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

2026 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-8160-2/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICRA57385.2026.11696652

Find Anything Like Humans: Online Semantic Mapping And Coarse-to-fine Navigation in Dynamic Environments

Yutian Zhang, Jianyu Zhang and Mengyuan Liu*

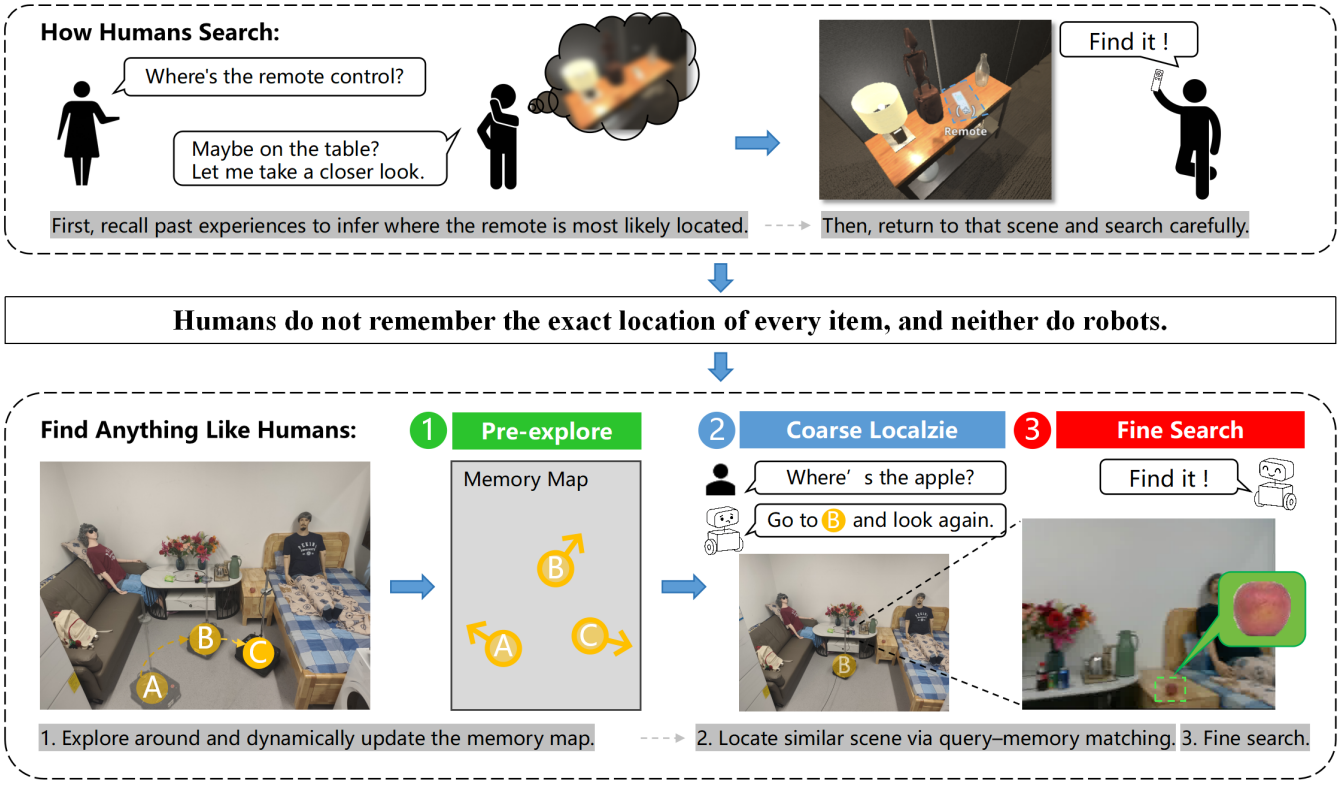


Fig. 1. **Human-inspired coarse-to-fine search.** When people look for an item, they seldom know its exact location: they first recall likely regions from memory and then return to inspect them carefully. Our method mirrors this coarse-to-fine routine: **(1) Pre-explore** the scene to build a memory map; **(2) Coarse localization** by retrieving likely views via query-memory matching (text or reference image); and **(3) Fine search** in the selected view to segment and verify the target before interaction.

***Abstract*— Enabling robots to follow natural-language instructions in dynamic environments requires scene representations that support open-ended queries, adapt to change, and operate in real time. However, existing approaches often rely on prompt-driven pipelines and dense 3D reconstruction, which limit flexibility and impose high computational cost. We propose Find Anything Like Humans (FALH), an online framework inspired by how people search: recalling likely regions from memory and then verifying them up close. During exploration, FALH constructs a compact scene memory by pairing visual features with observed poses, using class-agnostic detectors without predefined prompts. At query time, it retrieves candidate locations via feature similarity, then performs local verification to confirm the target and estimate a precise 3D goal. All components operate on a unified pose-feature representation that supports efficient recall, online updates, and robust performance in cluttered, changing scenes. Experiments in simulation (HM3D, AI2-THOR) and in the real world show that FALH outperforms object-centric baselines in both success rate and responsiveness under limited resources. Code, videos, and datasets are available at: https://github.com/yutian929/Find-Anything-Like-Humans.**

I. INTRODUCTION

Natural language navigation refers to the task of enabling a robot to follow free-form language instructions in order to locate and reach specific targets within a dynamic environment. This process typically involves interpreting user-provided descriptions, identifying objects or locations based on those descriptions, and planning actions to reach the intended goal. A critical component of this process

This work was supported by National Key Research and Development Program of China (No. 2024YFB4709800), Guangdong S&T Program (No. 2024B0101050002), Shenzhen Innovation in Science and Technology Foundation for The Excellent Youth Scholars (No. RCYX20231211090248064).

