DynaMem: Online Dynamic Spatio-Semantic Memory for Open World Mobile Manipulation

Peiqi Liu[1], Zhanqiu Guo[1], Mohit Warke[1], Soumith Chintala[1,3], Chris Paxton[2],
Nur Muhammad Mahi Shafiullah*[1], Lerrel Pinto*[1]

***Abstract*— Significant progress has been made in open-vocabulary mobile manipulation, where the goal is for a robot to perform tasks in any environment given a natural language description. However, most current systems assume a static environment, which limits the system's applicability in real-world scenarios where environments frequently change due to human intervention or the robot's own actions. In this work, we present DynaMem, a new approach to open-world mobile manipulation that uses a dynamic spatio-semantic memory to represent a robot's environment. DynaMem constructs a 3D data structure to maintain a dynamic memory of point clouds, and answers open-vocabulary object localization queries using multimodal LLMs or open-vocabulary features generated by state-of-the-art vision-language models. Powered by DynaMem, our robots can explore novel environments, search for objects not found in memory, and continuously update the memory as objects move, appear, or disappear in the scene. We run extensive experiments on the Stretch SE3 robots in three real and nine offline scenes, and achieve an average pick-and-drop success rate of 70% on non-stationary objects, which is more than a 2× improvement over state-of-the-art static systems.**

I. Introduction

Recent advances in robotics have made it possible to deploy robots in real world settings to tackle the open vocabulary mobile manipulation (OVMM) problem [1]. Here, the robots are tasked with navigating in unknown environments and interacting with objects following open vocabulary language instructions, such as "Pick up **X** from **Y** and put it in **Z**", where X, Y, and Z could be any object name or location. The two most common approaches to tackling OVMM are using policies trained in simulation and deploying them in the real world [2–4], or training modular systems that combine open vocabulary navigation (OVN) [5–8] with different robot manipulation skills [9–13]. Modular systems enjoy greater efficiency and success in real-world deployment [14] as they can directly leverage advances in vision and language models [9, 12], and are able to handle more diverse and out-of-domain environments with no additional training.

However, as recent analysis has shown, the primary challenge in deploying modular OVMM is that limitations of a module propagate to the entire system [9]. One key module in any OVMM system is the open vocabulary navigation (OVN) module responsible for navigating to goals in the environment. While many such OVN systems have been proposed in the literature [1, 5–13], they share a common limitation: they assume static, unchanging environments. Contrast this with the real world, where environments change and objects are moved by either robots or humans. Making such a restrictive assumption thus limits these systems' applicability in real-world settings. The primary reason behind this assumption is the lack of an effective dynamic spatio-semantic memory that can adapt to both addition and removal of objects and obstacles in the environment online.

In this work, we propose a novel spatio-semantic memory architecture, Dynamic 3D Voxel Memory (DynaMem), that can adapt online to changes in the environment. DynaMem maintains a voxelized pointcloud representation of an environment and adds or removes points as it observes the environment change. Additionally, it supports two different ways to query the memory with natural language: a vision-language model (VLM) featurized pointcloud, and a multimodal-LLM (mLLM) QA system. Finally, DynaMem enables efficient exploration in changing environments by offering a dynamic obstacle map and a value-based exploration map that can guide the robot to explore unseen, outdated, or query-relevant parts of the world.

We evaluate DynaMem as a part of full open-vocabulary mobile manipulation stack in three real world environments with multiple rounds of changes and manipulating multiple non-stationary objects, improving the static baseline by more than 2× (70% vs. 30%). Additionally, we identify an obstacle in efficiently developing dynamic spatio-semantic memory, namely the lack of dynamic benchmarks, since many OVN systems use static simulated environments [15, 16] or static datasets [17, 18]. We address this by developing a new dynamic benchmark, DynaBench. It consists of 9 different environments, each changing over time. We ablate our design choices in this benchmark. To the best of our knowledge, DynaMem is the first spatio-semantic memory structure supporting both adding and removing objects.

II. Method

In this section, we define our problem setup, and then describe our online, dynamic spatio-semantic memory for open world, open vocabulary mobile manipulation.

