DualMap: Online Open-Vocabulary Semantic Mapping for Natural Language Navigation in Dynamic Changing Scenes

Jiajun Jiang[1], Yiming Zhu[1], Zirui Wu[1], Jie Song[1,2]

***Abstract*— Navigating dynamic, real-world environments using natural language commands requires a system to be open-vocabulary, computationally efficient, and robust to object relocation. We present DualMap, an online open-vocabulary mapping system for language-guided navigation in dynamic scenes. DualMap features a hybrid segmentation frontend and lightweight intra-object checks, enabling the construction of high-quality semantic maps while avoiding costly 3D merging. The core of our system is a novel dual-map representation. It combines a global *abstract* map for high-level candidate selection with a local *concrete* map for precise goal-reaching. This structure is crucial for handling dynamic changes, as it allows for efficient online updates when an object is moved from its expected location. Extensive experiments in both simulation and the real world demonstrate state-of-the-art performance in mapping efficiency and navigation success in static and dynamic scenes. Project page: https://eku127.github.io/DualMap/**

I. Introduction

Imagine asking a home-assistant robot to "find the cracker box", a simple request in a household environment. The robot might first navigate to the kitchen counter where the box was last seen, only to find it missing. To succeed, the robot must realize the object has moved, and adapt its search accordingly. This common scenario highlights three critical challenges for robotic systems: 1) *Open-vocabulary* understanding, to interpret natural language queries for arbitrary objects; 2) *Efficient online mapping*, to incrementally build and maintain semantic maps in real-time; and 3) *Navigation with dynamic changes*, to adapt to objects that frequently move in human-centric environments.

Existing approaches struggle to address all three challenges simultaneously. While efficient online semantic mapping systems [1], [2] operate in real-time, they are built upon closed-set detectors and thus cannot handle open-ended natural language queries. Recent efforts have integrated vision foundation models to enable open-vocabulary mapping [3]–[5], but these methods fundamentally assume a static environment. Conversely, other open-vocabulary systems that do tackle dynamic changes [6], [7] typically require significant offline processing time to construct a map, leaving them impractical for real-time, lifelong navigation tasks.

In this work, we present **DualMap**, an online open-vocabulary map representation designed to meet all three aforementioned requirements. To achieve efficient mapping, our system first constructs a high-quality 3D semantic *concrete map* through two key innovations. A hybrid segmentation frontend provides fast, open-vocabulary object detection, while lightweight intra-object status checks enhance map fidelity by removing noise and correcting segmentation errors. Crucially, these designs eliminate the need for costly 3D inter-object merging common in prior works [4], [5].

To support robust navigation in dynamic environments, this fine-grained concrete map is then converted into a lightweight *abstract map* composed of typically static anchor objects and scene layout. This abstraction is based on the insight that global structural cues are sufficient for high-level planning, while precise object details can be retrieved locally via online perception. Our dual-map navigation strategy leverages this: it uses the global abstract map for initial candidate selection and the local concrete map for accurate localization. This enables efficient re-planning for navigation when a queried object has been moved, as the abstract map is continuously updated online using new observations gathered during navigation.

II. Methods

A. Online Concrete Map Construction

The concrete map $\mathcal{M}_c$ is a collection of all object instances observed in the environment, built efficiently without costly 3D processing. This efficiency is achieved through two key designs: a hybrid segmentation frontend and lightweight intra-object checks that replace the need for expensive 3D inter-object merging used in prior works.

a) Hybrid Open-Vocabulary Segmentation: For each RGBD frame, we use YOLO [8] and MobileSAM [9] to rapidly obtain object detections and their corresponding masks. YOLO's predefined category list is generated once at startup by prompting a large language model [10] with the robot's working context. In parallel, we run the open-set model FastSAM [11] to segment objects beyond these predefined categories to ensure open-vocabulary ability. This hybrid strategy achieves open-vocabulary, comprehensive object coverage in an online manner. Each segment is further arranged as an observation, which is represented by its 3D point cloud, class ID, and a fused semantic CLIP feature [12] derived from both visual and textual information for rich language grounding.

