OK-Robot: What Really Matters in Integrating Open-Knowledge Models for Robotics

Peiqi Liu*[1] Yaswanth Orru*[1] Jay Vakil[2] Chris Paxton[2]
Nur Muhammad Mahi Shafiullah[†1] Lerrel Pinto[†1]

New York University[1], AI at Meta[2]

https://ok-robot.github.io

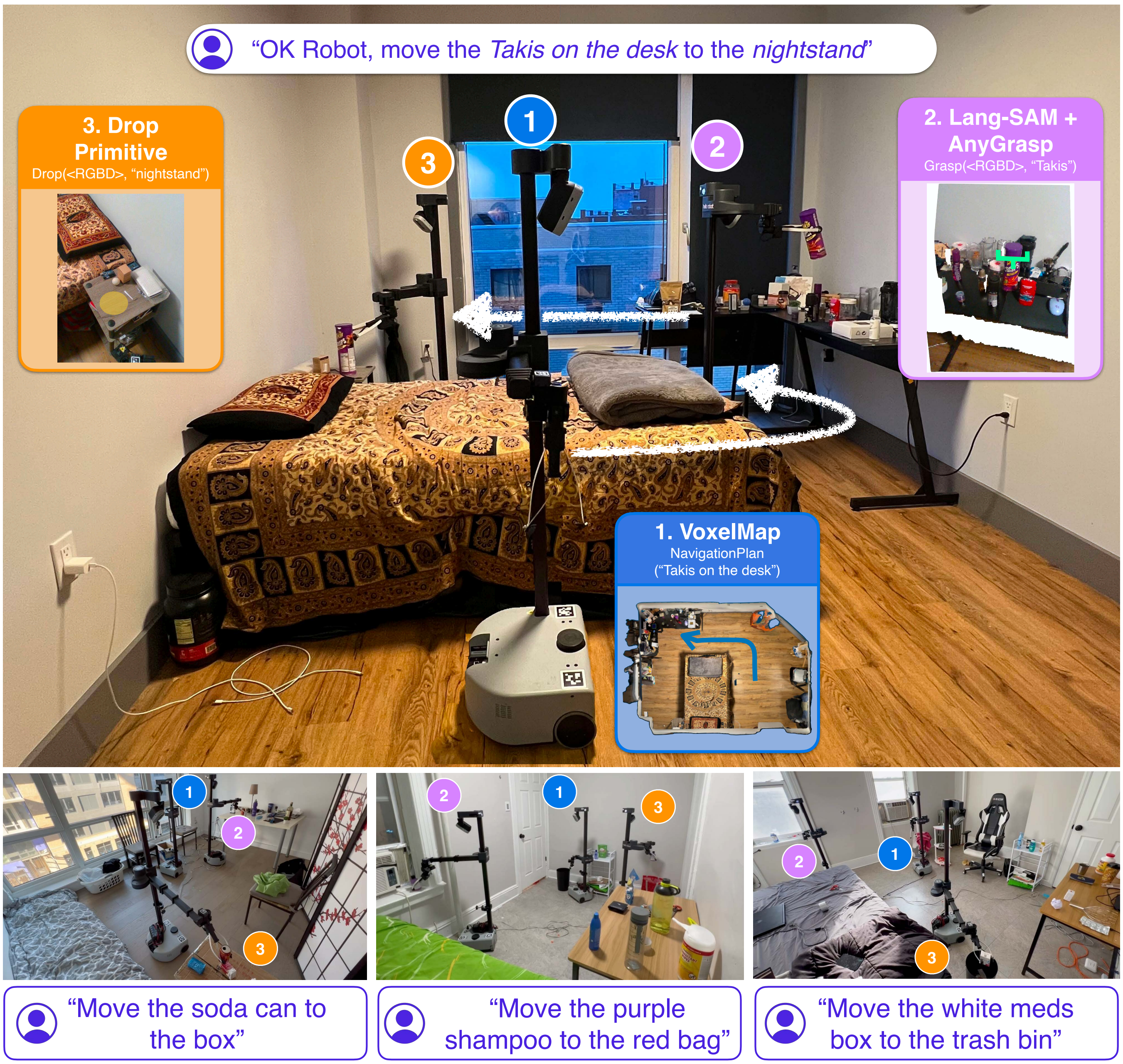


Fig. 1: OK-Robot is an Open Knowledge robotic system, which integrates a variety of learned models trained on publicly available data, to pick and drop objects in real-world environments. Using Open Knowledge models such as CLIP, Lang-SAM, AnyGrasp, and OWL-ViT, OK-Robot achieves a 58.5% success rate across 10 unseen, cluttered home environments, and 82.4% on cleaner, decluttered environments.

***Abstract*— Remarkable progress has been made in recent years in the fields of vision, language, and robotics. We now have vision models capable of recognizing objects based on language queries, navigation systems that can effectively control mobile systems, and grasping models that can handle a wide range of objects. Despite these advancements, general-purpose applications of robotics still lag behind, even though they rely on these fundamental capabilities of recognition, navigation, and grasping. In this paper, we adopt a systems-first approach to develop a new Open Knowledge-based robotics framework called `OK-Robot`. By combining Vision-Language Models (VLMs) for object detection, navigation primitives for movement, and grasping primitives for object manipulation, `OK-Robot` offers a integrated solution for pick-and-drop operations without requiring any training. To evaluate its performance, we run `OK-Robot` in 10 real-world home environments. The results demonstrate that `OK-Robot` achieves a 58.5% success rate in open-ended pick-and-drop tasks, representing a new state-of-the-art in Open Vocabulary Mobile Manipulation (OVMM) with nearly $1.8\times$ the performance of prior work. On cleaner, uncluttered environments, `OK-Robot`'s performance increases to 82%. However, the most important insight gained from `OK-Robot` is the critical role of nuanced details when combining Open Knowledge systems like VLMs with robotic modules. We published our code and robot videos on https://ok-robot.github.io to encourage further investigation.**

I. Introduction

Creating a general-purpose robot has been a longstanding dream of the robotics community. With the increase in data-driven approaches and large robot models, impressive progress is being made [1–4]. However, current systems are brittle, closed, and fail when encountering unseen scenarios. Even the largest robotics models can often only be deployed in previously seen environments [5, 6]. The brittleness of these systems is further exacerbated in settings where little robotic data is available, such as in unstructured home environments.

The poor generalization of robotic systems lies in stark contrast to large vision models [7–10], which show capabilities of semantic understanding [11–13], detection [7, 8], and connecting visual representations to language [9, 10, 14] At the same time, base robotic skills for navigation [15], grasping [16–19], and rearrangement [20, 21] are fairly mature. Hence, it is perplexing that robotic systems that combine modern vision models with robot-specific primitives perform so poorly. To highlight the difficulty of this problem, the recent NeurIPS 2023 challenge for open-vocabulary mobile manipulation (OVMM) [22] registered a success rate of 33% for the winning solution [23].

So what makes open-vocabulary robotics so hard? Unfortunately, there isn't a single challenge that makes this problem hard. Instead, inaccuracies in different components compound and together results in an overall drop. For example, the quality of open-vocabulary retrievals of objects in homes is dependent on the quality of query strings, navigation targets determined by VLMs may not be reachable to the robot, and the choice of different grasping models may lead to large differences in grasping performance. Hence, making progress on this problem requires a careful and nuanced framework that both integrates VLMs and robotics primitives, while being flexible enough to incorporate newer models as they are developed by the VLM and robotics community.

We present `OK-Robot`, an *Open Knowledge Robot* that integrates state-of-the-art VLMs with powerful robotics primitives for navigation and grasping to enable pick-and-drop. Here, *Open Knowledge* refers to learned models trained on large, publicly available datasets. When placed in a new home environment, `OK-Robot` is seeded with a scan taken from an iPhone. Given this scan, dense vision-language representations are computed using LangSam [24] and CLIP [9] and stored in a semantic memory. Then, when a language-query for an object to be picked comes in, semantic memory is queried with the language embedding to find that object. After this, navigation and picking primitives are applied sequentially to move to the desired object and pick it up. A similar process can be carried out for dropping the object.

To study `OK-Robot`, we tested it in 10 real world home environments. Through our experiments, we found that on a unseen natural home environment, a zero-shot deployment of our system achieves 58.5% success on average. However, this success rate is largely dependant on the "naturalness" of the environment, as we show that with improving the queries, decluttering the space, and excluding objects that are clearly adversarial (too large, too translucent, too slippery), this success rate reaches 82.4%. Overall, through our experiments, we make the following observations:

- **Pre-trained VLMs are highly effective for open-vocabulary navigation:** Current open-vocabulary vision-language models such as CLIP [9] or OWL-ViT [8] offer strong performance in identifing arbitrary objects in the real world, and enable navigating to them in a zero-shot manner (see Section II-A.)
- **Pre-trained grasping models can be directly applied to mobile manipulation:** Similar to VLMs, special purpose robot models pre-trained on large amounts of data can be applied out of the box to approach open-vocabulary grasping in homes. These robot models do not require any additional training or fine-tuning (see Section II-B.)
- **How components are combined is crucial:** Given the pretrained models, we find that they can be combined with no training using a simple state-machine model. We also find that using heuristics to counteract the robot's physical limitations can lead to a better success rate in the real world (see Section V.)
- **Several challenges still remain:** While, given the immense challenge of operating zero-shot in arbitrary homes, `OK-Robot` improves upon prior work, by analyzing the failure modes we find that there are significant improvements that can be made on the VLMs, robot models, and robot morphology, that will directly increase performance of open-knowledge manipulation agents (see Section VIII-C).

II. Technical Components and Method

Our method, on a high level, solves the problem described by the query: "Pick up **A** (from **B**) and drop it on/in **C**",

* Denotes equal contribution and † denotes equal advising.
Correspondence to: mahi@cs.nyu.edu

where A is an object and B and C are places in a real-world environment such as homes. The system we introduce is a combination of three primary subsystems combined on a Hello Robot: Stretch. Namely, these are the open-vocabulary object navigation module, the open-vocabulary RGB-D grasping module, and the dropping heuristic. In this section, we describe each of these components in more details and in Appendix Section V we discuss how we combine navigation, manipulation, placing primitives to deploy OK-Robot in real homes.

A. Open-home, open-vocabulary object navigation

The first component of our method is an open-home, open-vocabulary object navigation model that we use to map a home and subsequently navigate to any object of interest designated by a natural language query. The high level sketch of our navigation system can be viewed in Figure 3.

