot object navigation in diverse and complex real-world environments by leveraging our novel multi-layer representation and efficient two-stage navigation policy.

***Abstract*— Vision-Language Models (VLMs) have been increasingly integrated into object navigation tasks for their rich prior knowledge and strong reasoning abilities. However, applying VLMs to navigation presents two key challenges: effectively parsing and structuring complex environment information and determining *when and how* to query VLMs. Insufficient environment understanding and over-reliance on VLMs (e.g. querying at every step) can easily lead to unnecessary backtracking and reduced navigation efficiency, especially in large continuous environments. To address these challenges, we propose a novel framework that incrementally constructs a multi-layer environment representation consisting of viewpoints, object nodes, and room nodes during navigation. Viewpoints and object nodes facilitate intra-room exploration and accurate target localization, while room nodes support efficient inter-room planning. Building on this structured representation, we propose a novel two-stage navigation policy, integrating high-level planning guided by VLM reasoning with low-level VLM-assisted exploration to efficiently and reliably locate a goal object. We evaluated our approach on four simulated benchmarks (HM3D v1&v2, RoboTHOR, and MP3D), and achieved state-of-the-art performance on both the success rate (SR% $\uparrow 13.1\%$) and navigation efficiency (SPL% $\uparrow 6.2\%$). We further validate our method on a real robot platform, demonstrating strong robustness across 120 episodes in 10 different indoor environments. Project page is available at: *https://zwandering.github.io/STRIVE.github.io/*.**

I. Introduction

Object navigation is a fundamental task in robotics, where an agent must locate an instance of a given object category in unknown environments. This task is particularly challenging, as it requires the agent to understand complex visual information, reason about spatial relationships, and make decisions based on both current and past observations.

Advances in Vision-Language Models (VLMs) [1], [2], [3] have demonstrated strong capabilities in contextual visual understanding and common-sense reasoning. Building on this, recent works [4], [5], [6], [7], [8] have integrated VLMs into object navigation tasks, utilizing their rich prior knowledge, visual understanding, and commonsense reasoning abilities to guide navigation. However, existing approaches often face two significant challenges: First, the input to VLMs

typically lacks a structured representation of the environment and is often restricted to local observations. Without a coherent global view that integrates both current and previous observations, VLMs struggle to reason effectively about the environment and fail to make reasonable navigation decisions. Second, existing methods [6], [8] typically rely on VLMs to select among all frontier viewpoints at each step, without utilizing navigation progress or environment layouts to effectively guide VLMs' reasoning process. Besides, due to VLMs' limited understanding of 3D spatial information [9], [10], [11], they cannot jointly reason about the spatial relationships and the navigation history when evaluating each viewpoint. As a result, their evaluation of viewpoints is largely based on viewpoints' local semantic information, which often leads to redundant navigation behaviors such as backtracking or repeated exploration.

To address these challenges, we propose STRIVE (**ST**ructured **R**epresentation **I**ntegrating **V**LM Reasoning for **E**fficient Object Navigation), a novel framework that incrementally learns a structured representation of the environment and utilizes VLMs' reasoning abilities to guide the navigation. This representation consists of 3 layers: object nodes, viewpoint nodes, and room nodes. Object nodes represent all observed objects, provide rich semantic information about the environment and assist in target localization; Viewpoint nodes discretize the environment into a set of key locations, facilitating structured and efficient intra-room exploration; Room nodes further segment the environment into distinct rooms and facilitate room-level reasoning by the VLM. This multi-layer representation enables a more comprehensive understanding of the environment, allowing VLMs to better utilize their reasoning abilities for more effective decision-making. Furthermore, we design an efficient two-stage navigation policy based on this representation, combining high-level planning guided by the VLM's reasoning and VLM-assisted low-level exploration. Specifically, for the high-level planning, instead of making step-by-step decisions among all viewpoint nodes, the VLM selects the next room to explore based on the spatial layout and semantic information of each room. For low-level exploration within rooms, we employ a traditional frontier-based algorithm for efficient exploration, while leveraging the VLM to decide whether continued exploration of the current room is necessary. Making high-level planning on rooms effectively mitigates the issue of VLMs' insufficient 3D spatial understanding and prevents redundant actions, thereby enhancing navigation efficiency.

We evaluate our method on four widely-used simulated benchmarks: HM3D-v1 [12], HM3D-v2 [13], RoboTHOR [14], and MP3D [15]. STRIVE achieves state-of-the-art (SOTA) results, significantly outperforming 12 existing methods in both Success Rate (SR) and navigation efficiency, measured by Success weighted by Path Length (SPL). This highlights the effectiveness of our proposed multi-layer representation and the VLM-guided reasoning policy in improving object navigation. Specifically, STRIVE achieves 62.9% SR and 34.2% SPL on HM3D-v1, 79.6% SR and 38.7% SPL on HM3D-v2, 68.1% SR and 36.3% SPL on RoboTHOR, and 52.3% SR and 23.1% SPL on MP3D. Besides, we also conduct 120 real-world experiments across 10 different indoor environments on a Mecanum wheel platform [16], demonstrating the effectiveness and robustness of our method in real-world scenarios.