b) Object Association and Status Checks: New observations are associated with existing objects in $\mathcal{M}_c$ based on geometric and semantic similarity. We then perform lightweight intra-object status checks—namely *stability check* and *split detection*. Stability check filters out noisy or insufficiently observed objects. This check is triggered for any object that has not been updated for a prolonged period. An object passes if it is both sufficiently observed

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology

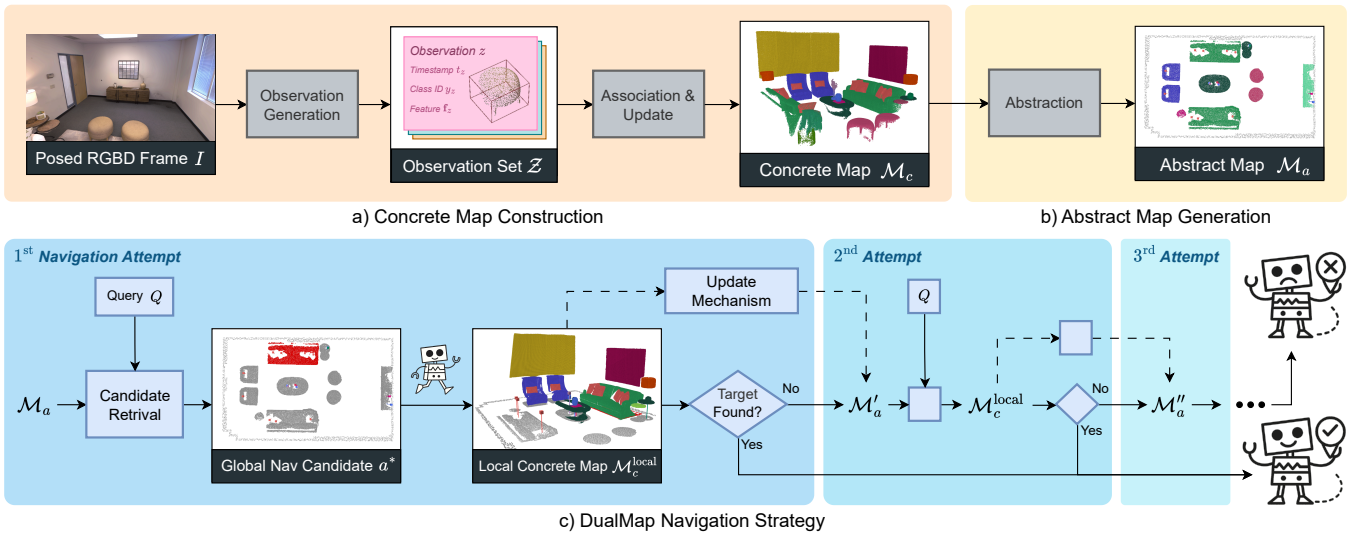


Fig. 1: DualMap system overview: a) A detailed 3D semantic concrete map $\mathcal{M}_c$ is built from online observations of posed RGBD frames; b) An anchor-based abstract map $\mathcal{M}_a$ is derived from $\mathcal{M}_c$, retaining global layout and static objects; c) Given a natural language query Q, the agent retrieves a global candidate a^* from $\mathcal{M}_a$ and starts navigation. During execution, it incrementally builds a local concrete map $\mathcal{M}_c^{\text{local}}$, checks for target object presence, and updates the abstract map $\mathcal{M}_a$ accordingly. If the target is not found near the a^*, a new navigation attempt is made using the updated map $\mathcal{M}_a'$. This loop continues until the target is found or the attempt limit is reached.

and has a dominant class ID (accounting for over two-thirds of its observation history); otherwise, it is pruned. To correct under-segmentation errors where adjacent items are merged, we use a split operation. This is triggered when an object receives observations with different class IDs at the same timestamp across frames. The object is then partitioned into new instances by class ID, preserving detail and improving scene fidelity. These status checks maintains high map fidelity without expensive 3D inter-object merging.