Yutian Zhang, Jianyu Zhang and Mengyuan Liu are with the State Key Laboratory of General Artificial Intelligence, Peking University, Shenzhen Graduate School, Shenzhen 518055, China.

Corresponding Author: liumengyuan@pku.edu.cn.

Authorized licensed use limited to: Peking University. Downloaded on October 21,2026 at 04:06:18 UTC from IEEE Xplore. Restrictions apply.

is the construction of a scene representation or map that summarizes what the robot has observed, where relevant objects might be located, and how the environment is changing over time. Studying this capability is essential for building intelligent agents that can interact naturally with humans in real-world scenarios, such as domestic assistance, search-and-rescue operations, and warehouse automation. Furthermore, advancing natural language navigation supports progress in several foundational research areas, including language grounding, open-vocabulary perception, and real-time spatial reasoning. To be effective in practical settings, such systems must construct meaningful and up-to-date scene representations from onboard sensing, adapt to diverse language instructions, and operate in real time as environments evolve.

To support these navigation tasks, recent research has explored open-vocabulary mapping [1]–[4] and scene understanding [5]–[8] as a means of bridging the gap between language and visual perception. However, many existing systems still rely heavily on prompt-driven pipelines and dense 3D reconstruction. A typical method uses predefined word lists to guide vision-language models [9]–[14] in detecting and recording objects from multiple views, then fuses these outputs into point clouds, voxel grids, or semantic fields. While such approaches go beyond closed-set detection [15], [16], they introduce several limitations. These systems depend on the specific prompts provided during mapping—objects that are not explicitly included in the prompt list are typically ignored, even if they appear in the scene. This means that relevant visual evidence may be discarded early on, and the resulting map lacks information about those objects. As a consequence, if such objects are later referred to in a navigation query, the system is unable to retrieve them from the map, even though they were actually observed. Moreover, dense geometric representations [17]–[24] and object-centric maps are computationally expensive to update and maintain, especially in dynamic environments. These constraints hinder the responsiveness needed for real-time navigation. To address these challenges, it is important to explore alternatives that decouple mapping from fixed vocabularies and instead focus on efficient, adaptable scene memories that reflect what the robot actually observes and can be directly queried by language.

To address the limitations of prompt-driven and resource-intensive mapping systems, we introduce Find Anything Like Humans (FALH), an online framework for language-guided navigation inspired by how people search in everyday environments. Rather than memorizing the exact location of every object, humans typically recall general areas where something was last seen and then return to inspect those regions more carefully. FALH adopts a similar coarse-to-fine strategy: it builds a compact scene memory during exploration and later uses that memory to locate and verify targets based on natural language or image queries. Crucially, this memory is constructed without relying on predefined vocabulary prompts, enabling broader coverage and more flexible retrieval when user expressions vary.

TABLE I

COMPARISON OF RECENT SEMANTIC MAPPING SYSTEMS.

System		Open-Vocab	Prompt-Free	Online	Dynamic
Hydra [25]	RSS'22			✓	
Khronos [26]	RSS'24			✓	✓
ConceptGraphs [1]	ICRA'24	✓			
HOV-SG [2]	ICRA'24	✓			
CLIO [3]	RA-L'24	✓		✓	
OpenIN [4]	RA-L'25	✓			✓
FALH (Ours)		✓	✓	✓	✓

The core of FALH is a scene memory built from class-agnostic detections and segmentations, linking visual features to the robot's pose to capture both what was observed and where it occurred, without requiring object labels or prompt lists during mapping. As the robot revisits previous poses, the memory is automatically updated, avoiding the need for complex matching, insertion, or deletion operations. When a query is issued, the system performs coarse localization by embedding the query into the same feature space and retrieving a ranked list of candidate locations based on similarity to stored observations. The robot then visits these candidates in order, performing fine search at each by verifying whether the target appears in the current view. If a confident match is found, depth sensing is used to estimate a precise 3D goal; otherwise, the search proceeds to the next best match. This combination of coarse retrieval and local confirmation allows the robot to remain efficient and responsive in cluttered, dynamic environments. All components operate on a unified pose–feature representation that supports compact storage, fast retrieval, and robust adaptation to environmental changes, resulting in an integrated and flexible system for open-ended object navigation.

To summarize, our key contributions are:

- We propose a human-like, coarse-to-fine search strategy that first recalls likely regions from a robot-centric memory and then performs precise local verification, enabling target acquisition without maintaining a persistent global object-level map.
- We design a prompt-free perception front end that builds scene memory from class-agnostic proposals and vision–language features, enabling broader generalization to unseen objects and diverse instructions.
- We conduct extensive experiments in simulation and real-world settings, showing that FALH consistently outperforms ConceptGraphs [1] and HOV-SG [2]. It also achieves reliable performance when deployed on a physical robot, with success rates remaining comparable across static and dynamic scenes.

II. RELATED WORK

A. Open-vocabulary perception for mapping

Closed-set detectors [15], [16] have enabled many 3D semantic mapping systems [25], [27]–[29], offering strong performance in terms of speed and accuracy. However, their

Authorized licensed use limited to: Peking University. Downloaded on October 21,2026 at 04:06:18 UTC from IEEE Xplore. Restrictions apply.

reliance on fixed label sets limits their ability to handle open-ended language instructions. To improve coverage, recent work [30]–[36] combines open-vocabulary detection [37]–[41] or segmentation [42]–[48] with multi-view fusion.