II. Related Works

Object Navigation. Existing object navigation methods are typically categorized into end-to-end learning approaches and modular approaches. End-to-end methods [17], [18], [19], [20], [21], [22], [23] use reinforcement learning to directly map observations to actions, but often suffer from low sample efficiency and poor generalization. In contrast, modular methods [24], [25], [26], [27], [28], [4], [29] decompose navigation into steps such as mapping, planning, and action execution, and often build semantic maps in bird's-eye view or 3D space to facilitate more interpretable and scalable navigation behavior. With the emergence of foundation models [1], object navigation has advanced towards zero-shot, open-vocabulary setting [27], [28], [30], [4], [29]. We also leverage VLM's reasoning abilities to improve zero-shot object navigation, but we employ a novel representation and a two-stage policy, enabling more efficient and effective VLM guidance.

VLM-guided Navigation. With internet-scale training data, Vision-Language Models (VLMs) [31], [1], [2], [3] have shown strong common-sense reasoning abilities and have been widely applied in Object Navigation tasks to guide the decision-making process. For example, InstructNav [5] leverages multi-sourced value maps to model key navigation elements. SG-Nav [4] constructs 3D scene graphs and prompts LLMs with structural relationships, while DORAEMON [29] combines RAG-VLM for understanding ontological tasks and Policy-VLM for enhanced policy planning. However, due to VLMs' limited 3D spatial understanding ability [9], [10], [11], over-reliance on VLMs for navigation can lead to inefficient behavior, such as frequent backtracking. To address this, we propose a two-stage navigation policy that combines VLM-guided high-level planning with VLM-assisted frontier-based low-level exploration strategies, leveraging the reasoning strength of VLMs while ensuring efficient and robust navigation behavior.

Scene Representation for Indoor Navigation. Scene representation is crucial for transforming raw observations into structured information for decision-making in navigation tasks. Frontier-based methods [25], [32], [33], [34] record frontiers on a grid map and integrate semantic information to guide exploration. In contrast, graph-based methods represent the environment as structured scene graphs to support navigation. Prior works [4], [8] use scene graphs to summarize semantic information and let VLMs to select among frontier locations. Others [35], [6] explicitly construct viewpoints in the scene graph, enabling VLMs to reason over the graph and choose among viewpoints to guide navigation. Unlike traditional scene graphs [36], [6], where viewpoints are typically derived from Voronoi partitions, we discretize the environment into semantically meaningful

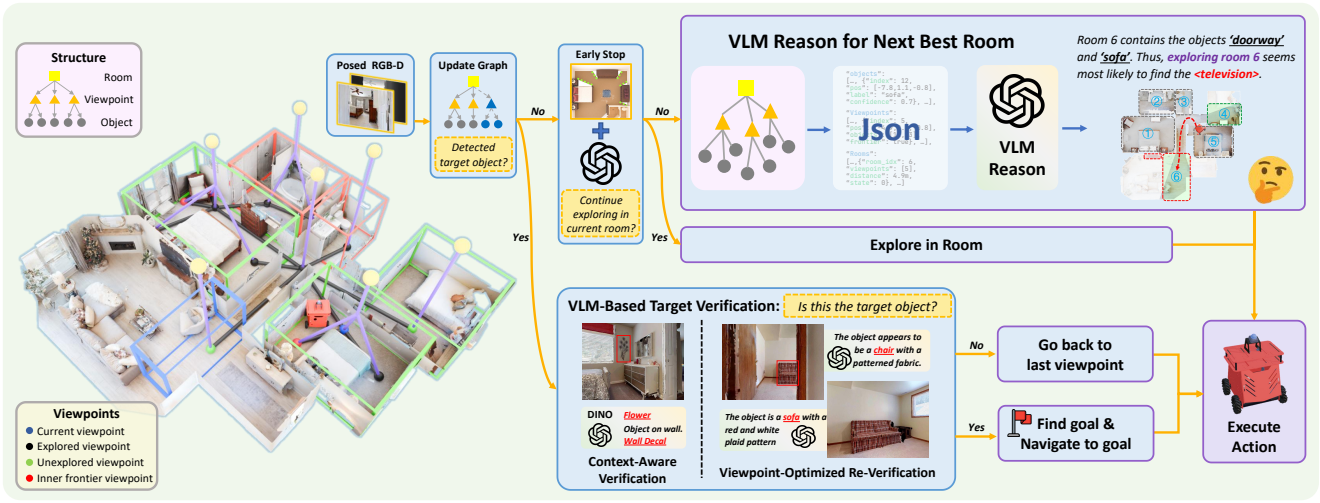

Fig. 2: **Overview of STRIVE.** We construct a multi-layer representation $\mathcal{R}$ (Sec. III-A) on-the-fly, consisting of object, viewpoint, and room nodes, which serves as a structured input for VLM. Based on $\mathcal{R}$, we introduce a two-stage navigation policy, where the VLM reasons and plans at room-level (Sec. III-B.2), while the agent explores in room at the viewpoint-level using a VLM-assisted frontier-based navigation strategy (Sec. III-B.1) and VLM-based target verification (Sec. III-B.3).

regions to select viewpoint nodes. As the middle layer, these viewpoints bridge the spatial structure (room nodes) and semantic content (object nodes), forming a structured representation facilitating VLM reasoning.