Scanning the home: For open vocabulary object navigation, we follow the approach from CLIP-Fields [25] and assume a pre-mapping phase where the home is "scanned" manually using an iPhone. This manual scan simply consists of taking a video of the home using the Record3D app on the iPhone, which results in a sequence of posed RGB-D images and takes less than one minute for a new room. Once collected, the RGB-D images, along with the camera pose and positions, are exported to our library for map-building.

Detecting objects: On each frame of the scan, we run an open-vocabulary object detector. We apply the detector on every frame, and extract each of the object bounding box, CLIP-embedding, detector confidence, and pass these information onto the object memory module. We further refine the bounding boxes into object masks with Segment Anything (SAM) [26]. Note that, in many cases, open-vocabulary object detectors require a set of natural language object queries to be detected. We supply a large set of such object queries, derived from the original Scannet200 labels [27] and presented in Appendix III, to help the detector captures most common objects in the scene.

Object-centric semantic memory: We use an object-centric memory similar to Clip-Fields [25] that we call the VoxelMap. VoxelMap is built by back-projecting the object masks in real-world coordinates using the depth image and the pose collected by the camera. This process gives us a point cloud where each point has an associated CLIP semantic vector. Then, we voxelize the point cloud to a 5 cm resolution. For each voxel, we calculate the detector-confidence weighted average for the CLIP embeddings that belong to that voxel.

Querying the memory module: Our semantic object memory gives us a static world representation represented as possibly non-empty voxels in the world, and a semantic vector in CLIP space associated with each voxel. Given a language query, we first convert it to a semantic vector using the CLIP language encoder. Then, we find the voxel where the dot product between the encoded embedding and the voxel's associated embedding is maximized. Since each voxel is associated with a real location in the home, this lets us find the location where a queried object is most likely to be found, similar to Figure 3(a).

We also implement querying for "A on B" by interpreting it as "A near B". We do so by selecting top-10 points for query A and top-50 points for query B. Then, we calculate the 10×50 pairwise L_2 distances and pick the A-point associated with the shortest (A, B) distance.

Navigating to objects in the real world: Once our navigation model gives us a 3D location coordinate in the real world, we use that as a navigation target for our robot to initialize our manipulation phase. To navigate the robot to the navigation target, we design a heuristic based navigation stack consisting of three components:

1) A navigation score function evaluating all robot navigable positions. Positions close to the target object but keeping a small but non-negligible space from obstacles will have high navigation score.
2) An obstacle map building approach similar to [28, 29] that marks all obstacle points and unexplored points as non-navigable.
3) An A* algorithm that generates a waypoint path given a navigation target position and an obstacle map.

When receiving a point coordinate of the object of interest, we first build an obstacle map from Record3D RGB-D images; then we select the best navigation target from all navigable positions by maximizing navigation score function; finally we use A* to generate a waypoint path which robots follow to initialize manipulation phase. More details about the navigation stack can be found in Appendix Section II.

B. Open-vocabulary grasping in the real world

Grasping or physically interacting with arbitrary objects in the real world is much more complex than open-vocabulary navigation. We opt for using a pre-trained grasping model to generate grasp poses in the real world, and filter them with language-conditioning using a modern VLM.

Grasp perception: Once the robot reaches the object location using the navigation method outlined in Section II-A, we use a pre-trained grasping model, AnyGrasp [19], to generate a grasp for the robot. We point the robot's RGB-D head camera towards the object's 3D location, given to us by the semantic memory, and capture an RGB-D image from it (Figure 4, column 1). We backproject and convert the depth image to a pointcloud and pass this information to the grasp generation model. Our grasp generation model, AnyGrasp, generates all collision free grasps (Figure 4 column 2) for a parallel jaw gripper in a scene given a single RGB image and a pointcloud. AnyGrasp provides us with grasp point, width, height, depth, and a "graspness score", indicating uncalibrated model confidence in each grasp.

Filtering grasps using language queries: Once we get all proposed grasps from AnyGrasp, we filter them using LangSam [24]. LangSam [24] segments the captured image and finds the desired object mask with a language query (Figure 4 column 3). We project all the proposed grasp points onto the image and find the grasps that fall into the object mask (Figure 4 column 4). We pick the best grasp using a heuristic. Given a grasp score $\mathcal{S}$ and the angle between the grasp normal and floor normal θ, the new heuristic score is $\mathcal{S} - (\theta^4/10)$. This heuristic balances high graspness scores with finding flat, horizontal grasps. We prefer horizontal grasps because they are robust to small calibration errors on the robot, while vertical grasps needs better hand-eye calibration to be successful. Robustness to hand-eye calibration errors lead to higher success as we transport the robot to different homes during our experiments.

Grasp execution: Once we identify the best grasp (Figure 4 column 5), we use a simple pre-grasp approach [30] to grasp our intended object. If $\overrightarrow{p}$ is the grasp point and $\overrightarrow{a}$ is the approach vector given by the grasping model, our robot gripper follows the following trajectory:

$$\langle \overrightarrow{p} - 0.2\overrightarrow{a},\ \overrightarrow{p} - 0.08\overrightarrow{a},\ \overrightarrow{p} - 0.04\overrightarrow{a},\ \overrightarrow{p} \rangle$$

Put simply, our method approaches the object from a pre-grasp position in a line with progressively smaller motions. Moving slower as we approach the object helps the robot not knock over light objects. Once we reach the predicted grasp point, we close the gripper in a close loop fashion to get a solid grip on the object without crushing it. After grasping the object, we lift up the robot arm, retract it fully, and rotate the wrist to have the object tucked over the body. This behavior maintains the robot footprint while ensuring the object is held securely by the robot and doesn't fall while navigating to the drop location.

C. Dropping heuristic

After picking up an object, we find and navigate to the drop location using the same methods described in Section II-A. Unlike in HomeRobot's baseline implementation [31] that assumes that the drop-off location is a flat surface, we extend our heuristic to cover concave objects such as sink, bins, boxes, and bags. First, we segment the point cloud P captured by the robot's head camera using LangSam [24] similar to Section II-B using the drop language query. Then we use a heuristic, more detailed description in Appendix IV, to determine the drop point and move the robot gripper above the drop point, and open the gripper to drop the object. While this heuristic doesn't explicitly reason about clutter, in our experiments it performs well on average.

III. Experiments

We evaluate our method in two set of experiments. On the first set of experiments, we evaluate between multiple alternatives for each of our navigation and manipulation modules. These experiments give us insights about which modules to use and evaluate in a home environment as a part of our method. On the next set of experiments, we took our robots to 10 homes and ran 171 pick-and-drop experiments to empirically evaluate how our method performs in completely novel homes, and to understand the failure modes of our system.

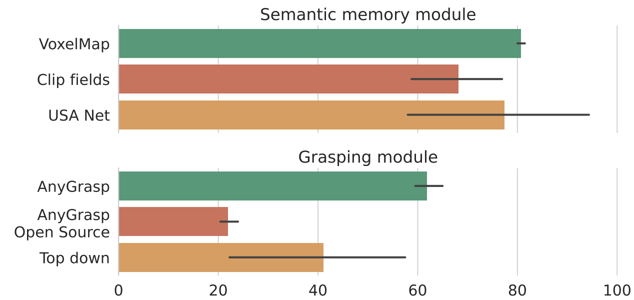


Fig. 2: Ablation experiment using different semantic memory and grasping modules, with the bars showing average performance and the error bars showing standard deviation over the environments.

A. Results of home experiments

Over the 10 home environment, `OK-Robot` achieved a 58.5% success rates in completing full pick-and-drops. Notably, this success rate is over novel objects sourced from each home with our zero-shot algorithm. As a result, each success and failure of the robot tells us something interesting about applying open-knowledge models in robotics, which we analyze over the next sections. In Appendix XI, we provide the details of all our home experiments and results from the same. In Appendix VI we show a subset of the target objects and in Appendix VII we show snapshots of homes where `OK-Robot` was deployed. Snippets of our experiments are in Figure 1, and full videos are presented on our project website.

B. Understanding the performance of `OK-Robot`

While our method can show zero-shot generalization in completely new environments, we probe `OK-Robot` to better understand its failure modes. Primarily, we elaborate on how our model performed in novel homes, what were the biggest challenges, and discuss potential solutions to them.

We show a breakdown of the failures in Figure 5. The three leading causes of failures are failing to retrieve the right object to navigate to from the semantic memory (9.3%), getting a difficult pose from the manipulation module (8.0%), and robot hardware difficulties (7.5%).

In Appendix VIII, we present results for ablation studies over system components, impact of clutter, object ambiguity, and affordance and detailed analysis of the failure modes presented in Figure 5.

Acknowledgments

NYU authors are supported by grants from Amazon, Honda, and ONR award numbers N00014-21-1-2404 and N00014-21-1-2758. NMS is supported by the Apple Scholar in AI/ML Fellowship. LP is supported by the Packard Fellowship. Our utmost gratitude goes to our friends and colleagues who helped us by hosting our experiments in their homes. Finally, we thank Siddhant Haldar, Paula Pascual and Ulyana Piterbarg for valuable feedback and conversations.

References

[1] Lerrel Pinto and Abhinav Gupta. *Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours*. 2015. arXiv: 1509.06825 [cs.LG].

[2] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection". In: *The International journal of robotics research* 37.4-5 (2018), pp. 421–436.

[3] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. "Do as I can, not as I say: Grounding language in robotic affordances". In: *Conference on Robot Learning (CoRL)* (2022).

[4] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. *On Bringing Robots Home*. 2023. arXiv: 2311.16098 [cs.RO].

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. "Rt-1: Robotics transformer for real-world control at scale". In: *arXiv preprint arXiv:2212.06817* (2022).

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. "Rt-2: Vision-language-action models transfer web knowledge to robotic control". In: *arXiv preprint arXiv:2307.15818* (2023).

[7] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. "Detecting twenty-thousand classes using image-level supervision". In: *European Conference on Computer Vision*. Springer. 2022, pp. 350–368.

[8] Matthias Minderer, Alexey Gritsenko, Austin Stone, et al. "Simple Open-Vocabulary Object Detection with Vision Transformers". In: *European Conference on Computer Vision*. Springer. 2022, pp. 728–755.

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *International Conference on Machine Learning (ICML)*. Vol. 139. 2021, pp. 8748–8763.