Most of these pipelines are prompt-driven: during mapping, they rely on a predefined word list to determine which object categories to detect and store. This design introduces several challenges. It makes performance sensitive to phrasing—synonyms, compound descriptions (e.g., color), or region-level references may not be captured. Short word lists miss uncommon categories, while longer ones increase computational cost and still leave gaps. Detection thresholds tuned for one environment may not transfer well to others, complicating multi-view fusion. Viewpoint inconsistencies can lead to duplicate or missing entries, while early false negatives eliminate visual evidence that cannot be recovered later. Expanding the label list or relaxing thresholds offers limited improvement: the resulting map remains constrained by the original prompts, and objects not included during mapping are usually omitted, even if they were visible. These limitations are particularly impactful for navigation, where missing entries during mapping prevent effective recall during query time.

Our approach eliminates prompt dependence during mapping by using class-agnostic detectors and segmenters to extract visual features directly from the scene, without relying on predefined word lists. These features are then stored together with the robot pose, which reduces vocabulary bias, avoids prompt tuning for each environment, and allows subsequent queries to flexibly retrieve information grounded in what was actually observed.

B. 3D scene representations

Prior work [25]–[29] has modeled 3D environments using three widely adopted forms. A common strategy uses 2D semantic grids [49], [50] aligned to the robot frame, where ground-plane cells encode local category or appearance cues. This supports fast updates and efficient indexing for short-horizon planning. Another line of work constructs dense 3D representations [20]–[24] such as point clouds or voxel grids, in which each element carries geometric and semantic features; several variants replace discrete grids with continuous implicit fields [51]–[53] to preserve fine structure and support volumetric reasoning. A third direction builds object-centric maps or scene graphs structured around detected entities and their spatial relations, often anchored to local metric submaps for planning and localization.

While effective, these designs come with practical trade-offs, particularly in dynamic settings. Dense 3D maps often store redundant features across nearby elements, leading to high memory usage and expensive updates as scenes evolve. Even minor changes may trigger recomputation over large areas. Object-centric representations reduce density but depend on persistent instance tracking and cross-view association; when objects move or layouts change, maintaining consistent identities becomes challenging and slow. Hybrid metric–topological systems improve expressiveness but still require complex update logic to remain consistent as new data arrives. For navigation, these overheads result in higher latency, outdated guidance, and increased engineering effort to balance memory usage with recall performance.

Our method takes a lighter approach: rather than building and maintaining a dense global map, we store a compact, robot-centric memory of visual features and associated poses. Memory entries are overwritten on revisits, and retrieval is based on feature similarity followed by local confirmation. This avoids global fusion and long-term identity tracking, keeps memory usage bounded, and allows new observations to quickly reshape search priorities. The resulting representation remains language-queryable, incrementally updated, and responsive to environmental changes.

III. METHOD

We enable human-like search by coupling a prompt-free, robot-centric scene memory with two stages—coarse localization and fine search. The system consumes RGB images and robot poses and, given a natural language (or reference image) query, returns a 3D target estimate. The memory is constructed online without prompts and updated continuously; at query time, the agent recalls likely locations from memory (coarse) and, upon arrival, verifies the target in the local view and lifts it to 3D using depth (fine). The overall workflow is shown in Fig. 2.

A. Robot-Centric Memory Map Generation

a) Prompt-free proposals and mask merging: At time t, given an RGB frame I_t and camera pose $\mathbf{p}_t$, we generate class-agnostic instance masks from two sources: (i) box proposals from a prompt-free detector (e.g., MAVL [54]) refined by a box-guided segmenter (e.g., EfficientViT-SAM [55]), producing $\mathcal{M}_{\text{box}}$; and (ii) full-image proposals from a fast class-agnostic segmenter (FastSAM [56]), producing $\mathcal{M}_{\text{full}}$. We then merge the two using a simple priority rule—box-driven masks take precedence, and full-image masks are used to complete missing extents:

$$\mathcal{M}_t = \text{Merge}^{\text{box} \triangleright \text{full}}\big(\mathcal{M}_{\text{box}}, \mathcal{M}_{\text{full}}\big) \tag{1}$$

Duplicate proposals are suppressed using intersection-over-union (IoU) together with a brief boundary-overlap check computed on one-pixel dilations.

b) Language-aligned visual features extraction and memory entries generation: For each mask $m \in \mathcal{M}_t$, we crop its tight region $c(m)$ and encode it with a vision encoder ϕ_I to obtain a feature vector $\mathbf{f} = \phi_I\big(c(m)\big) \in \mathbb{R}^d$, where d denotes the feature dimension. The frame then contributes a new memory entry

$$\text{Mem}_t = \big(\mathcal{F}_t, \mathbf{p}_t\big), \qquad \mathcal{F}_t = \{\mathbf{f}(m) \mid m \in \mathcal{M}_t\}, \tag{2}$$

which stores the set of features observed at pose $\mathbf{p}_t$.

c) Grid indexing and overwrite policy: We discretize the ground plane into cells $\mathcal{G} = \{g\}$ and deterministically map each frame to a cell via $g = \Gamma(\mathbf{p}_t)$ (e.g., by binning (x, y) in the robot frame). Each cell g stores a feature set $\mathcal{F}_g$ and a representative pose $\mathbf{p}_g$. Upon revisiting, we

Authorized licensed use limited to: Peking University. Downloaded on October 21,2026 at 04:06:18 UTC from IEEE Xplore. Restrictions apply.

overwrite the state: $\mathcal{F}_g \leftarrow \mathcal{F}_t$ and $\mathbf{p}_g \leftarrow \mathbf{p}_t$. This deliberate choice ensures real-time adaptability and extreme memory efficiency ($<$ 40 MB per 20 m^2), as it prioritizes the most recent scene configuration in dynamic environments over potentially outdated historical data. Information loss is later mitigated by the local verification in Section III-C. Perception updates run continuously in a dedicated thread.