III. Method

Task Definition: In Object Navigation, the agent is required to find an instance of a given object category (e.g. Find the *bed*.) in an unknown environment. At each time step t, the agent receives a posed RGB-D observation $\mathbf{O}_t = \{I_t, D_t, P_t = \langle \mathbf{p}_t, \mathbf{R}_t \rangle\}$, where I_t is the RGB image, D_t is the depth map, and P_t is the camera pose. The navigation policy then predicts an action $a_t \in \{\texttt{move\_forward}, \texttt{turn\_left}, \texttt{turn\_right}, \texttt{stop}\}$. The task is considered successful if the agent stops within d_s meters of the target object in less than T steps.

Overview: Fig. 2 provides an overview of STRIVE. STRIVE enables the VLM to reason at the room level while guiding the agent's in-room exploration at the viewpoint level. We describe the representation construction in Sec. III-A and the two-stage navigation policy in Sec. III-B.

A. Multi-layer Environment Representation

We propose a framework that models the environment using a three-layer graph representation $\mathcal{R}$, where each layer corresponds to a specific type of node: object nodes $V^{obj} = \{v_i^{obj}\}$, viewpoint nodes $V^{vp} = \{v_i^{vp}\}$, and room nodes $V^{room} = \{v_i^{room}\}$. Edges encode spatial and semantic relationships across nodes. We elaborate on the graph construction process in following sections.

1) Viewpoint Nodes

Inspired by [37], we construct a skeleton graph to discretize the environment. The graph is incrementally built as the agent navigates—each time the agent reaches a new viewpoint, it updates the graph. We define a coverage range

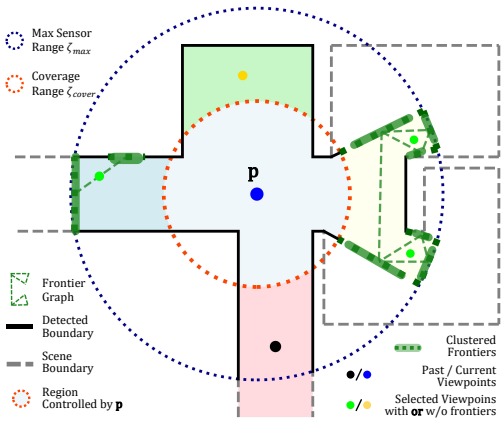

Fig. 3: **Visualization of the viewpoint selection algorithm.** *Green* and *yellow* nodes are the selected viewpoints.

ζ_{cover} as the maximum distance within which objects are assumed to be reliably detected. Each viewpoint node thus controls a circular region of radius ζ_{cover}. Edges between viewpoint nodes indicate direct traversability. The maximum sensor range ζ_{max} denotes the effective measurable distance of the depth camera.

Node: For efficient exploration, the goal is to control the entire area exhaustively with the minimal number of viewpoint nodes. New viewpoint nodes are preferably selected at locations that maximize the visibility of frontiers. To select the future viewpoint nodes, we begin by taking the input of agent's position $\mathbf{p}$, coverage range ζ_{cover}, and observed point cloud. First, we exclude the region controlled by the current viewpoint node, thereby dividing the remaining point cloud into separate regions, which fall into two categories: regions with or without frontiers.

Regions with frontiers: Inspired by [38], we use frontiers to guide viewpoint selection. The detected frontiers are clustered into frontier edge segments, each representing a candi-

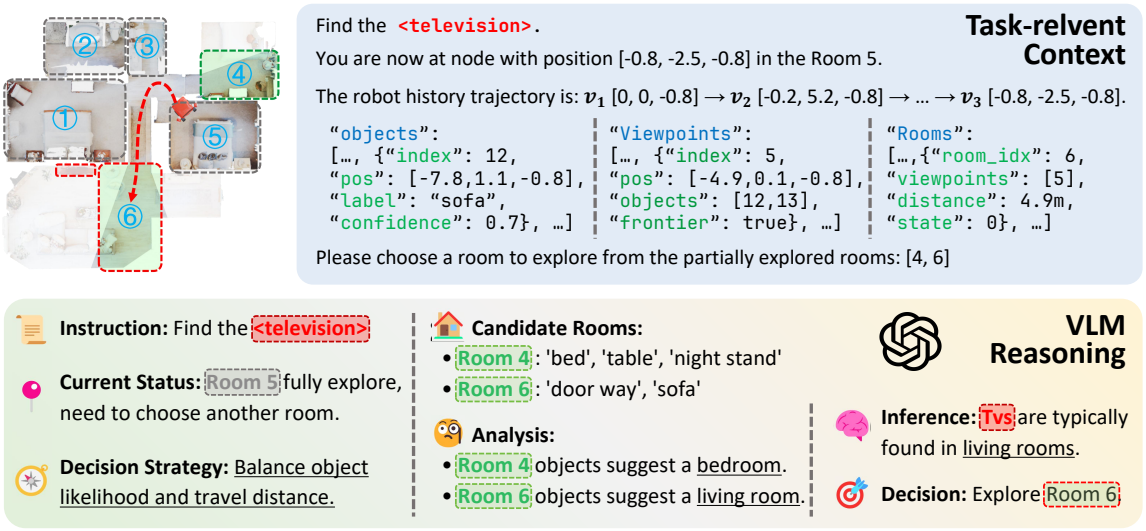

Fig. 4: Visualization of the structured prompt and the VLM's reasoning process of selecting the next best room.