[10] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. "Ok-vqa: A visual question answering benchmark requiring external knowledge". In: *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*. 2019, pp. 3195–3204.

[11] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, et al. *Flamingo: a Visual Language Model for Few-Shot Learning*. 2022. arXiv: 2204.14198 [cs.CV].

[12] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, et al. *Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection*. 2023. arXiv: 2303.05499 [cs.CV].

[13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. *Visual Instruction Tuning*. 2023. arXiv: 2304.08485 [cs.CV].

[14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language models are unsupervised multitask learners". In: *OpenAI Blog* (2019).

[15] Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot. "Navigating to objects in the real world". In: *Science Robotics* 8.79 (2023), eadf6991.

[16] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13438–13444.

[17] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. *Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics*. 2017. arXiv: 1703.09312 [cs.RO].

[18] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. "Graspnet-1billion: a large-scale benchmark for general object grasping". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11444–11453.

[19] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains". In: *IEEE Transactions on Robotics* (2023).

[20] Ankit Goyal, Arsalan Mousavian, Chris Paxton, Yu-Wei Chao, Brian Okorn, Jia Deng, and Dieter Fox. "Ifor: Iterative flow minimization for robotic object rearrangement". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 14787–14797.

[21] Weiyu Liu, Yilun Du, Tucker Hermans, Sonia Chernova, and Chris Paxton. *StructDiffusion: Language-Guided Creation of Physically-Valid Structures using Unseen Objects*. 2023. arXiv: 2211.04604 [cs.RO].

[22] Sriram Yenamandra, Arun Ramachandran, Mukul Khanna, et al. "The HomeRobot Open Vocab Mobile Manipulation Challenge". In: *Thirty-seventh Conference on Neural Information Processing Systems: Competition Track*. 2023. URL: https://aihabitat.org/challenge/2023_homerobot_ovmm/.

[23] Andrew Melnik, Michael Büttner, Leon Harz, Lyon Brown, Gora Chand Nandi, Arjun PS, Gaurav Kumar Yadav, Rahul Kala, and Robert Haschke. "UniTeam: Open Vocabulary Mobile Manipulation Challenge". In: *arXiv preprint arXiv:2312.08611* (2023).

[24] Luca Medeiros. *Lang Segment Anything*. https://github.com/luca-medeiros/lang-segment-anything. 2023.
[25] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. *CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory*. 2023. arXiv: 2210.05663 [cs.RO].
[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. "Segment Anything". In: *ICCV*. 2023, pp. 4015–4026.
[27] David Rozenberszki, Or Litany, and Angela Dai. *Language-Grounded Indoor 3D Semantic Segmentation in the Wild*. 2022. arXiv: 2204.07761 [cs.CV].
[28] Benjamin Bolte, Austin Wang, Jimmy Yang, Mustafa Mukadam, Mrinal Kalakrishnan, and Chris Paxton. *USA-Net: Unified Semantic and Affordance Representations for Robot Memory*. 2023. arXiv: 2304.12164 [cs.RO].
[29] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. "Audio Visual Language Maps for Robot Navigation". In: *arXiv preprint arXiv:2303.07522* (2023).
[30] Sudeep Dasari, Abhinav Gupta, and Vikash Kumar. *Learning Dexterous Manipulation from Exemplar Object Trajectories and Pre-Grasps*. 2023. arXiv: 2209.11221 [cs.RO].
[31] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, et al. "HomeRobot: Open Vocabulary Mobile Manipulation". In: *arXiv preprint arXiv:2306.11565* (2023). URL: https://github.com/facebookresearch/home-robot.
[32] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL].
[33] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *European Conference on Computer Vision (ECCV)* 65.1 (2020), pp. 99–106.
[34] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. "Acronym: A large-scale grasp dataset based on simulation". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 6222–6227.
[35] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. "Goat: Go to any thing". In: *arXiv preprint arXiv:2311.06430* (2023).
[36] Santiago Garrido, Luis Moreno, Mohamed Abderrahim, and Fernando Martin. "Path planning for mobile robot navigation using voronoi diagram and fast marching". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, pp. 2376–2381.
[37] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].
[38] Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. "Visual Representations for Semantic Target Driven Navigation". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. May 2019, pp. 8846–8852.
[39] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. "A persistent spatial semantic representation for high-level natural language instruction execution". In: *Conference on Robot Learning*. PMLR. 2022, pp. 706–717.
[40] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. "Film: Following instructions in language with modular methods". In: *arXiv preprint arXiv:2110.07342* (2021).
[41] Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D'Arpino, Kiana Ehsani, Ali Farhadi, et al. "Retrospectives on the embodied ai workshop". In: *arXiv preprint arXiv:2210.06849* (2022).
[42] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. "Objaverse: A universe of annotated 3d objects". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 13142–13153.
[43] Huy Ha and Shuran Song. *Semantic Abstraction: Open-World 3D Scene Understanding from 2D Vision-Language Models*. 2022. arXiv: 2207.11514 [cs.CV].
[44] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. "Visual language maps for robot navigation". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 10608–10615.
[45] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. "Open-vocabulary Queryable Scene Representations for Real World Planning". In: *arXiv preprint arXiv:2209.09874*. 2022.
[46] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. "Conceptfusion: Open-set multimodal 3d mapping". In: *arXiv preprint arXiv:2302.07241* (2023).
[47] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. "Lerf: Language embedded radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 19729–19739.
[48] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. "ViNT: A Foundation Model for Visual Navigation". In: *7th Annual Conference on Robot Learning (CoRL)*. 2023.
[49] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agar-

wal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning". In: *arXiv preprint arXiv:2309.16650* (2023).

[50] Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. "Robot Learning in Homes: Improving Generalization and Reducing Dataset Bias". In: *Advances in Neural Information Processing Systems* 31 (2018), pp. 9094–9104.

[51] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics". In: *Robotics: Science and Systems (RSS)*. 2017.

[52] Jeffrey Mahler, Matthew Matl, Xinyu Liu, Albert Li, David Gealy, and Ken Goldberg. *Dex-Net 3.0: Computing Robust Robot Vacuum Suction Grasp Targets in Point Clouds using a New Analytic Model and Deep Learning*. 2018. arXiv: 1709.06670 [cs.RO].

[53] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. "QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation". In: *arXiv preprint arXiv:1806.10293* (2018).

[54] Yuzhe Qin, Rui Chen, Hao Zhu, Meng Song, Jing Xu, and Hao Su. *S4G: Amodal Single-view Single-Shot SE(3) Grasp Detection in Cluttered Scenes*. 2019. arXiv: 1910.14218 [cs.RO].

[55] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. *6-DOF GraspNet: Variational Grasp Generation for Object Manipulation*. 2019. arXiv: 1905.10520 [cs.CV].

[56] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, and D. Fox. "Progprompt: Generating situated robot task plans using large language models". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, p. 11523.

[57] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. "Perceiver-Actor: A multi-task transformer for robotic manipulation". In: *CoRL*. PMLR. 2023, pp. 785–799.

[58] Priyam Parashar, Jay Vakil, Sam Powers, and Chris Paxton. "Spatial-Language Attention Policies for Efficient Robot Learning". In: *arXiv preprint arXiv:2304.11235* (2023).

[59] Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. "Behavior Transformers: Cloning k modes with one stone". In: *Advances in neural information processing systems* 35 (2022), pp. 22955–22968.

[60] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. *From Play to Policy: Conditional Behavior Generation from Uncurated Robot Data*. 2022. arXiv: 2210.10047 [cs.RO].

[61] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. "Act3D: 3D Feature Field Transformers for Multi-Task Robotic Manipulation". In: *Conference on Robot Learning*. PMLR. 2023, pp. 3949–3965.

[62] Wenxuan Zhou, Bowen Jiang, Fan Yang, Chris Paxton, and David Held. "Learning Hybrid Actor-Critic Maps for 6D Non-Prehensile Manipulation". In: *arXiv preprint arXiv:2305.03942* (2023).

[63] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. "CLIPort: What and where pathways for robotic manipulation". In: *CoRL*. PMLR. 2022, pp. 894–906.

[64] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. "Learning latent plans from play". In: *CoRL*. PMLR. 2020, pp. 1113–1132.

[65] Corey Lynch and Pierre Sermanet. "Language Conditioned Imitation Learning over Unstructured Data". In: *Robotics: Science and Systems* (2021). URL: https://arxiv.org/abs/2005.07648.

[66] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. "VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models". In: *CoRL*. 2023.

[67] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. "Code as Policies: Language model programs for embodied control". In: *icra*. IEEE. 2023, pp. 9493–9500.

[68] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. "Voyager: An Open-Ended Embodied Agent with Large Language Models". In: *arXiv preprint arXiv: Arxiv-2305.16291* (2023).

[69] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. "ProgPrompt: Generating Situated Robot Task Plans using Large Language Models". In: *ICRA*. IEEE. 2023, pp. 11523–11530.

[70] Naoki Yokoyama, Alex Clegg, Joanne Truong, Eric Undersander, Tsung-Yen Yang, Sergio Arnaud, Sehoon Ha, Dhruv Batra, and Akshara Rai. "ASC: Adaptive Skill Coordination for Robotic Mobile Manipulation". In: *arXiv preprint arXiv:2304.00410* (2023).

[71] Austin Stone, Ted Xiao, Yao Lu, et al. *Open-World Object Manipulation using Pre-trained Vision-Language Models*. 2023. arXiv: 2303.00905 [cs.RO].

[72] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. "GPT-4V(ision) for Robotics: Multimodal Task Planning from Human Demonstration". In: *arXiv preprint arXiv:2311.12015* (2023).

[73] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning". In: *arXiv preprint arXiv:2307.06135* (2023).

[74] Charles C Kemp, Aaron Edsinger, Henry M Clever, and Blaine Matulevich. “The design of Stretch: A compact, lightweight mobile manipulator for indoor human environments”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 3150–3157.

Appendix I
Description of alternate system components

In this section, we provide more details about the alternate system components that we evaluated in Section VIII-A.

Alternate semantic navigation strategies: We evaluate the following semantic memory modules:

- **VoxelMap [31]:** VoxelMap converts every detected object to a semantic vector and stores such info into an associated voxel. Occupied voxels serve as an obstacle map.
- **CLIP-Fields [25]:** CLIP-Fields converts a sequence of posed RGB-D images to a semantic vector field by using open-label object detectors and semantic language embedding models. The result associates each point in the space with two semantic vectors, one generated via a VLM [9], and another generated via a language model [32], which is then embedded into a neural field [33].
- **USA-Net [28]:** USA-Net generates multi-scale CLIP features and embeds them in a neural field that also doubles as a signed distance field. As a result, a single model can support both object retrieval and navigation.