B. Abstract Map and Navigation Strategy

The abstract map $\mathcal{M}_a$ provides a simplified, stable scene representation for efficient long-range planning and robust navigation failure handling.

a) Map Abstraction: We first classify the objects from $\mathcal{M}_c$ into static **anchor objects** (e.g., desks, beds) and movable **volatile objects** (e.g., cups, books). We classify each object $o \in \mathcal{M}_c$ using a two-step process. First, we compare its CLIP feature similarity to predefined anchor and volatile category lists. If the similarity to one list exceeds the other by a margin of $\Delta\tau = 0.05$, the object is classified accordingly. In ambiguous cases, we then compare the object's feature $\mathbf{f}_o$ to a generic anchor template feature $\mathbf{f}_t$, encoded from descriptive phrases (e.g., "furniture that is not often moved"). The object is classified as an anchor a only if this final similarity exceeds a threshold τ_a; otherwise, it is deemed volatile v.

For the abstract map $\mathcal{M}_a$, we retain the geometry of anchor objects for global planning while abstracting volatile objects to their semantic features. These features are then associated with a supporting anchor if a spatial "on" relation is detected. This relation is established if the volatile object's 2D projection falls within the anchor's footprint and its base is vertically proximate (within $\delta = 0.1\,\text{m}$) to the anchor's primary supporting plane, which is derived from the anchor's point cloud Z-axis histogram. This abstraction design allows us to maintain a compact yet informative map of the scene's stable structure for high-level navigation.

b) Navigation with Online Updates: Our navigation strategy leverages this dual representation to robustly handle dynamic object changes. The process, illustrated in Fig. 1-c, unfolds as follows:

1) **Candidate Retrieval:** Given a language query Q, the system retrieves the most relevant anchor candidate a^* from the global abstract map $\mathcal{M}_a$ via similarity calculation. The selected anchor a^* suggests that the queried object is most likely situated nearby. Both the anchor a^* and its similarity score $s(a^*)$ are used for further navigation.
2) **Local Concrete Mapping:** A global path toward a^* is planned using a Voronoi-based planner over the abstract map [13]. During the process of navigation towards a^*, the system incrementally building a local, up-to-date concrete map, $\mathcal{M}_c^{\text{local}}$, of its current surroundings. For each object $o \in \mathcal{M}_c^{\text{local}}$, we compute the cosine similarity $s(o)$ between its feature and the query feature. If $s(o)$ is within a margin ϵ of $s(a^*)$ and o lies within the projected footprint of a^*, it is considered a confident match. A local path is then planned via RRT* [14] to reach the target.
3) **Re-planning & Map Update:** If no confident match is found near a^*, suggesting the queried object may have changed location, the system re-executes the candidate retrieval over the updated abstract map $\mathcal{M}_a' \setminus \{a^*\}$ and selects a new anchor $a^{*\prime}$. Here, $\mathcal{M}_a'$ is obtained by updating the original abstract map $\mathcal{M}_a$ with the

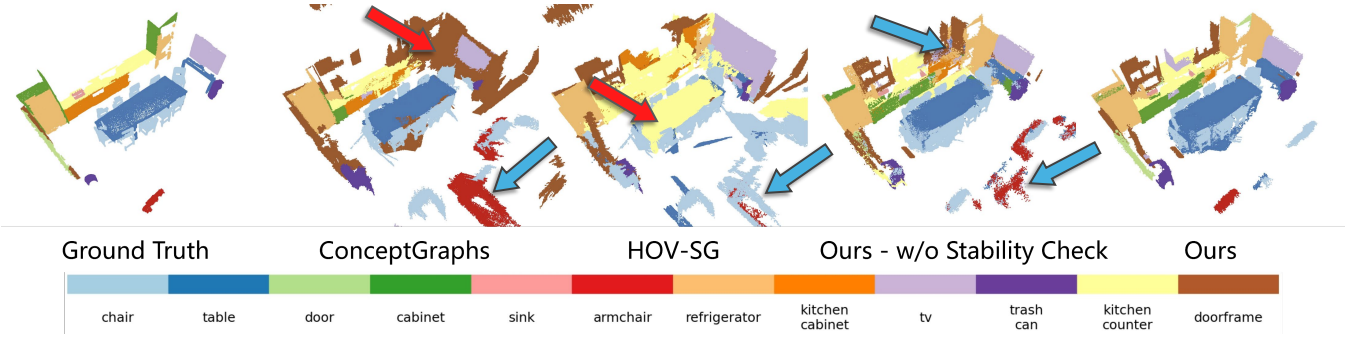


Fig. 2: Qualitative comparison of semantic segmentation results on ScanNet `scene0011_00`. Red arrows highlight semantically inaccurate predictions, while blue arrows indicate meaningless segmentations, suggesting noisy predictions.