B. Coarse Localization

We support two query modes and treat them uniformly. For a text query q_{text}, a text encoder [9], [12] ϕ_T produces the query embedding $\mathbf{t} = \phi_T(q_{\text{text}})$. For a reference image query with crop c_{ref}, a vision encoder [9], [13], [14] ϕ_I similarly yields $\mathbf{t} = \phi_I(c_{\text{ref}})$.

For each populated cell i with stored feature set $\mathcal{F}_i$ and representative pose $\mathbf{p}_i$, we compute a score as the best text–image match (inner product) among its features:

$$s_i = \max_{\mathbf{f} \in \mathcal{F}_i} \langle \mathbf{f}, \mathbf{t} \rangle. \tag{3}$$

We select the top K cells as candidates and use their poses $\{\mathbf{p}_{i_1}, \ldots, \mathbf{p}_{i_K}\}$ as candidate locations. The visit order balances match quality and travel effort:

$$\pi^\star = \arg\min_{\pi} \sum_{k} \Big(\lambda \, \mathrm{Dist}\big(\mathbf{p}_{\pi(k-1)}, \mathbf{p}_{\pi(k)}\big) \; - \; s_{\pi(k)} \Big), \tag{4}$$

where $\lambda > 0$ is a trade-off weight, $\mathrm{Dist}(\cdot,\cdot)$ denotes the planar distance, and $\mathbf{p}_{\pi(0)}$ is the current pose of the robot.

As the robot moves, the perception thread keeps overwriting revisited cells and recomputing the scores $\{s_i\}$, so both the candidate set and the planned order are refreshed online for text and reference image queries alike.

C. Fine Search

Upon reaching a candidate location, the agent performs a fine search based on its current view. Let $\mathbf{t}$ denote the query feature: for a language query q_{text}, we set $\mathbf{t} = \phi_T(q_{\text{text}})$; for a reference image query with crop c_{ref}, we set $\mathbf{t} = \phi_I(c_{\text{ref}})$.

Language queries. An open-vocabulary detector first proposes bounding boxes. If any box has sufficient confidence, a box-guided segmenter refines it into an instance mask. Otherwise, we fall back to the prompt-free segmentation pathway used during memory construction, producing a small set of candidate masks $\{m_r\}$.

reference image queries. We skip the detector and directly use the prompt-free segmentation pathway to obtain candidate masks $\{m_r\}$ from the current view.

For each candidate mask m_r, we extract its crop and encode it using the vision encoder to obtain a feature vector $\mathbf{f}_r = \phi_I\big(c(m_r)\big)$. We then compute its similarity to the query as $\hat{s}_r = \langle \mathbf{f}_r, \mathbf{t} \rangle$. Let $r^\star = \arg\max_r \hat{s}_r$ be the index of the best-matching mask. If $\hat{s}_{r^\star} \geq \tau$ (a fixed threshold), we consider the object matched; otherwise, the agent proceeds to the next candidate location.

Once a match is confirmed, we lift the accepted mask to 3D using the depth map. Given camera intrinsics $\mathbf{K}$ and per-pixel depth D, each pixel $u = (u_x, u_y)$ inside $m_{r^\star}$ is projected to the camera coordinates as

$$\mathbf{P}(u) = D(u)\, \mathbf{K}^{-1} \, [u_x, \, u_y, \, 1]^T. \tag{5}$$

Let $\mathcal{P} = \{\mathbf{P}(u) \mid u \in m_{r^\star}\}$ denote the resulting 3D point set, which serves directly as input to downstream navigation and manipulation modules.

TABLE II
SUCCESS RATE (SR) ON SELECTED THREE HM3D SCENES (00802, 00812, 00818).

Method		00802	00812	00818	Avg. SR
ConceptGraphs [1]	ICRA'24	66.3%	54.2%	68.7%	63.1%
HOV-SG [2]	ICRA'24	61.4%	48.2%	55.4%	55.0%
Ours		**72.3%**	**69.9%**	**78.3%**	**73.5%**

IV. EXPERIMENTS

We evaluate FALH in simulated environments and in real-world everyday scenes. The study has three parts. First, we evaluate *object navigation in simulation* (§IV-B) on HM3D [57] and AI2-THOR [58] and analyze how retrieval from the scene memory and fine search affect performance; for AI2-THOR, we report results on static and dynamic scenes using the same planner and settings. Second, we present *real-world deployments* (§IV-C) to evaluate robustness outside simulation. Third, we examine *modular combinations for class-agnostic segmentation and retrieval* (§IV-D) by comparing twelve detector–segmenter–encoder pipelines and summarizing their overall performance; this selection yields the default configuration used in later experiments.

Unless otherwise noted, all navigation results use the configuration identified in the modular study and follow a common evaluation setup across methods and scenes.

A. Datasets, Metrics, and Implementation

a) Datasets: We evaluate FALH on three tasks: **(1) class-agnostic segmentation**, **(2) language-guided mask retrieval**, and **(3) object navigation**. For (1) and (2), we use ScanNet and an AI2-THOR dataset we collected, consisting of synchronized RGB-D sequences with odometry along controlled trajectories. For static object navigation, we evaluate on HM3D in Habitat [59], selecting three representative scenes (00802, 00812, 00818), and on AI2-THOR using three scenes (FloorPlan1, FloorPlan201, FloorPlan303). For dynamic object navigation, we construct dynamic environments in AI2-THOR using a custom tool that enables object-level relocations, and collect a corresponding dataset for evaluation. For real-world evaluation shown in Fig. 3, we use a wheeled platform equipped with an Intel RealSense D435i RGB-D camera and a MID360 LiDAR; LiDAR measurements are processed by FAST-LIO2 [60] for pose estimation.

b) Metrics: For *class-agnostic segmentation*, we report the average precision at 50% IoU (AP@50). For *language-guided mask retrieval*, we report Top-1 and Top-3 retrieval accuracy—whether the ground-truth mask appears among the top-ranked predictions for each query. For *efficiency*,

Authorized licensed use limited to: Peking University. Downloaded on October 21,2026 at 04:06:18 UTC from IEEE Xplore. Restrictions apply.