A. Problem Statement

We create our algorithm, DynaMem, to solve open vocabulary mobile manipulation (OVMM) problems in an open, constantly changing world. The goal in OVMM is for a mobile robot to execute a series of manipulation commands given arbitrary language goals. We assume the following requirements for the memory module for dynamic, online operation:

[1]New York University, [2]Hello Robot Inc., [3]Meta Inc. * Denotes equal advising. Code and experiment videos: https://dynamem.github.io

- **Observations:** The mobile robot is equipped with an onboard RGB-D camera, and unlike prior work [9], doesn't start with a map of the environment. Rather, the robot explores the world and use the online observed sequence of posed RGB-D images to build its map.
- **Environment dynamism:** The environment can change without the knowledge of the robot.
- **Localization queries:** Given a natural language query (i.e. "teddy bear"), the memory module has to return the 3D location of the object or determine that the object doesn't exist in the scene observed thus far.
- **Obstacle queries:** The memory module must determine whether a point in space is occupied by an obstacle. Both the location of the objects and obstacles can move, previous observations often contradict each other and must be resolved by the memory.

B. *Dynamic 3D Voxel Map*

Our answer to the challenge posed in the Section II-A is DynaMem. DynaMem is an evolving sparse voxel map with associated information stored at each voxel. In each non-empty voxel, alongside its 3D location (x, y, z), we also store source image ID I (which image the voxel was backprojected from), a high-dimensional semantic feature vector f coming from a VLM like CLIP [19] or SigLIP [20], and the latest observation time, t, in seconds.

To make this data structure dynamic, we describe the process with which we add and update with new observations and remove outdated objects and associated voxels.

Adding Points: When the robot receives a new set of observations, i.e. RGB-D images with global poses, we convert them to 3D coordinates in a global reference frame, and generate a semantic feature vector for each point. The global coordinates are calculated from the global camera pose and the backprojected depth image using the known camera transformation matrix. Once we have calculated the points and associated features, we cluster the new points and assign them to the nearest voxel grids. The observation time and image ID are updated to keep track of the latest observation contributing to a particular voxel. If a voxel was empty before assignment, we assume its count $C = 0$ and feature vector $f = \overrightarrow{0}$.

Removing Points: When an object is moved or removed, its associated voxels in DynaMem may get removed. We use ray-casting to find the outdated voxels. The operation follows a simple principle: if a voxel falls within the frustum between the camera plane and the associated view point cloud, that voxel must be unoccupied. To reduce the impact of the depth noise at long range, we don't consider any pixel whose associated depth value is over 2m.

C. *Querying DynaMem for Object Localization*

As described in Section II-A, we define the object localization or 3D visual grounding problem as a function mapping a text query and posed RGBD images to either the 3D coordinate of the query object, or $\emptyset$ if the object is not in the scene. Unlike previous work, we abstain from returning a location when an object is not found. To enable this, we factor this grounding problem into two sub-problems. The first is finding the latest image where the queried object could have appeared. The second is identifying whether the object is actually present in that image. For the first sub-problem, we propose two alternate approaches of visual grounding: one using the intrinsic semantic features of DynaMem, and another using state-of-the-art multimodal LLMs such as GPT-4o [21] and Gemini 1.5 Pro [22]. Then, we use an open-vocabulary object detector model such as OWL-v2 [23] to search that image for the queried object. If we don't find the queried object, we assume that the object has either moved, or the response from the voxelmap or mLLM was inaccurate, and respond with "object not found". If OWL-v2 returns an object bounding box, we find the median pixel from the object mask and return its 3D location. To navigate in a real-world environment, robots use an obstacle map in conjunction with a navigation algorithm like A* in [9, 24].

III. Real world Experiments

We evaluate our method, DynaMem, on a Hello Robot: Stretch SE3 in real world environments. As a baseline, we compare with OK-Robot [9], a state-of-the-art method for OVMM. OK-Robot uses a static voxelmap as its memory representation, and thus it highlights the importance of dynamic memory for OVMM in a changing environment. For DynaMem, we run two variations of the algorithm in the real world: one with VLM-feature based queries and one with mLLM-QA based queries.