TABLE I: Open-vocabulary 3D Semantic Segmentation and Efficiency

Dataset	Method	mIoU ↑	FmIoU ↑	mAcc ↑	ODR ≈ 1	Avg. Mem (MB) ↓	Peak Mem (MB) ↓	TPF (s) ↓
Replica	ConceptGraphs	0.1501	0.3858	0.3559	2.02	7148.9	23551.9	4.188
	HOV-SG	0.2050	0.4846	0.3835	3.81	73368.0	158126.6	42.005
	Ours	**0.2538**	**0.5207**	**0.4024**	**0.97**	**3095.2**	**4564.0**	**0.276**
ScanNet	ConceptGraphs	0.0882	0.3077	0.3538	6.97	9780.3	26155.2	6.301
	HOV-SG	0.1333	**0.3381**	0.3714	20.34	9223.0	25735.0	8.039
	Ours	**0.1604**	0.3288	**0.3794**	**2.56**	**2120.9**	**2820.2**	**0.163**

local concrete map $\mathcal{M}_c^{\text{local}}$. The agent then resumes navigation toward $a^{*\prime}$, using the original similarity score $s(a^*)$ to remain consistent with the initial query. This strategy enables the agent to leverage contextual cues encountered in the earlier navigation process, increasing the likelihood of success if the target object was partially observed along the way.

This online update loop is the key to DualMap's robustness, as it turns a navigation failure into an opportunity to improve its scene representation and successfully complete the task.

III. Experiments

A. Experimental Setup

a) Baselines and Metrics: We evaluate DualMap against two competitive open-vocabulary mapping systems: ConceptGraphs [4] and HOV-SG [5]. We assess mapping quality using standard segmentation metrics (mIoU, F-mIoU, mAcc) and efficiency (memory usage, time per frame). We also introduce the Object Density Ratio (ODR)—the ratio of predicted to ground-truth object counts—to measure how realistically the map's object density reflects the scene. Navigation capability is measured by Success Rate (SR), defined as the agent stopping within 1 meter of the queried object. In dynamic scenes, success requires finding the target within three attempts.

b) Environments and Scenarios: Our evaluation spans both simulated and real-world settings. For quantitative analysis, we use the Replica [15], ScanNet [16], and HM3D [17] datasets. Crucially, to evaluate robustness to dynamic changes, we create custom scenarios with three HM3D scenes in Habitat Simulator where objects are relocated during a task. We define two types of changes: *In-anchor relocation* (an object moves within a local region, e.g., a cup on a table) and the more challenging *Cross-anchor relocation* (an object moves between regions, e.g., from a table to a shelf). Real-world validation is performed on both wheeled and quadrupedal robots equipped with a LiDAR and an RGBD camera (Fig. 3-a).

B. Mapping Performance and Efficiency

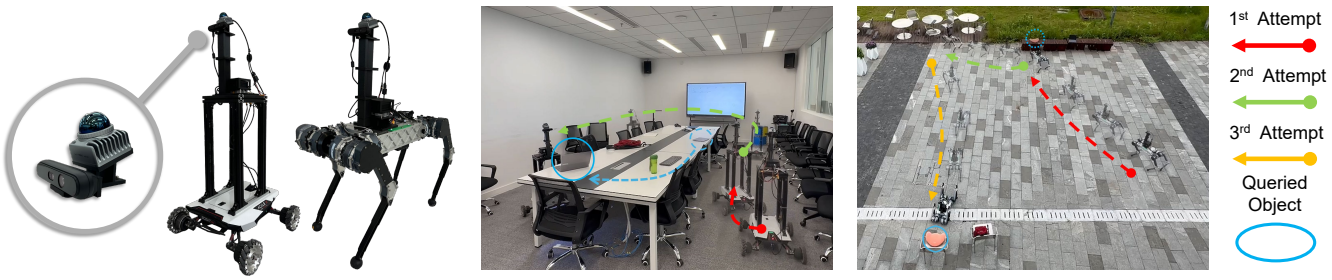
We first evaluate the quality and efficiency of our online map construction, with quantitative and qualitative results presented in Table I and Fig. 2.