date boundary for exploration. The segments are organized into a graph $G_{frontier}$ (green dashed lines), where nodes denote segments and edges denote mutual visibility. Then the *Maximum Clique* is iteratively removed from $G_{frontier}$. For each removed clique, its center is added as a new viewpoint node v_i^{vp} (*green node* in Fig. 3) to our representation $\mathcal{R}$. This new node serves as the exploration anchor for all frontier segments within the clique.

Regions without frontiers: For these regions, the center of the region is directly added as a new viewpoint node v_i^{vp} (*yellow nodes* in Fig. 3) to our representation $\mathcal{R}$.

Edges between V^{vp}: We evaluate straight-line traversability between each pair of viewpoint nodes. An edge is added if the direct line between two nodes is free of obstacles.

2) Object Nodes

We leverage open-vocabulary detection and segmentation methods [39], [40] to obtain segmented 3D object instances. Specifically, given the image I_t at time step t, we first use Grounding DINO [39] to perform open-vocabulary object detection. The detected bounding boxes are then fed into SAM [40] to obtain segmentation masks. Finally, by combining the depth map D_t and the camera pose P_t, we reconstruct the 3D point cloud of each detected object. For each object, we instantiate an object node at its center, recording attributes such as 3D position, point cloud, predicted label, confidence score and 3D bounding box. Newly instantiated nodes are merged with previously observed nodes if they correspond to the same physical object.

Edges between V^{vp} and V^{obj}: An edge is added between v_i^{vp} and v_j^{obj} if v_j^{obj} is within the cover range ζ_{cover} of v_i^{vp} and is visible from v_i^{vp}. An object can be associated with multiple viewpoints. If an object isn't connected to any viewpoint, we connect it to the closest visible viewpoint.

3) Room Nodes

Following [41], [36], we identify all walls in the environment and iteratively dilate them to segment the environment into connected components. Then each connected component is added as a room node v_i^{room} to our representation $\mathcal{R}$. Finally, edges are added between each room node and the viewpoint nodes located within the corresponding room.

Alg 1: Viewpoint Construction Process

Require: Position p, Point Cloud $\mathcal{P}$
Output: Updated Viewpoint Nodes $\mathcal{V}^{vp}$

Calculate controlled region P_c
$\hat{\mathcal{P}} \leftarrow \mathcal{P} - P_c$
Regions $\mathcal{R} \leftarrow \texttt{Cluster}(\hat{\mathcal{P}})$
for region r_i in $\mathcal{R}$ **do**
  if r_i has no frontiers **then**
    $v^{vp} \leftarrow \texttt{Center}(r_i)$, Add new v^{vp} to $\mathcal{V}^{vp}$
  else
    $\mathcal{C}_i^F \leftarrow \texttt{FindFrontierCliques}(r_i)$
    for $c_{i,j}^F$ in $\mathcal{C}_i^F$ **do**
      $v^{vp} \leftarrow \texttt{Center}(c_{i,j}^F)$, Add new v^{vp} to $\mathcal{V}^{vp}$
    end for
  end if
end for

B. Object Navigation Policy

In this section, we present our efficient two-stage navigation policy, where the VLM performs high-level reasoning and planning at the room level, while the agent conducts fine-grained exploration within each room at the viewpoint level, guided by a VLM-assisted frontier-based strategy and VLM-based target verification.

1) Explore in Room with Early Stop

For efficient low-level exploration within rooms, we introduce VLM-assisted early stop, combining VLM with traditional frontier-based algorithm. We first classify frontiers into two types: *True Frontiers*, which lie along room boundaries indicating incomplete exploration, and *Inner Frontiers*, resulting from objects' occlusions. The agent iteratively navigates to the nearest viewpoint with *True Frontiers* and

explores until all *True Frontiers* are cleared. If *Inner Frontiers* still remain in the current room, we query the VLM to decide whether further exploration inside this room is necessary.

2) Next Best Room

In situations where exploration of the current room is completed *without* finding the target object, we must determine the next room to explore. To guide this decision, we leverage VLM's commonsense reasoning abilities by providing task-relevant context and general exploration heuristics.

The task-relevant context is consolidated into a combined *Prompt* as described in Fig. 4, which contains *1)* target object instruction, *2)* agent's current state, *3)* agent's navigation history, and *4)* the environment representation $\mathcal{R}$ formatted as a JSON file.

Besides the task-relevant context, we also provide the VLM with general exploration heuristics. Specifically, we explicitly instruct the VLM to evaluate two factors: *1)* The semantic similarity between the objects in each room and the target object. *2)* The distance from the agent's current position to each room, aiming to optimize the exploration path by minimizing unnecessary backtracking. Using a Chain-of-Thought reasoning strategy, the VLM selects the most suitable unexplored room for further exploration. Finally, the viewpoint closest to the current position in the selected room is chosen as the next action viewpoint.