We compare them in the same three environments with a fixed set of queries, the results of which are shown in Figure 2.

Alternate grasping strategies: Similarly, we compare multiple grasping strategies to find out the best grasping strategy for our method.

- **AnyGrasp [19]:** AnyGrasp is a single view RGB-D based grasping model. It is trained on the GraspNet dataset which contains 1B grasp labels.
- **Open Graspness [19]:** Since the AnyGrasp model is free but not open source, we use an open licensed baseline trained on the same dataset.
- **Contact-GraspNet [16]:** We use Contact-GraspNet as a prior work baseline, which is trained on the Acronym [34] dataset. One limitation of Contact-GraspNet is that it was trained on a fixed camera view for a tabletop setting. As a result, in our application with a moving camera and arbitrary locations, it failed to give us meaningful grasps.
- **Top-down grasp [31]:** As a heuristic based baseline, we compare with the top-down heuristic grasp provided in the HomeRobot project.

Appendix II
Detailed navigation stack

In this section, we provide more details about how a design a heuristic based navigation stack to navigate the robot in front of target object.

Going and looking at an object [15, 25, 35] can be done while remaining at a safe distance from the object itself. In contrast, our navigation module must place the robot at an arms length so that the robot can manipulate the target object afterwards. Thus, our navigation method has to balance the following objectives:

1) The robot needs to be close enough to the object to manipulate it,
2) The robot needs some space to move its gripper, so there needs to be a small but non-negligible space between the robot and the object, and,
3) The robot needs to avoid collision during manipulation, and thus needs to keep its distance from all obstacles.

We use three different navigation score functions, each associated with one of the above points, and evaluate them on each point of the space to find the best position to place the robot. Let a random point be $\overrightarrow{x}$, the closest obstacle point as $\overrightarrow{x}_{obs}$, and the target object as $\overrightarrow{x_o}$. We define the following three functions s_1, s_2, s_3 to capture our three criterion. We define s as their weighted sum. The ideal navigation point $\overrightarrow{x}^*$ is the point in space that minimizes $s(\overrightarrow{x})$, and the ideal direction is given by the vector from $\overrightarrow{x^*}$ to $\overrightarrow{x_o}$.

$$
\begin{aligned}
s_1(\overrightarrow{x}) &= ||\overrightarrow{x} - \overrightarrow{x_o}|| \\
s_2(\overrightarrow{x}) &= 40 - \min(||\overrightarrow{x} - \overrightarrow{x_o}||, 40) \\
s_3(\overrightarrow{x}) &= \begin{cases} 1/||\overrightarrow{x} - \overrightarrow{x}_{obs}||, & \text{if } ||\overrightarrow{x} - \overrightarrow{x}_{obs}||_0 \leq 30 \\ 0, & \text{otherwise} \end{cases} \\
s(\overrightarrow{x}) &= s_1(\overrightarrow{x}) + 8s_2(\overrightarrow{x}) + 8s_3(\overrightarrow{x})
\end{aligned}
$$

To navigate to this target point safely from any other point in space, we follow a similar approach to [28, 29] by building an obstacle map from our captured posed RGB-D images. We build a 2D, 10cm×10cm grid of obstacles over which we navigate using the A* algorithm. To convert our VoxelMap to an obstacle map, we first set a floor and ceiling height. Presence of occupied voxels in between them implies the grid cell is occupied, while presence of neither ceiling nor floor voxels mean that the grid cell is unexplored. We mark both occupied or unexplored cells as not navigable. Around each occupied point, we mark any point within a 20 cm radius as also non-navigable to account for the robot's radius and a turn radius. During A* search, we use the s_3 as a heuristic function on the node costs to navigate further away from any obstacles, which makes our generated paths similar to ideal Voronoi paths [36] in our experiments.

Appendix III
ScanNet200 text queries

To detect objects in a given home environment using OWL-ViT, we use the Scannet200 labels. The full label set is here: `['shower head', 'spray', 'inhaler', 'guitar case', 'plunger', 'range hood', 'toilet paper dispenser', 'adapter', 'soy sauce', 'pipe', 'bottle', 'door', 'scale', 'paper towel', 'paper towel roll', 'stove', 'mailbox', 'scissors', 'tape', 'bathroom stall', 'chopsticks', 'case of water bottles', 'hand sanitizer', 'laptop', 'alcohol disinfection', 'keyboard', 'coffee maker', 'light', 'toaster', 'stuffed animal', 'divider', 'clothes dryer', 'toilet seat cover dispenser', 'file cabinet', 'curtain', 'ironing`

(a) Open-vocabulary object localization using VoxelMap

(b) Open-vocabulary navigation planning using VoxelMap and heuristics weighted A*

Fig. 3: Open-vocabulary, open knowledge object localization and navigation in the real-world. We use the VoxelMap [31] for localizing objects with natural language queries, and use an A* algorithm similar to USANet [28] for path planning.

Fig. 4: Open-vocabulary grasping in the real world. From left to right, we show the (a) robot POV image, (b) all suggested grasps from AnyGrasp [19], (c) object mask given label from LangSam [24], (d) grasp points filtered by the mask, and (e) grasp chosen for execution.

board’, ’fire extinguisher’, ’fruit’, ’object’, ’blinds’, ’container’, ’bag’, ’oven’, ’body wash’, ’bucket’, ’cd case’, ’tv’, ’tray’, ’bowl’, ’cabinet’, ’speaker’, ’crate’, ’projector’, ’book’, ’school bag’, ’laundry detergent’, ’mattress’, ’bathtub’, ’clothes’, ’candle’, ’basket’, ’glass’, ’face wash’, ’notebook’, ’purse’, ’shower’, ’power outlet’, ’trash bin’, ’paper bag’, ’water dispenser’, ’package’, ’bulletin board’, ’printer’, ’windowsill’, ’disinfecting wipes’, ’bookshelf’, ’recycling bin’, ’headphones’, ’dresser’, ’mouse’, ’shower gel’, ’dustpan’, ’cup’, ’storage organizer’, ’vacuum cleaner’, ’fireplace’, ’dish rack’, ’coffee kettle’, ’fire alarm’, ’plants’, ’rag’,

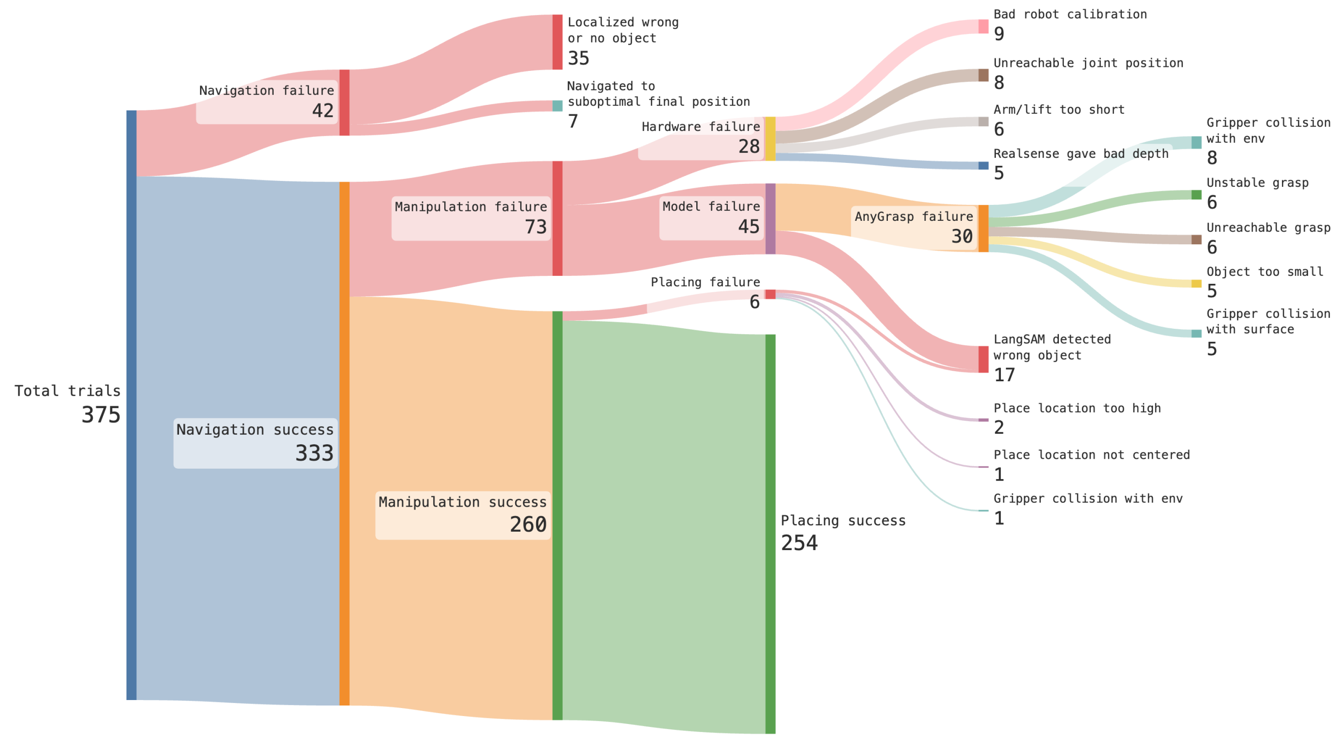


Fig. 5: All the success and failure cases in our home experiments, aggregated over all three cleaning phases, and broken down by mode of failure. From left to right, we show the application of the three components of `OK-Robot`, and show a breakdown of the long-tail failure modes of each of the components.