a) Mapping Quality: DualMap achieves state-of-the-art semantic segmentation results across all datasets. As shown in Table I, on the Replica dataset, our method improves mIoU by **10.3%** over ConceptGraphs and **2.8%** over HOV-SG. This performance gain stems from our object-level split detection mechanism that preserves object diversity and a text-enhanced feature embedding that improves semantic understanding. The qualitative results in Fig. 2 corroborate these findings. While baselines often produce misclassified objects (red arrows) or noisy, meaningless segments from sensor noise (blue arrows), DualMap's stability check effectively filters such errors, resulting in a cleaner and more accurate scene representation.

b) Efficiency: The efficiency gains of our approach are substantial. DualMap reduces peak memory usage by over **96%** and is **99.3%** faster per frame (TPF) compared to the next best-performing method, HOV-SG. This is a direct result of our hybrid open-vocabulary detection strategy and lightweight 2D refinement, which completely avoid the costly 3D post-processing and merging steps required by prior works.

C. Navigation in Static and Dynamic Scenes

We evaluate the navigation performance in both simulated and real-world environments.



a) Wheeled and Quadruped Platforms b) **Meeting Room** – *"Find Silver Laptop"* c) **Outdoor** – *"Find Red Cushion"*

Fig. 3: Real-world navigation in dynamic environments. a) Robotic platforms equipped with a perception module that integrates a LiDAR and an RGB-D camera, both mounted on a rigid 3D-printed mount. b–c) Two examples of language-guided navigation in dynamic real-world scenes, where the agent tries to locate relocated objects across multiple attempts.

TABLE II: Object Navigation Success Rate (SR) on HM3D.

Scene Type	Method	00829	00848	00880	Trials	Avg. SR
Static	ConceptGraphs	69.2%	53.8%	61.5%		61.5%
	HOV-SG	53.8%	46.2%	57.7%	78	52.6%
	Ours	**73.1%**	**69.2%**	**69.2%**		**70.5%**
Dynamic	In-anchor (Ours)	66.7%	66.7%	61.1%	54	64.8%
	Cross-anchor (Ours)	55.6%	61.1%	64.7%	53	60.3%

TABLE III: Success Rates under Different Candidate Selection Strategies for Relocated Objects on HM3D

Strategy	Random Pick	Based on $\mathcal{M}_a$	**Based on $\mathcal{M}'_a$**
SR	13.2%	47.2%	**60.3%**

a) Simulated Environments: As shown in Table II, DualMap consistently achieves the highest Success Rate (SR) in static scenes across all HM3D environments. Its key advantage, however, is demonstrated in dynamic scenes. For *in-anchor* relocations, DualMap accurately localizes the moved object using its local concrete map. For more challenging *cross-anchor* relocations, its ability to update the abstract map online is crucial. This online update mechanism allows DualMap to maintain a high success rate even when objects undergo large positional changes.

b) Importance of Online Map Updates: To validate the effectiveness of our navigation strategy, we conducted an ablation study on the challenging *cross-anchor* task (Table III). The focus of this experiment is specifically on the robot's ability to handle positional shifts of dynamic objects, rather than structural changes to the static environment, such as the addition or removal of furniture. Simply using the original, static abstract map $\mathcal{M}_a$ for re-planning yields a success rate of only 47.2%. By using our final strategy of updating the map to $\mathcal{M}'_a$ during navigation, the success rate improves to **60.3%**. This confirms that actively using new observations to handle navigation failures is critical for success in dynamic worlds.

c) Real-World Deployment: To confirm the practical applicability of our system, we deployed DualMap on both wheeled and quadrupedal robots in four diverse real-world scenes (Fig. 3). The results, summarized in Table IV, show that DualMap achieves robust performance levels comparable to those in simulation, successfully navigating to objects in both static and dynamic scenarios.