Notably, in the later navigation stage, continuing forward is more effective than backtracking, as remaining steps may not allow long detours. In light of this, we introduce a penalized distance that weights the geodesic distance by factors reflecting the steps already taken and the number of explored viewpoints along the path to each candidate room.

3) VLM-Based Target Verification

Accurate detection of the target object is crucial in object navigation. However, relying solely on detection model [39] often results in false positives. To address this, we propose incorporating the VLM to verify detected target objects, leveraging its ability to reason about the contextual information of the surrounding environment.

Context-Aware Verification: When the agent detects a potential target object, we prompt the VLM with the detected object and its surrounding visual context for verification. The VLM leverages both the object's appearance and its surrounding semantic information to determine whether it matches the target category, e.g. recognizing a painting of plant as a 'decoration' rather than 'plant'.

Viewpoint-Optimized Re-Verification: The agent may initially observe and detect the target object from a suboptimal viewpoint (e.g., under occlusion or from a long distance), resulting in inaccurate detection. To address this, we perform a second observation from a better viewpoint. Unlike the baseline method [4], which further observes the target object from multiple viewpoints, we compute the optimal viewpoint along the path from the current position to the target object and only perform one VLM re-verification at that viewpoint. This strategy improves detection accuracy without sacrificing navigation efficiency.

IV. Experiment

A. Experiment Setup

We evaluate our method by comparing it with state-of-the-art methods on the Object Navigation task in Habitat [45] simulator. We also conduct real-world experiments across diverse environments.

Dataset: We perform simulated experiments on 4 datasets: 1) HM3D-v1 [12], including 2000 episodes across 20 high-fidelity scenes with 6 target object categories. 2) HM3D-v2 [45], including 1000 episodes across 36 high-fidelity scenes with 6 target object categories. 3) RoboTHOR [14], including 1800 episodes across 15 scenes with 12 target object categories. 4) MP3D [15], including 2195 episodes across 11 scenes with 21 target object categories.

Evaluation Metrics: Following [13], we use 4 metrics to evaluate the performance: 1) Success Rate (SR): the percentage of episodes in which the agent reaches the target object within a success distance. 2) Success Penalized by Path Length (SPL): the success rate penalized by the ratio of the shortest path length to the actual path length. 3) Distance to Goal (DTG): the final distance to the target object at the end of the episode. 4) SoftSPL: replacing the binary success term in SPL with a "soft" value that indicates the progress made by the agent towards the goal.

Implementation Details: Each episode allows a maximum of 500 steps and a success distance of 1.0m. The agent observes the environment using a 640×480 RGB-D image, with depth values from 0.5m to 5.0m and a horizontal field of view (HFOV) of 79°. The camera is mounted 0.88m above the ground. The agent moves forward by 0.25m per step and rotates by 30°. For VLM, we use Gemini [3] (gemini-2.0-flash), and for object detection and segmentation, we use MM-GroundingDINO-L [39] and SAM-2.1-L [40]. All experiments are conducted on RTX4090 GPUs.

For real-world experiments, we deploy `STRIVE` on the Mecanum wheel platform [16], which is equipped with a Ricoh Theta Z1 360-degree camera for RGB image capturing and a Livox Mid-360 LiDAR for 3D point cloud acquisition. To maintain compatibility with the input format in simulation, the collected point clouds are converted into depth maps when necessary.

B. Quantitative Results in Simulator

We compare `STRIVE` with state-of-the-art object navigation methods in different settings as shown in Tab. I. `STRIVE` significantly outperforms all baselines across all benchmarks. Compared to the second-best methods, it achieves improvements of +6.4% SR / +3.6% SPL on HM3D-v1, +13.1% SR / +6.2% SPL on HM3D-v2, +20.6% SR / +12.3% SPL on RoboTHOR, and +11.2% SR / +5.5% SPL on MP3D. For HM3D-v1, since our method is designed for single-floor object navigation, we additionally evaluate only on episodes where the agent's starting position and the target object are on the same floor. Under this setting, `STRIVE` achieves 72.7% SR and 38.2% SPL. Overall, the improvement in SPL shows that our representation and navigation policy effectively improve navigation efficiency. Furthermore, increased

Method	Open-Set	Zero-Shot	HM3D-v1		HM3D-v2		RoboTHOR		MP3D	
			SR (%) ↑	SPL (%) ↑	SR (%) ↑	SPL (%) ↑	SR (%) ↑	SPL (%) ↑	SR (%) ↑	SPL (%) ↑
SemEXP [24]	✗	✗	-	-	-	-	-	-	36.0	14.4
PONI [25]	✗	✗	-	-	-	-	-	-	31.8	12.1
ZSON [42]	✓	✗	25.5	12.6	-	-	-	-	15.3	4.8
L3MVN [27]	✗	✓	50.4	23.1	36.3	15.7	41.2	22.5	34.9	14.5
ESC [28]	✓	✓	39.2	22.3	-	-	38.1	22.2	28.7	11.2
VoroNav [6]	✓	✓	42.0	26.0	-	-	-	-	-	-
VLFM [30]	✓	✓	52.5	30.4	63.6	32.5	-	-	36.4	17.5
VLFM♯ [30]	✓	✓	50.9	23.6	56.9	27.5	-	-	32.5	15.9
SG-Nav [4]	✓	✓	54.0	24.9	49.6	25.5	47.5	24.0	40.2	16.0
OpenFMNav [43]	✓	✓	54.9	24.4	-	-	44.1	23.3	37.2	15.7
TriHelper [44]	✓	✓	56.5	25.3	-	-	-	-	-	-
InstructNav [5]	✓	✓	-	-	58.0	20.9	-	-	-	-
DORAEMON [29]	✓	✓	55.6	21.4	66.5	20.6	-	-	41.1	15.8
STRIVE	✓	✓	**62.9**	**34.2**	**79.6**	**38.7**	**68.1**	**36.3**	**52.3**	**23.1**
STRIVE*	✓	✓	**72.7**	**38.2**	-	-	**-**	**-**	**-**	**-**