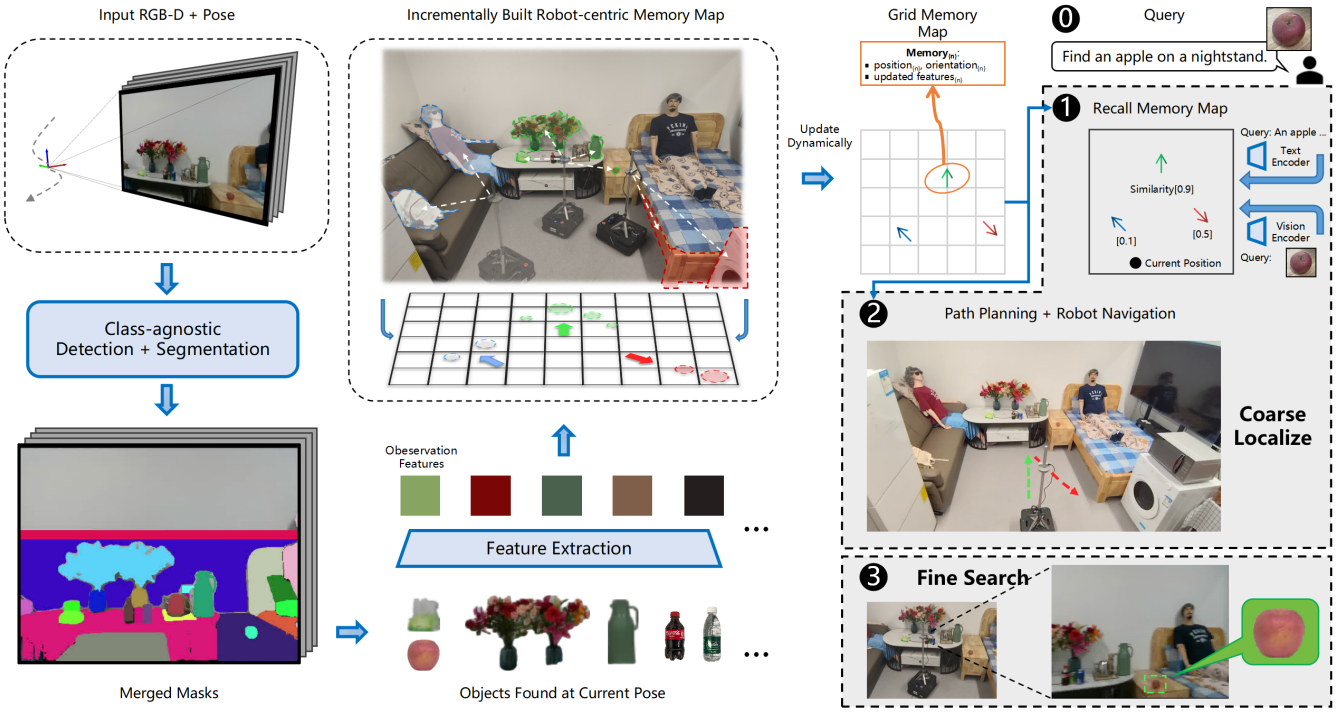


Fig. 2. **Overview of FALH.** Given the current RGB-D frame and robot pose, FALH applies class-agnostic detection and segmentation to generate instance masks. Tight crops are embedded by a vision encoder into language-aligned feature vectors and are stored in the incrementally built robot-centric scene memory together with the corresponding pose. When a language (or image) query is issued, the query is encoded and *coarse localization* retrieves top candidates from memory; their poses serve as waypoints for navigation. Upon reaching a candidate location, a *fine search* inspects the local view and, using depth, returns a precise 3D goal. The memory map is updated continuously as the robot moves, keeping guidance responsive to scene changes.



Fig. 3. **Real-world object navigation using FALH**. The robot begins by pre-exploring the environment to build a scene memory, associating visual features with observed locations. At query time, it matches the input (language or image) to memory entries and conducts a coarse-to-fine search to identify and localize the target. When the environment changes, the memory is incrementally updated to stay consistent with the real world.

TABLE III
SUCCESS RATE (SR) ON THREE AI2-THOR SCENES (FLOORPLAN1, FLOORPLAN201, FLOORPLAN303).

Scene Type	FloorPlan1	FloorPlan201	FloorPlan303	Avg. SR
Static	89.0%	80.2%	76.9%	82.1%
Dynamic	82.4%	75.8%	73.6%	77.3%

TABLE IV
SUCCESS RATE (SR) IN REAL-WORLD SCENES.

Scene Type	Static		Dynamic	
	Queries	SR	Queries	SR
Apartment	24	79.2%	18	72.2%
Corridor	18	77.8%	12	75.0%
Office	27	85.2%	21	76.2%

we measure the inference throughput in frames per second (FPS). For *navigation*, we report success rate (SR), defined as the fraction of queries whose returned goal lies within a fixed tolerance of the annotated target.

c) Implementation Details: All experiments are conducted on a workstation with an NVIDIA RTX 2080 Ti GPU and an Intel Core i7-14700KF CPU. The full system is implemented with ROS. The method has modest compute and memory demands (uses <16 GB of system memory). In a $\sim 20\,\mathrm{m}^2$ room, the constructed memory map requires less than 40 MB of storage.