During our experiments in three dynamic environments and with 30 queries, we find that DynaMem with both VLM-feature based and mLLM-QA based queries have a total success rate of 70%. This is a significant improvement over the OK-Robot system, which has a total success rate of 30%. Notably, DynaMem is particularly adept at handling dynamic objects in the environment: only 6.7% of the trials failed due to our system not being able to navigate to such dynamic objects in the scene. This is in contrast to the OK-Robot system, where 53.3% of the trials failed because it could not find an object that moved in the environment. In contrast, navigating to static goals fails in only 10% of the cases for DynaMem with VLM-feature, 13.3% for OK-Robot and 20% for DynaMem with mLLM-QA.

IV. Conclusions

In this work, we introduced DynaMem, a spatio-semantic memory for open-vocabulary mobile manipulation that can handle changes to the environment during operation. We showed in three real world environments that DynaMem can navigate to, pick, and drop objects even while object and obstacle locations are changing.

References

[1] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner, Z. Kira, M. Savva, A. Chang, D. S. Chaplot, D. Batra, R. Mottaghi, Y. Bisk, and C. Paxton, "Homerobot: Open-vocabulary mobile manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2306.11565

[2] K. Ehsani, T. Gupta, R. Hendrix, J. Salvador, L. Weihs, K.-H. Zeng, K. P. Singh, Y. Kim, W. Han, A. Herrasti, *et al.*, "Imitating shortest paths in simulation enables effective navigation and manipulation in the real world," *arXiv preprint arXiv:2312.02976*, 2023.

[3] 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 (CVPR)*, June 2023, pp. 17 896–17 906.

[4] K.-H. Zeng, Z. Zhang, K. Ehsani, R. Hendrix, J. Salvador, A. Herrasti, R. Girshick, A. Kembhavi, and L. Weihs, "Poliformer: Scaling on-policy rl with transformers results in masterful navigators," 2024. [Online]. Available: https://arxiv.org/abs/2406.20083

[5] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," 2023. [Online]. Available: https://arxiv.org/abs/2210.05663

[6] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C. M. de Melo, J. B. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," 2023. [Online]. Available: https://arxiv.org/abs/2309.16650

[7] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlon, "Clio: Real-time task-driven open-set 3d scene graphs," *arXiv preprint arXiv:2404.13696*, 2024.

[8] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," in *International Conference on Computer Vision (ICCV)*, 2023.

[9] P. Liu, Y. Orru, J. Vakil, C. Paxton, N. M. M. Shafiullah, and L. Pinto, "Ok-robot: What really matters in integrating open-knowledge models for robotics," 2024. [Online]. Available: https://arxiv.org/abs/2401.12202

[10] R.-Z. Qiu, Y. Hu, G. Yang, Y. Song, Y. Fu, J. Ye, J. Mu, R. Yang, N. Atanasov, S. Scherer, and X. Wang, "Learning generalizable feature fields for mobile manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2403.07563

[11] B. Bolte, A. Wang, J. Yang, M. Mukadam, M. Kalakrishnan, and C. Paxton, "Usa-net: Unified semantic and affordance representations for robot memory," 2023. [Online]. Available: https://arxiv.org/abs/2304.12164

[12] M. Chang, T. Gervet, M. Khanna, S. Yenamandra, D. Shah, S. Y. Min, K. Shah, C. Paxton, S. Gupta, D. Batra, R. Mottaghi, J. Malik, and D. S. Chaplot, "Goat: Go to any thing," 2023. [Online]. Available: https://arxiv.org/abs/2311.06430

[13] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard, "Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation," *arXiv preprint arXiv:2403.17846*, 2024.

[14] 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. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.adf6991

[15] D. Z. Chen, A. X. Chang, and M. Nießner, "Scanrefer: 3d object localization in rgb-d scans using natural language," 2020. [Online]. Available: https://arxiv.org/abs/1912.08830

[16] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," 2017. [Online]. Available: https://arxiv.org/abs/1702.04405

[17] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4927–4936.

[18] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz, *et al.*, "Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data," *arXiv preprint arXiv:2111.08897*, 2021.

[19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[20] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," 2023. [Online]. Available: https://arxiv.org/abs/2303.15343

[21] O. Team, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[22] G. T. Google, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," 2024. [Online]. Available: https://arxiv.org/abs/2403.05530

[23] M. Minderer, A. Gritsenko, and N. Houlsby, "Scaling open-vocabulary object detection," 2024. [Online]. Available: https://arxiv.org/abs/2306.09683

[24] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," 2023. [Online]. Available: https://arxiv.org/abs/2210.05714