TABLE I: Comparison with SOTA methods with different settings on HM3D-v1, HM3D-v2, RoboTHOR, and MP3D datasets. We report the Success Rate (SR) and Success weighted by Path Length (SPL) metrics. Best results are **in bold**, second best are underlined. VLFM♯ replaces pre-trained PointNav module with a shortest-path planner for fair comparison. STRIVE* denotes evaluation restricted to episodes where agent's starting position and the target object are located on the same floor.

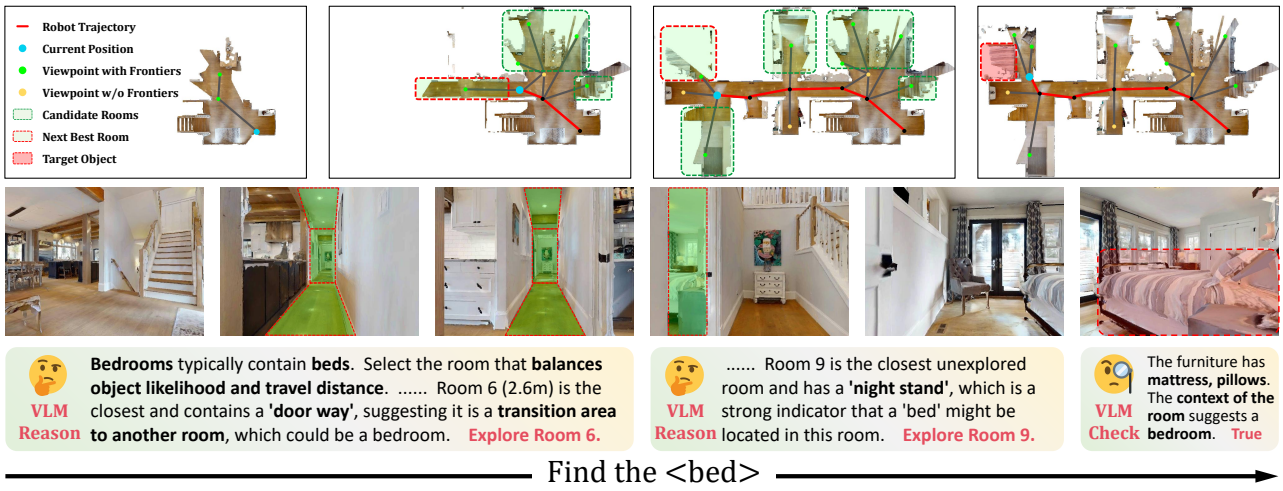

Fig. 5: **Qualitative visualization of STRIVE.** The 1st&2nd steps show the VLM's reasoning process, where it selects Room 6 and 9 by jointly considering room-layout ('doorway'), semantic cues ('nightstand') and travel cost (penalized distance). The final step shows VLM-based verification, using contextual cues (e.g., mattress, pillows) to confirm the target object.

navigation efficiency enables the agent to explore a larger area within a limited number of steps. The improvement in SR results from the combined effects of more efficient navigation, better utilization of the VLM's reasoning capabilities, and more accurate VLM-based verification.

C. Qualitative Results in Simulator

We visualize the navigation process of STRIVE in Fig. 5. The results demonstrate that our structured representation enables the VLM to reason effectively about both spatial layout (e.g., a room with a "doorway" can lead to other rooms) and semantic cues (e.g., nightstands suggesting bedrooms), leading to improved room selection. Furthermore, the VLM balances the likelihood of finding the target object against travel distance cost when planning room-to-room exploration. It also leverages contextual information to re-verify detected objects and effectively reduces false positives.

Object Category	Sofa	Chair	Monitor	Table	Garbage bin	Average
Success Rate(%)	85	90	70	85	75	81
Runtime(s)	68.26	45.22	70.86	68.80	62.53	63.13

TABLE II: **Real-World Quantitative Results.** We conducted 20 episodes for each object category and report the success rate and average runtime.