B. Object Navigation in Simulation

HM3D (static scenes): We compare FALH with ConceptGraphs and HOV-SG under identical planners and evaluation settings. Experiments are conducted on three HM3D scenes (00802, 00812, 00818). As shown in Table II, FALH achieves the highest overall success rate (Avg SR 73.5%), outperforming ConceptGraphs (63.1%) and HOV-

Authorized licensed use limited to: Peking University. Downloaded on October 21,2026 at 04:06:18 UTC from IEEE Xplore. Restrictions apply.

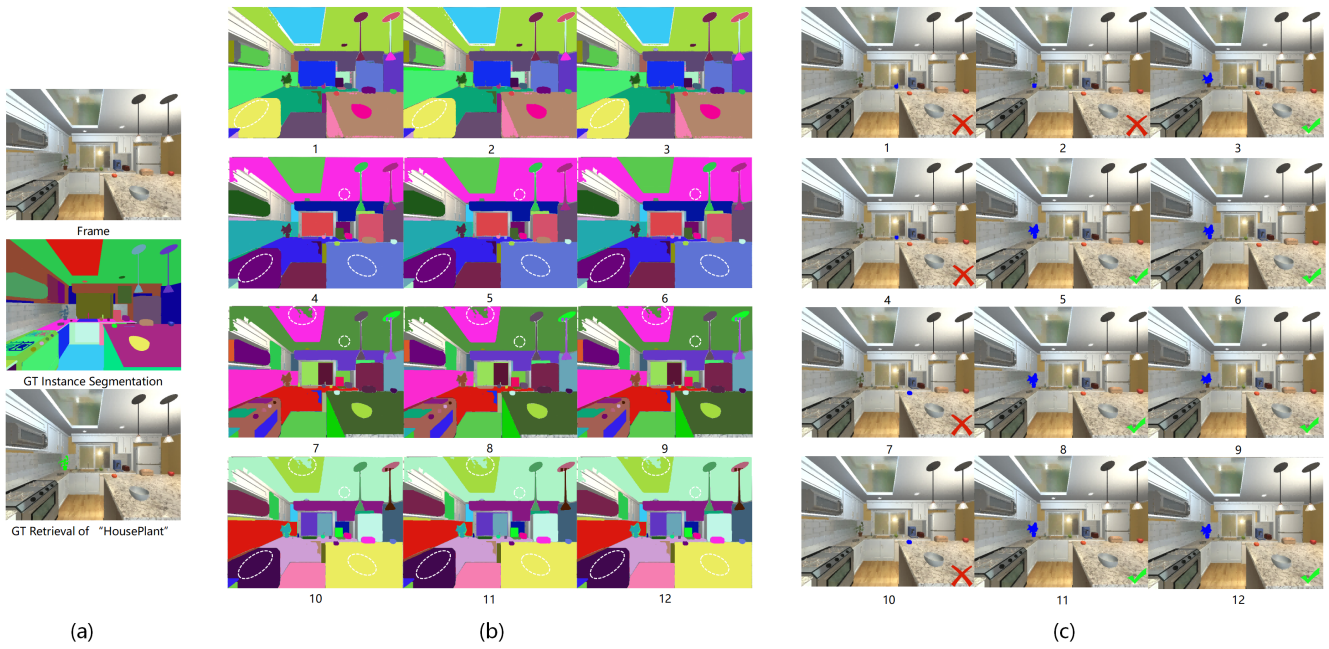


Fig. 4. **Evaluation of Modular Combinations for Class-Agnostic Segmentation and Retrieval.** (a) Input frame, ground-truth (GT) instance segmentation, and GT retrieval for the text query "HousePlant". (b) Instance masks from 12 detector–segmenter–visual-encoder combinations (IDs 1–12; see Table V, column "Panel No."). White dashed ellipses mark regions where the target is missed or fragmented. (c) Retrieval results for IDs 1–12: ✓ denotes a correct top-1 retrieval (target present in the retrieved view), and × denotes an incorrect result. Panel numbers in (c) match those in (b).

TABLE V

EVALUATION OF MODULAR COMBINATIONS FOR CLASS-AGNOSTIC SEGMENTATION AND RETRIEVAL

Detector	Segmenter	Visual Encoder	Panel No.	AP@50	Top-1	Top-3	FPS
MAVL [54]	EfficientViT-SAM [55]	CLIP [9]	1	0.4201	0.3134	0.5382	4.49
MAVL [54]	EfficientViT-SAM [55]	DINOv2 [13]	2	0.4201	0.3805	0.6069	2.92
MAVL [54]	**EfficientViT-SAM [55]**	**DINOv3 [14]**	**3**	**0.4201**	**0.3881**	**0.6246**	**4.58**
MAVL [54]	FastSAM [56]	CLIP [9]	4	0.4195	0.3149	0.5254	4.47
MAVL [54]	FastSAM [56]	DINOv2 [13]	5	0.4195	0.3732	0.6086	3.37
MAVL [54]	FastSAM [56]	DINOv3 [14]	6	0.4195	0.3797	0.6026	4.67
RAM [61] + YOLO-World [40]	EfficientViT-SAM [55]	CLIP [9]	7	0.4590	0.3156	0.5148	4.68
RAM [61] + YOLO-World [40]	EfficientViT-SAM [55]	DINOv2 [13]	8	0.4590	0.3658	0.5956	2.94
RAM [61] + YOLO-World [40]	EfficientViT-SAM [55]	DINOv3 [14]	9	0.4590	0.3752	0.6096	4.60
RAM [61] + YOLO-World [40]	FastSAM [56]	CLIP [9]	10	0.4382	0.3028	0.5303	4.08
RAM [61] + YOLO-World [40]	FastSAM [56]	DINOv2 [13]	11	0.4382	0.3516	0.6034	3.55
RAM [61] + YOLO-World [40]	FastSAM [56]	DINOv3 [14]	12	0.4382	0.3713	0.5849	4.43

SG (55.0%). Both baselines apply inter-object merging to limit segment counts, but still tend to produce redundant fragments, which can introduce ambiguity during retrieval and compromise overall navigation performance.