'can', 'piano', 'bathroom cabinet', 'shelf', 'cushion', 'monitor', 'fan', 'tube', 'box', 'blackboard', 'ball', 'bicycle', 'guitar', 'trash can', 'hand sanitizers', 'paper towel dispenser', 'whiteboard', 'bin', 'potted plant', 'tennis', 'soap dish', 'structure', 'calendar', 'dumbbell', 'fish oil', 'paper cutter', 'ottoman', 'stool', 'hand wash', 'lamp', 'toaster oven', 'music stand', 'water bottle', 'clock', 'charger', 'picture', 'bascketball', 'sink', 'microwave', 'screwdriver', 'kitchen counter', 'rack', 'apple', 'washing machine', 'suitcase', 'ladder', 'ping pong ball', 'window', 'dishwasher', 'storage container', 'toilet paper holder', 'coat rack', 'soap dispenser', 'refrigerator', 'banana', 'counter', 'toilet paper', 'mug', 'marker pen', 'hat', 'aerosol', 'luggage', 'poster', 'bed', 'cart', 'light switch', 'backpack', 'power strip', 'baseball', 'mustard', 'bathroom vanity', 'water pitcher', 'closet', 'couch', 'beverage', 'toy', 'salt', 'plant', 'pillow', 'broom', 'pepper', 'muffins', 'multivitamin', 'towel', 'storage bin', 'nightstand', 'radiator', 'telephone', 'pillar', 'tissue box', 'vent', 'hair dryer', 'ledge', 'mirror', 'sign', 'plate', 'tripod', 'chair', 'kitchen cabinet', 'column', 'water cooler', 'plastic bag', 'umbrella', 'doorframe', 'paper', 'laundry hamper', 'food', 'jacket', 'closet door', 'computer tower', 'stairs', 'keyboard piano', 'person', 'table', 'machine', 'projector screen', 'shoe'].

APPENDIX IV
DROPPING HEURISTIC

Once segment the point cloud P captured by the robot's head camera using LangSam [24] similar to Section II-B using the drop language query. Then, we align that segmented point cloud such that X-axis is aligned with the way the robot is facing, Y-axis is to its left and right, and the Z-axis of the point cloud is aligned with the floor normal. Then, we normalize the point cloud so that the robot's (x, y) coordinate is $(0, 0)$, and the floor plane is at $z = 0$. We call this pointcloud P_a. On the aligned, segmented point cloud, we consider the (x, y) coordinates for each point, and find the median values x_m and y_m on each axis. Finally, we find a drop height using $z_{\max} = 0.2 + \max\{z \mid (x, y, z) \in P_a; 0 \leq x \leq x_m; |y - y_m| < 0.1\}$ on the segmented, aligned pointcloud. We add a small buffer of 0.2 to the height to avoid collisions between the robot and the drop location.

APPENDIX V
DEPLOYING OK-ROBOT IN HOMES

In this section, we elaborate more on how our navigation, pick, and drop primitives are combined to create our robot method that can be applied in any novel home. For a new home environment, we "scan" the room in under a minute. Then, it takes less than five minutes to process the scan into our VoxelMap. Once that is done, the robot can be immediately placed at the base and start operating. From arriving into a completely novel environment to start operating autonomously in it, our system takes under 10 minutes on average to complete the first pick-and-drop task.

Transitioning between modules: The transition between different modules is predefined and happens automatically once a user specifies the object to pick and where to drop it. Since we do not implement error detection or correction, our state machine model is a simple linear chain of steps leading from navigating to object, to grasping, to navigating to goal, and to dropping the object at the goal to finish the task.

Protocol for home experiments: To run our experiment in a novel home, we move the robot to a previously unobserved room. We record the scene and create our VoxelMap. Concurrently, we pick between 10-20 objects arbitrarily in each scene that can fit in the robot gripper. These are objects found in the scene, and are not chosen ahead of time. We come up with a language query for each chosen object using GPT-4V [37] to keep the queries consistent and free of experimenter bias. We query our navigation module to filter out all the navigation failures; i.e. objects that our semantic memory module could not locate properly. Then, we execute pick-and-drop on remaining objects sequentially without resets between trials.

APPENDIX VI
SAMPLE OBJECTS FROM OUR TRIALS

During our experiments, we tried to sample objects that can plausibly be manipulated by the Hello Robot: Stretch gripper from the home environments. As a result, `OK-Robot` encountered a large variety of objects with different shapes and visual features. A subsample of such objects are presented in the Figures 6, 7.

APPENDIX VII
SAMPLE HOME ENVIRONMENTS FROM OUR TRIALS

We show snapshots from a subset of home environments where we evaluated our method in Figures 11. Beyond the home experiment results presented here, we also reproduced `OK-Robot` in two homes in Pittsburgh, PA, and Fremont, CA. These homes were larger and more complex: a cluttered, actively-used home kitchen environment, and a large, controlled test apartment used in prior work [22, 31]. In Appendix Figure 12, we show the robot performing pick-and-drop in these two environments. These homes were different from our initial ten experiments in a few ways. Both were larger compared to the average NY homes, requiring more robot motion to navigate to different goals. The PA environment (Figure 12 top) notably had much more clutter. However, given only a scan, `OK-Robot` was able to successfully pick and drop objects like stuffed lion, plush cactus, toy drill, or green water bottle in both environments.

APPENDIX VIII
EXPERIMENT EVALUATIONS

A. Ablations over system components

Apart from the navigation and manipulation strategies used in `OK-Robot`, we also evaluated a number of alternative open vocabulary navigation and grasping modules. We compared them by evaluating them in three different environments in our lab. Apart from VoxelMap [31], we evaluate CLIP-Fields [25], and USA-Net [28] for semantic navigation. For grasping module, we consider AnyGrasp and its open-source baseline, Open Graspness [19], Contact-GraspNet [16], and Top-Down grasp heuristic from home-robot [31]. More details about them are provided in Appendix Section I.

In Figure 2, we see their comparative performance in three lab environments. For semantic memory modules, we see that VoxelMap, used in `OK-Robot` and described in Sec. II-A, outperforms other semantic memory modules by a small margin. It also has much lower variance compared to the alternatives, meaning it is more reliable. As for grasping modules, AnyGrasp clearly outperforms other grasping methods, performing almost 50% better in a relative scale over the next best candidate, top-down grasp.

B. Impact of clutter, object ambiguity, and affordance

What makes home environments especially difficult compared to lab experiments is the presence of physical clutter, language-to-object mapping ambiguity, and hard-to-reach positions. To gain a clear understanding of how such factors play into our experiments, we go through two "clean-up" processes in each environment. During the clean-up, we pick a subset of objects that are free from ambiguity from the previous rounds, clean the clutter around objects, and generally relocated them in an accessible locations. The two clean-up rounds at each environment gives us insights about the performance gap caused by the natural difficulties of a home-like environment.

We show a complete analysis of the tasks listed section III-A which failed in various stages in Figure 8. As we can see from this breakdown, as we clean up the environment and remove the ambiguous objects, the navigation accuracy goes up, and the total error rate goes down from 15% to 12% and finally all the way down to 4%. Similarly, as we clean up clutters from the environment, we find that the manipulation accuracy also improves and the error rates decrease from 25% to 16% and finally 13%. Finally, since the drop-module is agnostic of the label ambiguity or manipulation difficulty arising from clutter, the failure rate of the dropping primitive stays roughly constant through the three phases of cleanup.

C. Detailed analysis of common failure cases

Natural language queries for objects: One of the primary reasons our `OK-Robot` can fail is when a natural language