D. Real-World Experiments

We conduct 120 real-world experiments across 15 different target categories and 10 different environments, including offices, meeting rooms, classrooms, lounges, dining areas, corridors, and kitchens. Part of the experiment environments and results are shown in Fig. 1. Compared to simulation, real-world deployment presents additional challenges. Lidar-captured point clouds are much sparser than depth maps, and real environments are often more cluttered, introducing noise that affects both exploration and object detection. Despite

V^{vp}	V^{obj}	V^{room}	SR ↑	SPL ↑	S-SPL ↑	DTG(m) ↓
✓	✗	✗	71.3	33.2	35.2	1.86
✓	✓	✗	72.4	34.0	36.1	1.95
✓	✗	✓	72.9	33.8	35.4	1.86
✓	✓	✓	**75.0**	**34.9**	**36.5**	**1.80**

TABLE III: **Ablation study of representation on HM3D.** We adopt a *viewpoint-level navigation policy* for experiment consistency.

	SR ↑	SPL ↑	S-SPL ↑	DTG(m) ↓	Tokens ↓
w/o Early Stop	74.8	34.8	36.4	1.62	-
w/o Penalized Dist	73.7	36.1	36.9	1.47	-
w/o VLM-Verify	72.1	32.7	34.1	1.83	-
Viewpoint Policy	75.0	34.9	36.5	1.80	22935
STRIVE	**79.6**	**38.7**	**38.9**	**1.29**	**8068**

TABLE IV: **Ablation study of navigation policy components on HM3D.** Viewpoint Policy stands for VLM planning on viewpoint-level.

these challenges, our agent demonstrates robust performance.

We elaborate on 2 difficult environments in Fig. 1. In the left scenario, the agent is initialized inside a small enclosed room connected to a larger lounge area. Despite the presence of inner frontiers—regions occluded by furniture—the agent correctly decides to abort exhaustive exploration of this room. Instead, it exits the room early and shifts its attention to unexplored rooms nearby, where it ultimately locates the target object. In the right scenario, the agent is initialized in a dining area and instructed to find a 'Garbage bin'. STRIVE successfully uses the VLM to reason on semantic associations ('refrigerator' and 'bins') and find the target object efficiently.

We also evaluated the runtime performance of our method in real-world scenarios. In Tab. II, we present the success rates and average runtimes for 100 episodes over 5 object categories, with 20 episodes for each object category. Our method achieves high success rates and short average times over all 5 categories, demonstrating its effectiveness in the real-world environment.

E. Ablation Study

Multi-layer Representation: We conduct an ablation study to evaluate the contributions of object nodes and room nodes in our multi-layer scene representation. Since room-level navigation depends on room nodes, it cannot be used when they are removed. To ensure consistency, we adopt a basic navigation policy allowing the VLM to guide navigation at the viewpoint level instead of the room level. As shown in Tab. III, both object nodes and room nodes contribute significantly to performance improvement. Including object nodes improves the agent's ability to localize target objects, while adding room nodes enhances its understanding of the environment layout. Using both layers together leads to a substantial improvement over using either individually.

Navigation Policy: We conduct ablation studies to evaluate the effectiveness of our navigation policy, VLM-assisted early stopping, penalized distance, and VLM-based verification, as summarized in Tab. IV. The last two rows compare our room-level planning policy against a basic viewpoint-level approach. Results show that room-level planning enables the agent to better leverage the VLM's reasoning capabilities, significantly boosting performance. We also report the average VLM token usage per episode. By querying the VLM only for room-level planning, our method significantly reduces token consumption compared to viewpoint-level planning. Finally, the VLM-assisted early stopping, penalized distance, and verification modules each contribute to further performance gains.

V. Conclusion

In this paper, we introduce STRIVE, a novel framework that incrementally constructs a structured scene representation and leverages VLM's reasoning capabilities to achieve efficient object navigation. STRIVE incrementally builds a multi-layer representation of the environment, consisting of room, viewpoint and object nodes. Based on this representation, we design an efficient two-stage VLM-guided navigation policy, which leverages VLM reasoning for room-level planning while using VLM together with traditional frontier-based methods for efficient exploration within rooms. To further improve robustness, we incorporate VLM-based target verification, utilizing VLMs' contextual understanding to improve detection accuracy. Experiments across four simulated benchmarks demonstrate that STRIVE achieves state-of-the-art performance, significantly improving both success rate and navigation efficiency. Furthermore, our real-world experiments show the robustness and practicality of STRIVE in navigating complex and diverse real-world environments.

ACKNOWLEDGMENT

This work was in part supported by NSF IIS-2112633.

References

[1] 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 *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[2] T. Wu, G. Yang, Z. Li, K. Zhang, Z. Liu, L. Guibas, D. Lin, and G. Wetzstein, "Gpt-4v (ision) is a human-aligned evaluator for text-to-3d generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 22 227–22 238.

[3] 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.

[4] H. Yin, X. Xu, Z. Wu, J. Zhou, and J. Lu, "Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 5285–5307, 2024.

[5] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "Instructnav: Zero-shot system for generic instruction navigation in unexplored environment," in *8th Annual Conference on Robot Learning*.

[6] 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.

[7] S. Saxena, B. Buchanan, C. Paxton, B. Chen, N. Vaskevicius, L. Palmieri, J. Francis, and O. Kroemer, "Grapheqa: Using 3d semantic scene graphs for real-time embodied question answering," *arXiv preprint arXiv:2412.14480*, 2024.