AI2-THOR (static and dynamic scenes): We evaluate language-guided object navigation in AI2-THOR using three diverse layouts (FloorPlan1, FloorPlan201, FloorPlan303). All methods are tested with the same planner and hyperparameters. We report both per-scene and mean success rate (SR) under two conditions: (i) static scenes, and (ii) dynamic scenes, where a subset of objects is moved after exploration and the same queries is reissued.

Static scenes. Table III reports the per-scene SR and the mean SR across the three layouts. FALH achieves consistently high SR by first recalling likely locations from the scene memory and then verifying the target with a local fine search. The remaining failures are mainly due to visually similar distractors or very small targets whose apparent size makes reliable detection challenging.

Dynamic scenes. After exploration, a subset of objects is relocated and the same language queries are reissued. As the robot navigates, the memory is updated online and candidate rankings are revised, sustaining strong SR under object motion (Table III). Remaining failures primarily involve large relocations combined with limited visibility or ambiguous descriptions; brief local exploratory sweeps before fine search mitigate many of these cases.

Authorized licensed use limited to: Peking University. Downloaded on October 21,2026 at 04:06:18 UTC from IEEE Xplore. Restrictions apply.

C. Object Navigation in Real-World Environments

We evaluate FALH on a wheeled mobile robot across several everyday indoor environments. Each scene is populated with a variety of household objects (e.g., clothes, bags) to create diverse navigation targets. During the mapping phase, the robot is manually teleoperated to explore the space and build its scene memory. We then evaluate static object navigation using the same inference pipeline as in simulation. For dynamic trials, a subset of objects is repositioned, and the robot is reissued queries corresponding to the moved items. As reported in Table IV, the results are consistent with simulation: the robot-centric memory effectively retrieves candidate regions, and local fine search around these areas yields accurate 3D goals. When object layouts change, the system maintains robust guidance by incrementally updating memory and refreshing the coarse location stage.

D. Evaluation of Modular Combinations for Class-Agnostic Segmentation and Retrieval

To support open-vocabulary object navigation, we benchmark twelve modular combinations for instance-level segmentation and retrieval. Each pipeline performs dense segmentation over the scene and encodes all visible instances for matching against language queries. The combinations span different detectors, segmenters, and visual encoders, and are evaluated on AP@50, retrieval accuracy (Top-1 / Top-3), and runtime (FPS). Representative outputs are shown in Fig. 4, and detailed results are reported in Table V.

Among all variants, **MAVL + EfficientViT-SAM + DINOv3** achieves the best overall performance, with the highest retrieval scores (Top-1: 0.3881, Top-3: 0.6246) and real-time speed (4.58 FPS). This configuration is adopted in all object navigation experiments described in earlier sections. Other combinations exhibit expected trade-offs: DINOv3 consistently outperforms CLIP for instance retrieval, while pipelines using RAM + YOLO-World tend to show lower precision across metrics.

V. CONCLUSIONS

We introduced FALH, a framework for natural language navigation, inspired by how humans search—recalling likely regions from memory and verifying them up close. Central to our approach is a robot-centric memory that records visual observations along with their associated robot poses, allowing the system to retain what was seen and from where. This structure enables efficient localization and robust adaptation to dynamic, cluttered environments. The memory is constructed online using class-agnostic perception, without relying on predefined prompts or fixed vocabularies during mapping. Object navigation experiments in both simulated and real-world settings demonstrate that FALH outperforms object-centric baselines while maintaining low computational overhead. Nonetheless, the overwrite policy involves a trade-off between efficiency and long-term context. We mitigate this by coupling coarse retrieval with local verification, ensuring robustness despite historical data loss. Future work will explore temporally structured memory, uncertainty-aware updates, and multi-turn interactions to support more persistent and generalizable object search.

REFERENCES

[1] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[2] 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 IEEE International Conference on Robotics and Automation(ICRA)*, 2024.

[3] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3d scene graphs," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8921–8928, 2024.

[4] Y. Tang, M. Wang, Y. Deng, Z. Zheng, J. Deng, S. Zuo, and Y. Yue, "Openin: Open-vocabulary instance-oriented navigation in dynamic domestic environments," *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 9256–9263, 2025.

[5] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, "Openscene: 3d scene understanding with open vocabularies," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2023.

[6] R. Ding, J. Yang, C. Xue, W. Zhang, S. Bai, and X. Qi, "Lowis3d: Language-driven open-world instance-level 3d scene understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 8517–8533, 2024.

[7] R. Ding, J. Yang, C. Xue, W. Zhang, S. Bai, and X. Qi, "Pla: Language-driven open-vocabulary 3d scene understanding," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2023.

[8] K. Mazur, E. Sucar, and A. J. Davison, "Feature-realistic neural fusion for real-time, open set scene understanding," in *IEEE International Conference on Robotics and Automation(ICRA)*, 2023.

[9] 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(ICML)*, 2021.

[10] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *IEEE/CVF international conference on computer vision(ICCV)*, 2023.

[11] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li, *et al.*, "Regionclip: Region-based language-image pretraining," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2022.

[12] P. K. A. Vasu, H. Pouransari, F. Faghri, R. Vemulapalli, and O. Tuzel, "Mobileclip: Fast image-text models through multi-modal reinforced training," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2024.

[13] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv:2304.07193*, 2023.

[14] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, *et al.*, "Dinov3," *arXiv:2508.10104*, 2025.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE/CVF conference on computer vision and pattern recognition(CVPR)*, 2016.