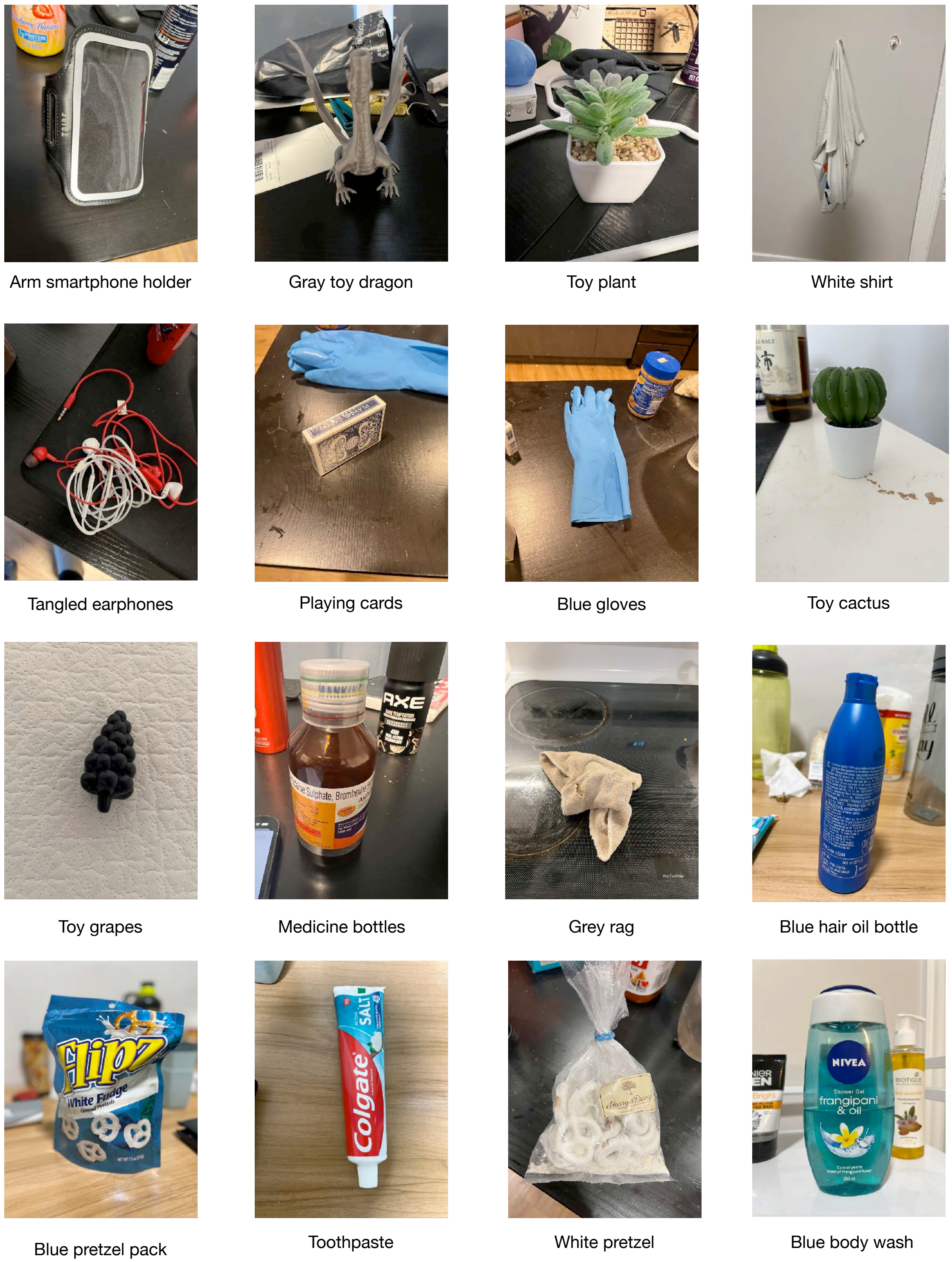
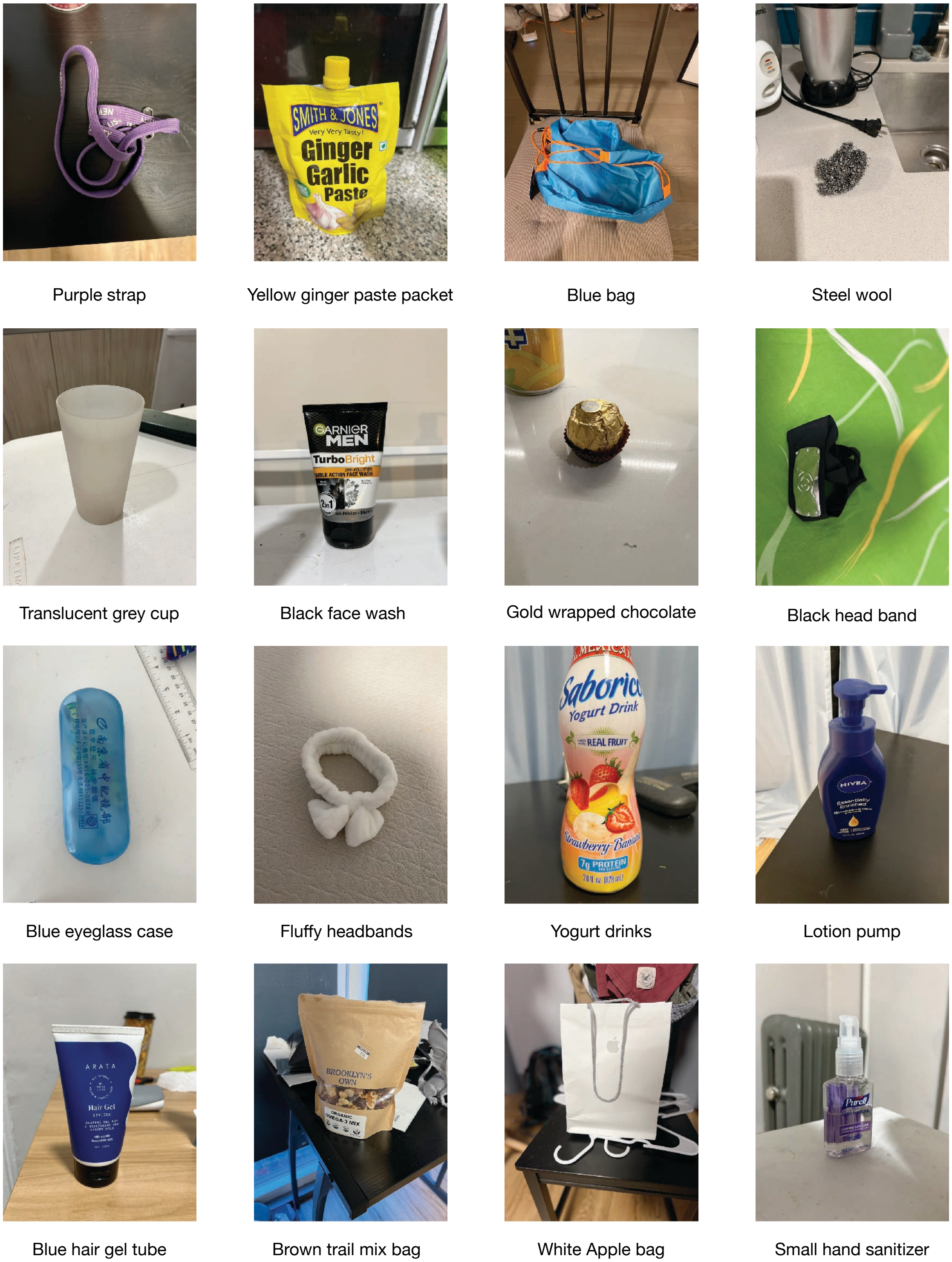


Fig. 6: Sample objects on our home experiments, sampled from each home environment, which `OK-Robot` was able to pick and drop successfully.



Purple strap | Yellow ginger paste packet | Blue bag | Steel wool

Translucent grey cup | Black face wash | Gold wrapped chocolate | Black head band

Blue eyeglass case | Fluffy headbands | Yogurt drinks | Lotion pump

Blue hair gel tube | Brown trail mix bag | White Apple bag | Small hand sanitizer

Fig. 7: Sample objects on our home experiments, sampled from each home environment, which `OK-Robot` **failed** to pick up successfully.

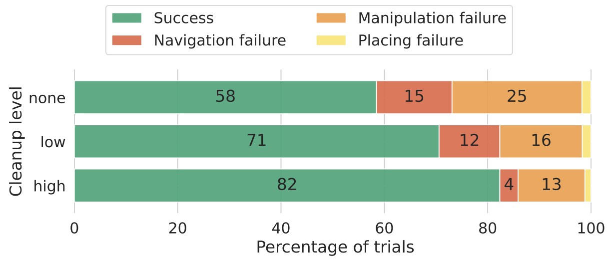


Fig. 8: Failure modes of our method in novel homes, broken down by the failures of the three modules and the cleanup levels.

query given by the user doesn't retrieve the intended object from the semantic memory. In Figure 9 we show how some queries may fail while semantically very similar but slightly modified wording of the same query might succeed.

Grasping module limitations: One failure mode of our manipulation module comes from executing grasps from a pre-trained manipulation model's output based on a single RGB-D image. Moreover, this model wasn't even designed for the Hello Robot: Stretch gripper. As a result, sometimes the proposed grasps are unreliable or unrealistic (Figure 10).

Sometimes, the grasp is infeasible given the robot joint limits, or is simply too far from the robot body. In other cases, the model generates a good grasp pose, but as the robot is executing the grasping primitive, it collides with some minor environment obstacle. Our grasping module also categorically struggles with flat objects, like chocolate bars and books, since it's difficult to grasp them off a surface with a two-fingered gripper.

Robot hardware limitations: While our robot of choice, a Hello Robot: Stretch, is able to pick-and-drop a variety of objects, certain hardware limitations also dictate what our system can and cannot manipulate. For example, the fully extended robot arm has a 1 kg payload limit, and thus our method is unable to pick objects like a full dish soap bottle. Similarly, objects that are far from navigable floor space, i.e. in the middle of a bed, or on high places, are difficult for the robot to reach because of the reach limits of the arm. The robot hardware or the RealSense camera can occasionally get miscalibrated over time, especially during continuous home operations. This miscalibration can lead to manipulation errors since that module requires hand-eye coordination in the robot. The robot base wheels have small diameters and in some cases struggle to move smoothly between carpet and floor.

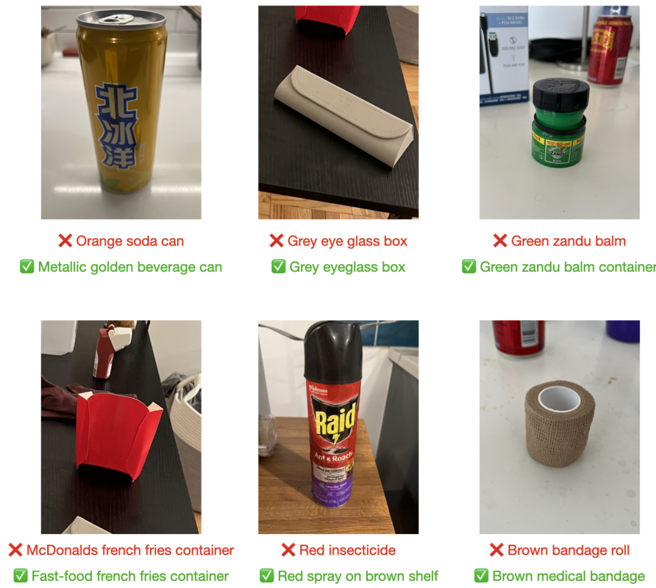


Fig. 9: Samples of failed or ambiguous language queries into our semantic memory module. Since the memory module depends on pretrained large vision language model, its performance shows susceptibility to particular "incantations" similar to current LLMs.

Fig. 10: Samples of failures of our manipulation module. Most failures stem from using only a single RGB-D view to generate the grasp and the limiting form-factor of a large two-fingered parallel jaw gripper.

Appendix IX
Related Works

A. Vision-Language models for robotic navigation

Early applications of pre-trained open-knowledge models in robotics has been in open-vocabulary navigation. Navigating to various objects is an important task which has been looked at in a wide range of previous works [31, 35, 38], as well as in the context of longer pick-and-place tasks [39, 40]. However, these methods have generally been applied to relatively small numbers of objects [41]. Recently, Objaverse [42] has shown navigation to thousands of object types, for example, but much of this work has been restricted to simulated or highly controlled environments.