TABLE IV: Real-World Object Navigation Results

Platform	Scene Type	Static		Dynamic	
		Trials	SR	Trials	SR
Wheeled	Meeting Room	14	85.7%	27	70.3%
	Apartment	46	69.6%	33	51.5%
Quadruped	Indoor Hallway	19	78.9%	27	55.6%
	Outdoor	12	75.0%	18	50.0%

IV. Limitations and Future Work

While DualMap demonstrates robust performance, we identify several limitations that present promising directions for future research. Primarily, the system's reliance on an external localization module [18] limits its self-sufficiency; integrating a lightweight SLAM component would create a more self-contained and easily deployable framework. Furthermore, DualMap currently lacks a model for short-term dynamics, such as moving people. A promising direction is to incorporate lightweight human representations (e.g., SMPL models) to reason about motion and human-object interactions without the overhead of full mesh reconstruction. The framework's reasoning is also constrained by its singular spatial "on" relation and the assumption of static anchors. This could be overcome by generalizing spatial relations (e.g., to "against" or "under") and treating foundational elements like floors and walls as ultimate anchors, thus handling relocatable furniture and wall-mounted objects. Finally, performance degrades in outdoor environments due to increased sensor noise and a scarcity of the "object-on-object" configurations the system relies on. Future work should therefore focus on robust sensor fusion and adapting the framework to sparse, unstructured outdoor settings.

V. Conclusion

We present DualMap, an online open-vocabulary semantic mapping system for language-guided navigation in dynamic environments. By combining a hybrid segmentation frontend with intra-object checks, it achieves efficient mapping without costly 3D merging. The dual-map design enables robust navigation through online updates and candidate reselection. Extensive experiments show that our work provides a practical and effective solution for robots operating in environments with frequent object relocations.

REFERENCES

[1] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *Proc. Robotics: Science and Systems (RSS)*, 2022.

[2] L. Schmid, M. Abate, Y. Chang, and L. Carlone, "Khronos: A Unified Approach for Spatio-Temporal Metric-Semantic SLAM in Dynamic Environments," in *Proc. Robotics: Science and Systems (RSS)*, 2024.

[3] K. M. Jatavallabhula, *et al.*, "Conceptfusion: Open-set multimodal 3d mapping," in *Proc. Robotics: Science and Systems (RSS)*, 2023.

[4] Q. Gu, *et al.*, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5021–5028.

[5] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," in *Proc. Robotics: Science and Systems (RSS)*, 2024.

[6] Y. Tang, *et al.*, "Openin: Open-vocabulary instance-oriented navigation in dynamic domestic environments," *IEEE Robotics and Automation Letters*, no. 99, pp. 1–8, 2025.

[7] Z. Yan, *et al.*, "Dynamic open-vocabulary 3d scene graphs for long-term language-guided mobile manipulation," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4252–4259, 2025.

[8] 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, pp. 16 901–16 911.

[9] C. Zhang, *et al.*, "Faster segment anything: Towards lightweight sam for mobile applications," 2023, arXiv:2306.14289.

[10] J. Achiam, *et al.*, "Gpt-4 technical report," 2023, arXiv:2303.08774.

[11] X. Zhao, *et al.*, "Fast segment anything," 2023, arXiv:2306.12156.

[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, pp. 15 963–15 974.

[13] S. Thrun and A. Bücken, "Integrating grid-based and topological maps for mobile robot navigation," in *Proceedings of the National Conference on Artificial Intelligence*, 1996, pp. 944–951.

[14] K. Sertac and F. Emilio, "Incremental sampling-based algorithms for optimal motion planning," in *Proc. Robotics: Science and Systems (RSS)*, 2010.

[15] J. Straub, *et al.*, "The replica dataset: A digital replica of indoor spaces," 2019, arXiv:1906.05797.

[16] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5828–5839.

[17] K. Yadav, *et al.*, "Habitat-matterport 3d semantics dataset," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 4927–4936.

[18] 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.