[17] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[18] T. Laidlow, M. Bloesch, W. Li, and S. Leutenegger, "Dense rgb-d-inertial slam with map deformations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2017.

Authorized licensed use limited to: Peking University. Downloaded on October 21,2026 at 04:06:18 UTC from IEEE Xplore. Restrictions apply.

[19] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2019.

[20] M. M. Johari, C. Carta, and F. Fleuret, "Eslam: Efficient dense slam system based on hybrid representation of signed distance fields," in *IEEE/CVF conference on computer vision and pattern recognition(CVPR)*, 2023.

[21] X. Zuo, N. Yang, N. Merrill, B. Xu, and S. Leutenegger, "Incremental dense reconstruction from monocular video with guided sparse feature volume fusion," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3876–3883, 2023.

[22] Y. Xin, X. Zuo, D. Lu, and S. Leutenegger, "Simplemapping: Real-time visual-inertial dense mapping with deep multi-view stereo," *IEEE International Symposium on Mixed and Augmented Reality(ISMAR)*, 2023.

[23] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting slam," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2024.

[24] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2024.

[25] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," in *arXiv:2201.13360*, 2022.

[26] L. Schmid, M. Abate, Y. Chang, and L. Carlone, "Khronos: A unified approach for spatio-temporal metric-semantic slam in dynamic environments," in *arXiv:2402.13817*, 2024.

[27] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[28] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *International Conference on 3D Vision (3DV)*, 2018.

[29] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[30] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, A. Maalouf, S. Li, G. Iyer, S. Saryazdi, N. Keetha, *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," in *arXiv:2302.07241*, 2023.

[31] K. Yamazaki, T. Hanyu, K. Vo, T. Pham, M. Tran, G. Doretto, A. Nguyen, and N. Le, "Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[32] M. Tie, J. Wei, K. Wu, Z. Wang, S. Yuan, K. Zhang, J. Jia, J. Zhao, Z. Gan, and W. Ding, "O2v-mapping: Online open-vocabulary mapping with neural implicit representation," in *European Conference on Computer Vision(ECCV)*, 2024.

[33] S. Koch, N. Vaskevicius, M. Colosi, P. Hermosilla, and T. Ropinski, "Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2024.

[34] Y. Mehan, K. Gupta, R. Jayanti, A. Govil, S. Garg, and M. Krishna, "Questmaps: Queryable semantic topological maps for 3d scene understanding," in *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2024.

[35] S. Garg, K. Rana, M. Hosseinzadeh, L. Mares, N. Sünderhauf, F. Dayoub, and I. Reid, "Robohop: Segment-based topological map representation for open-world visual navigation," in *IEEE International Conference on Robotics and Automation(ICRA)*, 2024.

[36] Y. Deng, J. Wang, J. Zhao, X. Tian, G. Chen, Y. Yang, and Y. Yue, "Opengraph: Open-vocabulary hierarchical 3d graph representation in large-scale outdoor environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8402–8409, 2024.

[37] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *European Conference on Computer Vision(ECCV)*, 2022.

[38] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, *et al.*, "Grounded language-image pre-training," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2022.

[39] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation," in *arXiv:2104.13921*, 2021.

[40] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2024.

[41] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, and otherss, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European Conference on Computer Vision(ECCV)*, 2024.

[42] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann, "Openmask3d: Open-vocabulary 3d instance segmentation," in *arXiv:2306.13631*, 2023.

[43] K. Liu, F. Zhan, J. Zhang, M. Xu, Y. Yu, A. El Saddik, C. Theobalt, E. Xing, and S. Lu, "Weakly supervised 3d open-vocabulary segmentation," *Advances in Neural Information Processing Systems(NeurIPS)*, 2023.

[44] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," in *arXiv:2201.03546*, 2022.

[45] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling open-vocabulary image segmentation with image-level labels," in *European Conference on Computer Vision(ECCV)*, 2022.

[46] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu, "Open-vocabulary semantic segmentation with mask-adapted clip," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2023.

[47] H. Zhang, F. Li, X. Zou, S. Liu, C. Li, J. Yang, and L. Zhang, "A simple framework for open-vocabulary segmentation and detection," in *IEEE/CVF International Conference on Computer Vision(ICCV)*, 2023.

[48] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, and X. Wang, "Groupvit: Semantic segmentation emerges from text supervision," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2022.

[49] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," *International Symposium on Experimental Robotics(ISER)*, 2023.

[50] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *International Conference on Robot Learning(CoRL)*, 2023.

[51] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," in *arXiv:2210.05663*, 2022.

[52] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," in *Advances in Neural Information Processing Systems(NeurIPS)*, 2022.

[53] J. Zhang, R. Dong, and K. Ma, "Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip," *IEEE/CVF International Conference on Computer Vision(ICCV)*, 2023.

[54] M. Maaz, H. Rasheed, S. Khan, F. S. Khan, R. M. Anwer, and M.-H. Yang, "Class-agnostic object detection with multi-modal transformer," in *European conference on computer vision(ECCV)*, 2022.

[55] Z. Zhang, H. Cai, and S. Han, "Efficientvit-sam: Accelerated segment anything model without performance loss," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR*, 2024.

[56] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv:2306.12156*, 2023.

[57] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva, *et al.*, "Habitat-matterport 3d semantics dataset," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2023.

[58] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, *et al.*, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv:1712.05474*, 2017.

[59] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min, *et al.*, "Habitat 3.0: A co-habitat for humans, avatars and robots," *arXiv:2310.13724*, 2023.

[60] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[61] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu, *et al.*, "Recognize anything: A strong image tagging model," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2024.

Authorized licensed use limited to: Peking University. Downloaded on October 21,2026 at 04:06:18 UTC from IEEE Xplore. Restrictions apply.