The early work addressing this problem builds upon representations derived from pre-trained vision language models, such as SemAbs [43], CLIP-Fields [25], VLMaps [44], NLMap-SayCan [45], and later, ConceptFusion [46] and LERF [47]. Most of these models show object localization in pre-mapped scenes, while CLIP-Fields, VLMaps, and NLMap-SayCan show integration with real robots for indoor navigation tasks. USA-Nets [28] extends this task to include an affor-

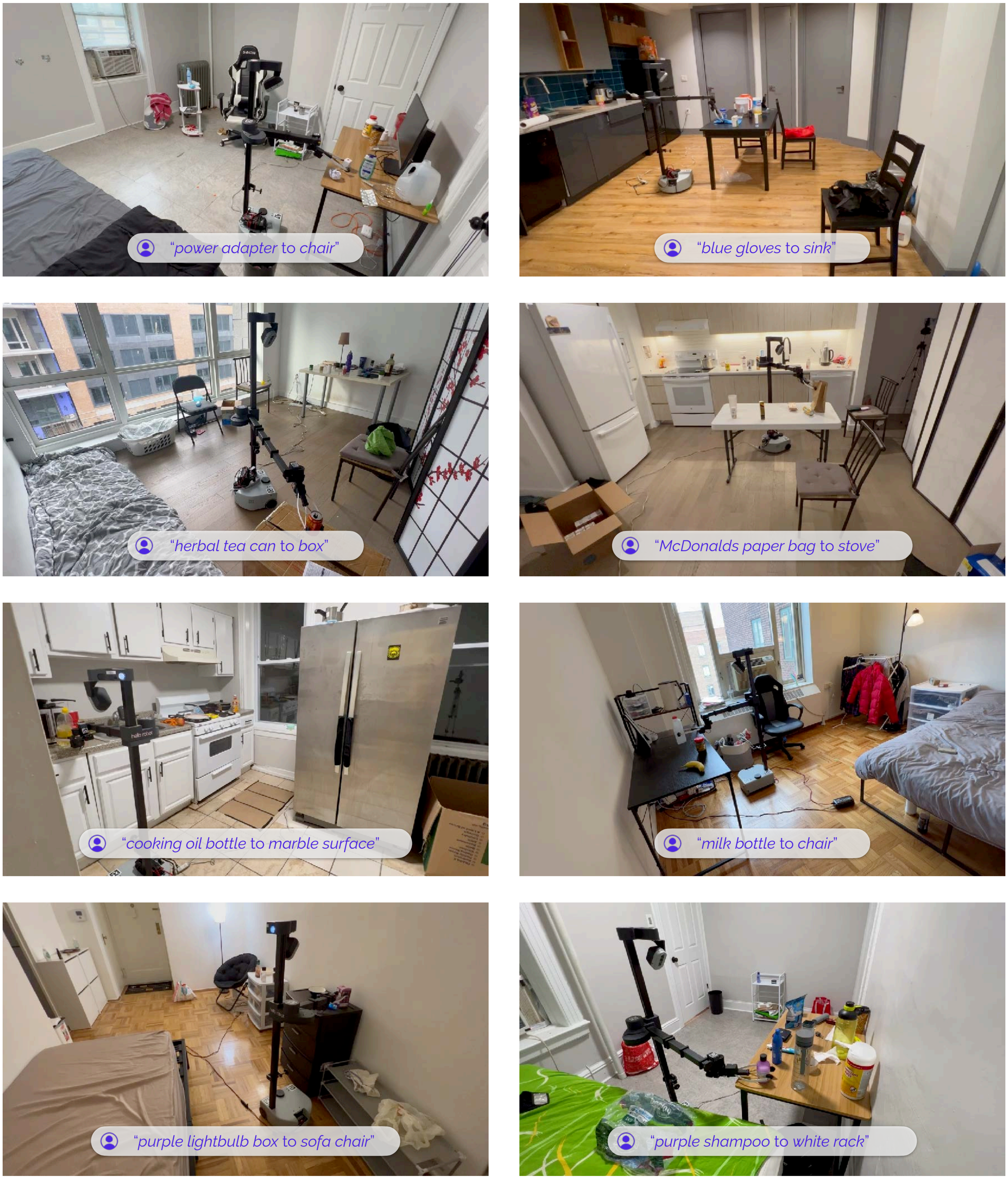


Fig. 11: Eight out of the ten New York home environments in which we evaluated `OK-Robot`. In each figure caption, we show the queries that the system is being evaluated on.

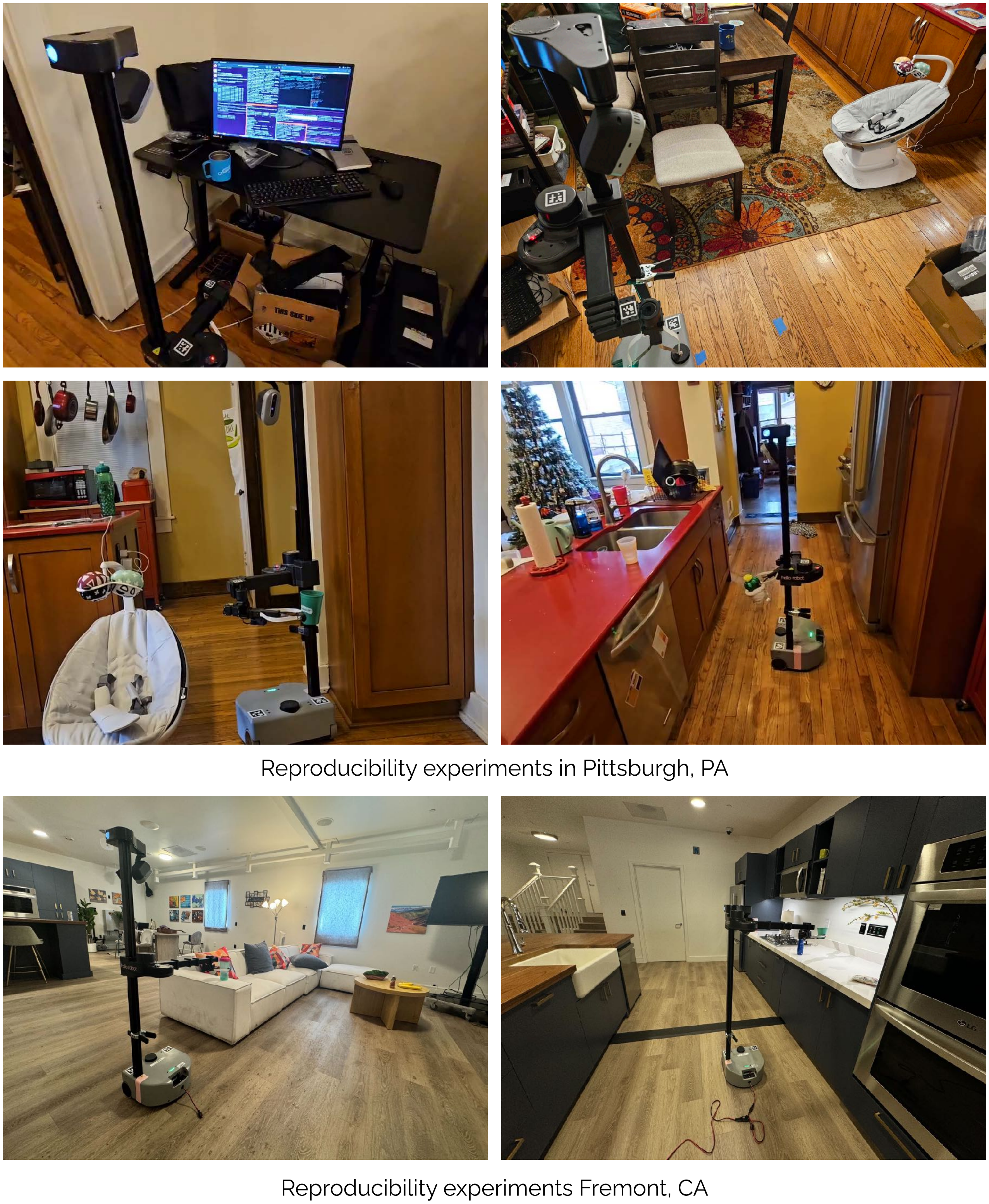


Fig. 12: Home environments outside of New York where we successfully reproduced `OK-Robot`. We ensured that `OK-Robot` can function in these homes by trying pick-and-drop on a number of objects in the homes.

dance model, navigating with open-vocabulary queries while doing object avoidance. ViNT [48] proposes a foundation model for robotic navigation which can be applied to vision-language navigation problems. More recently, GOAT [35] was proposed as a modular system for "going to anything" and navigating to any object in any environment given either language or image queries. ConceptGraphs [49] proposed an open scene graph representation capable of handling complex queries using LLMs. Any such open-vocabulary embodied model has the potential to improve modular systems like `OK-Robot`.

B. Pretrained robot manipulation models

While humans can frequently look at objects and immediately know how to grasp it, such grasping knowledge is not easily accessible to robots. Over the years, there has been many works that has focused on creating such a general robot grasp generation model [1, 50–55] for arbitrary objects and potentially cluttered scenes via learning methods. Our work focuses on more recent iterations of such methods [16, 19] that are trained on large grasping datasets [18, 34]. While these models only perform one task, namely grasping, they predict grasps across a large object surface and thus enable downstream complex, long-horizon manipulation tasks [20, 21, 56].

More recently, there is a set of general-purpose manipulation models moving beyond just grasping [57–61]. Some of these works perform general language-conditioned manipulation tasks, but are largely limited to a small set of scenes and objects. HACMan [62] demonstrates a larger range of object manipulation capabilities, focused on pushing and prodding. In the future, such models could expand the reach of our system.

C. Open vocabulary robot systems

Many recent works have worked on language-enabled tasks for complex robot systems. Some examples include language conditioned policy learning [57, 63–65], learning goal-conditioned value functions [3, 66], and using large language models to generate code [67–69]. However, a fundamental difference remains between systems which aim to operate on arbitrary objects in an open-vocab manner, and systems where one can specify one among a limited number of goals or options using language. Consequently, Open-Vocabulary Mobile Manipulation has been proposed as a key challenge for robotic manipulation [31]. There has previously been efforts to build such a system [70, 71]. However, unlike such previous work, we try to build everything on an open platform and ensure our method can work without having to re-train anything for a novel home. Recently, UniTeam [23] won the 2023 HomeRobot OVMM Challenge [22] with a modular system doing pick-and-place to arbitrary objects, with a zero-shot generalization requirement similar to ours.

In parallel, recently, there have been a number of papers doing open-vocabulary manipulation using GPT or especially GPT4 [37]. GPT4V can be included in robot task planning frameworks and used to execute long-horizon robot tasks, including ones from human demonstrations [72]. ConceptGraphs [49] is a good recent example, showing complex object search, planning, and pick-and-place capabilities to open-vocabulary objects. SayPlan [73] also shows how these can use used together with a scene graph to handle very large, complex environments, and multi-step tasks; this work is complementary to ours, as it doesn't handle how to implement pick and place.

Appendix X
Limitations, Open Problems and Requests for Research

While our method shows significant success in completely novel home environments, it also shows many places where such methods can improve. In this section, we discuss a few of such potential improvement in the future.

A. Live semantic memory and obstacle maps

All the current semantic memory modules and obstacle map builders build a static representation of the world, without a good way of keeping it up-to-date as the world changes. However, homes are dynamic environments, with many small changes over the day every day. Future research that can build a dynamic semantic memory and obstacle map would unlock potential for continuous application of such pick-and-drop methods in a novel home out of the box.