[8] J. Loo, Z. Wu, and D. Hsu, "Open scene graphs for open-world object-goal navigation," in *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

[9] Z. Zhang, F. Hu, J. Lee, F. Shi, P. Kordjamshidi, J. Chai, and Z. Ma, "Do vision-language models represent space and how? evaluating spatial frame of reference under ambiguities," *arXiv preprint arXiv:2410.17385*, 2024.

[10] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 455–14 465.

[11] Z. Qi, Z. Zhang, Y. Fang, J. Wang, and H. Zhao, "Gpt4scene: Understand 3d scenes from videos with vision-language models," *arXiv preprint arXiv:2501.01428*, 2025.

[12] K. Yadav, S. K. Ramakrishnan, A. Gokaslan, O. Maksymets, R. Jain, R. Ramrakhya, A. X. Chang, A. Clegg, M. Savva, E. Undersander, D. S. Chaplot, and D. Batra, "Habitat challenge 2022," https://aihabitat.org/challenge/2022/, 2022.

[13] K. Yadav, J. Krantz, R. Ramrakhya, S. K. Ramakrishnan, J. Yang, A. Wang, J. Turner, A. Gokaslan, V.-P. Berges, R. Mootaghi, O. Maksymets, A. X. Chang, M. Savva, A. Clegg, D. S. Chaplot, and D. Batra, "Habitat challenge 2023," https://aihabitat.org/challenge/2023/, 2023.

[14] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, L. Weihs, M. Yatskar, and A. Farhadi, "Robothor: An open simulation-to-real embodied ai platform," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.

[16] J. Zhang, "Autonomy stack for mecanum wheel platform," https://github.com/jizhang-cmu/autonomy_stack_mecanum_wheel_platform, 2024, accessed: 2025-04-29.

[17] R. Dang, L. Wang, Z. He, S. Su, J. Tang, C. Liu, and Q. Chen, "Search for or navigate to? dual adaptive thinking for object navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8250–8259.

[18] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," *arXiv preprint arXiv:1911.00357*, 2019.

[19] J. Ye, D. Batra, A. Das, and E. Wijmans, "Auxiliary tasks and exploration enable objectgoal navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 117–16 126.

[20] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, "Habitat-web: Learning embodied object-search strategies from human demonstrations at scale," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5173–5183.

[21] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "Pirlnav: Pretraining with imitation and rl finetuning for objectnav," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 896–17 906.

[22] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8846–8852.

[23] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," *arXiv preprint arXiv:1810.06543*, 2018.

[24] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4247–4258, 2020.

[25] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 890–18 900.

[26] J. Zhang, L. Dai, F. Meng, Q. Fan, X. Chen, K. Xu, and H. Wang, "3d-aware object goal navigation via simultaneous exploration and identification," 2023. [Online]. Available: https://arxiv.org/abs/2212.00338

[27] B. Yu, H. Kasaei, and M. Cao, "L3mvn: Leveraging large language models for visual target navigation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2023, p. 3554–3560. [Online]. Available: http://dx.doi.org/10.1109/IROS55552.2023.10342512

[28] 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," 2023. [Online]. Available: https://arxiv.org/abs/2301.13166

[29] T. Gu, L. Li, X. Wang, C. Gong, J. Gong, Z. Zhang, Y. Xie, L. Ma, and X. Tan, "Doraemon: Decentralized ontology-aware reliable agent with enhanced memory oriented navigation," *arXiv preprint arXiv:2505.21969*, 2025.

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

[31] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," 2022. [Online]. Available: https://arxiv.org/abs/2201.12086

[32] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, "How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers," *arXiv preprint arXiv:2305.16925*, 2023.

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

[34] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, vol. 8, no. 79, p. eadf6991, 2023.

[35] D. An, H. Wang, W. Wang, Z. Wang, Y. Huang, K. He, and L. Wang, "Etpnav: Evolving topological planning for vision-language navigation in continuous environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[36] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," 2022.

[37] F. Yang, D.-H. Lee, J. Keller, and S. Scherer, "Graph-based topological exploration planning in large-scale 3d environments," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 12 730–12 736.

[38] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE, 1997, pp. 146–151.

[39] X. Zhao, Y. Chen, S. Xu, X. Li, X. Wang, Y. Li, and H. Huang, "An open and comprehensive pipeline for unified object grounding and detection," *arXiv preprint arXiv:2401.02361*, 2024.

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

[41] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," in *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

[42] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 340–32 352, 2022.

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

[44] L. Zhang, Q. Zhang, H. Wang, E. Xiao, Z. Jiang, H. Chen, and R. Xu, "Trihelper: Zero-shot object navigation with dynamic assistance," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 10 035–10 042.

[45] X. Puig, E. Undersander, A. Szot, M. D. Cote, R. Partsey, J. Yang, R. Desai, A. W. Clegg, M. Hlavac, T. Min, T. Gervet, V. Vondruš, V.-P. Berges, J. Turner, O. Maksymets, Z. Kira, M. Kalakrishnan, J. Malik, D. S. Chaplot, U. Jain, D. Batra, A. Rai, and R. Mottaghi, "Habitat 3.0: A co-habitat for humans, avatars and robots," 2023.