B. Grasp plans instead of proposals

Currently, the grasping module proposes generic grasps without taking the robot's body and dynamics into account. Similarly, given a grasp pose, often the open loop grasping trajectory collides with environmental obstacles, which can be easily improved by using a module to generate grasp plans rather than grasp poses only.

C. Improving interactivity between robot and user

One of the major causes of failure in our method is in navigation: where the semantic query is ambiguous and the intended object is not retrieved from the semantic memory. In such ambiguous cases, interaction with the user would go a long way to disambiguate the query and help the robot succeed more often.

D. Detecting and recovering from failure

Currently, we observe a multiplicative error accumulation between our modules: if any of our independent components fail, the entire process fails. As a result, even if our modules each perform independently at or above 80% success rate, our final success rate can still be below 60%. However, with better error detection and retrying algorithms, we can recover from much more single-stage errors, and similarly improve our overall success rate [23].

E. Robustifying robot hardware

While Hello Robot - Stretch [74] is an affordable and portable platform on which we can implement such an open-home system for arbitrary homes, we also acknowledge that with robust hardware such methods may have vastly enhanced capacity. Such robust hardware may enable us to reach high and low places, and pick up heavier objects. Finally, improved robot odometry will enable us to execute much more finer grasps than is possible today.

APPENDIX XI
LIST OF HOME EXPERIMENTS

A full list of experiments in homes can be found in Table I.

TABLE I: A list of all tasks in the home enviroments, along with their categories and success rates out of 10 trials.

Pick object	Place location	Result
Home 1		
Cleanup level: none		
silver cup	white table	Success
blue eye glass case	chair	Success
printed paper cup, coffee cup [white table]	----	Manipulation failure
small red and white medication	Chair	Success
power adapter	Grey Bed	Success
wrapped paper	----	Navigation failure
blue body wash	study table	Success
blue air spray	white table	Success
black face wash	----	Manipulation failure
yellow face wash	chair	Success
body spray	----	Navigation failure
small hand sanitizer	----	Manipulation failure
blue inhaler device(window)	white table	Success
inhaler box(window)	dust bin	Success
multivitamin container	----	Navigation failure
red towel	white cloth bin (air conditioner)	Success
white shirt	white cloth bin (air conditioner)	Success
Cleanup level: low		
silver cup	white table	Success
blue eye glass case	----	Navigation failure
printed paper cup, coffee cup [white table]	dust bin	Success
small red and white medication	Chair	Success
power adapter	----	Navigation failure
blue body wash	white table	Success
blue air spray	white table	Success
yellow face wash	white table	Success
small hand sanitizer	study table	Success
blue inhaler device(window)	----	Manipulation failure
inhaler box(window)	dust bin	Success
red towel	white cloth bin(air conditioner)	Success
white shirt	white cloth bin(air conditioner)	Success
Cleanup level: high		
silver cup	white table	Success
printed paper cup, coffee cup [white table]	dust bin	Success
blue body wash	white table	Success
blue air spray	white table	Success
yellow face wash	----	Manipulation failure
small hand sanitizer	----	Manipulation failure
inhaler box(window)	dust bin	Success
white shirt	white cloth bin(air conditioner)	Success
Home 2		
Cleanup level: None		
fanta can	dust bin	Success
tennis ball	small red shopping bag	Success
black head band [bed]	----	Manipulation failure
purple shampoo bottle	white rack	Success
toothpaste	small red shopping bag	Success
orange packaging	dust bin	Success

Continued on the next page

Pick object	Place location	Result
green hair cream jar [white rack]	----	Navigation failure
green detergent pack [white rack]	white table	Success
blue moisturizer [white rack]	----	Navigation failure
green plastic cover	----	Navigation failure
storage container	----	Manipulation failure
blue hair oil bottle	white rack	Success
blue pretzels pack	white rack	Success
blue hair gel tube	----	Manipulation failure
red bottle [white rack]	brown desk	Success
blue bottle [air conditioner]	white cloth bin(air conditioner)	Success
wallet	----	Manipulation failure
Cleanup level: low		
fanta can	black trash can	Success
tennis ball	red target bag	Success
black head band [bed]	red target bag	Success
purple shampoo bottle	red target bag	Success
toothpaste	red target bag	Success
orange packaging	black trash can	Success
green detergent pack [white rack]	----	Manipulation failure
blue moisturizer [white rack]	----	Navigation failure
blue hair oil bottle	white rack	Success
blue pretzels pack	white rack	Success
wallet	----	Manipulation failure
Cleanup level: high		
fanta can	black trash can	Success
purple shampoo bottle	small red shopping bag	Success
orange packaging	black trash can	Success
blue moisturizer [white rack]	white rack	Success
blue hair oil bottle	----	Manipulation failure
blue hair gel tube	dust bin	Success
red bottle [white rack]	target bag	Placing failure
blue bottle [air conditioner]	white cloth bin(air conditioner)	Success
Home 3		
Cleanup level: none		
apple	white plate	Success
ice cream	white and green bag	Success
green lime juice bottle	red basket	Success
yellow packet	----	Manipulation failure
red packet	----	Manipulation failure
orange can	card board box	Success
cooking oil bottle	----	Manipulation failure
pasta sauce	----	Manipulation failure
orange box [stove]	----	Manipulation failure
green bowl	sink	Success
washing gloves	green bag [card board box]	Success
small oregano bottle	red basket	Success
yellow noodles packet [stove]	red basket	Success
blue dish wash bottle	card board box	Success
scrubber	----	Navigation failure
dressing salad bottle	----	Navigation failure
Cleanup level: low		

Continued on the next page

Pick object	Place location	Result
apple	white plate	Success
ice cream	red basket	Success
green lime juice bottle	red basket	Success
yellow packet	green bag	Success
red packet	----	Manipulation failure
orange can	card board box	Success
cooking oil bottle	marble surface [red basket]	Success
green bowl	----	Manipulation failure
washing gloves	sink	Success
small oregano bottle	red basket	Success
yellow noodles packet [stove]	----	Manipulation failure
blue dish wash bottle	card board box	Success
	Cleanup level: high	
apple	white plate	Success
ice cream	red basket	Success
green lime juice bottle	red basket	Success
orange can	card board box	Success
cooking oil bottle	----	Manipulation failure
washing gloves	sink	Success
small oregano bottle	red basket	Success
yellow noodles packet [stove]	red basket	Success
blue dish wash bottle	card board box	Success
	Home 4	
	Cleanup level: none	
pepsi	black chair	Success
birdie	cloth bin	Success
black hat	----	Navigation failure
owl like wood carving	bed	Success
red inhaler	----	Manipulation failure
black sesame seeds	----	Manipulation failure
banana	----	Manipulation failure
loose-leaf herbal tea jar	black chair	Success
red pencil sharpener	----	Navigation failure
fast-food French fries container	blue shopping bag [metal drying rack]	Placing failure
milk	plastic storage drawer unit	Success
socks[bed]	----	Navigation failure
purple gloves	----	Manipulation failure
target bag	cloth bin	Success
muffin	grey bed	Success
tissue paper	table	Success
grey eyeglass box	----	Manipulation failure
	Cleanup level: low	
pepsi	basket	Success
birdie	white drawer	Success
owl like wood carving	----	Navigation failure
red inhaler	plastic storage drawer unit	Success
black sesame seeds	bed	Success
loose-leaf herbal tea jar	table	Success
fast-food French fries container	chair	Success
milk	chair	Success
purple gloves	basket	Success
target bag	basket	Placing failure

Continued on the next page

Pick object	Place location	Result
muffin	table	Success
tissue paper	----	Manipulation failure
grey eyeglass box	----	Navigation failure
	Cleanup level: high	
pepsi	basket	Success
birdie	bed	Success
red inhaler	plastic storage drawer unit	Success
black sesame seeds	desk	Success
banana	----	Manipulation failure
loose-leaf herbal tea jar	----	Manipulation failure
milk	chair	Success
purple gloves	basket	Success
target bag	basket	Success
muffin	bed	Success
	Home 5	
	Cleanup level: none	
tiger balm topical ointment	----	Navigation failure
pink shampoo	trader joes shapping bag	Success
aveeno sunscreen protective lotion	trader joes shapping bag	Success
small yellow nozzle spray	----	Manipulation failure
black hair care spray	----	Manipulation failure
green hand sanitizer	----	Manipulation failure
white hand sanitizer	----	Navigation failure
white bowl [ketchup]	black sofa chair	Success
blue bowl	----	Manipulation failure
blue sponge	trader joes shapping bag	Success
ketchup	----	Manipulation failure
white salt	----	Manipulation failure
black pepper	black drawer	Success
blue bottle	----	Navigation failure
purple light bulb box	trader joes shopping bag	Success
white plastic bag	bed	Success
rag	white rack	Success
	Cleanup level: low	
pink shampoo	----	Navigation failure
aveeno sunscreen protective lotion	-----	Manipulation failure
small yellow nozzle spray	-----	Manipulation failure
white bowl [ketchup]	black sofa chair	Success
blue sponge	bed	Success
ketchup	trader joes shopping bag	Success
white salt	trader joes shopping bag	Success
black pepper	----	Navigation failure
blue bottle	black sofa chair	Success
purple light bulb box	-----	Manipulation failure
rag	white rack	Success
	Cleanup level: high	
pink shampoo	trader joes shopping bag	Success
green hand sanitizer	black sofa chair	Success
white bowl [ketchup]	-----	Manipulation failure
blue sponge	bed	Success
ketchup	black drawer	Success

Continued on the next page

Pick object	Place location	Result
white salt	white drawer	Success
purple light bulb box	trader joes shopping bag	Success
rag	black sofa chair	Success
	Home 6	
	Cleanup level: none	
translucent grey cup	----	Manipulation failure
green mouth spray box	stove	Success
green eyeglass container	chair	Success
blue bag	----	Manipulation failure
black burn ointment box	-----	Navigation failure
white vitamin bottle	----	Navigation failure
McDonald's paper bag	stove	Success
purple medicine packaging	chair	Success
grey rag	sink	Success
sparkling water can [sink]	countertop	Success
gold wrapped chocolate	----	Manipulation failure
lemon tea carton	table	Success
metallic golden beverage can	table	Success
red bottle	table	Success
tea milk bottle	----	Navigation failure
nyu water bottle [sink]	table	Success
white hand wash	-----	Navigation failure
	Cleanup level: low	
translucent grey cup	----	Navigation failure
green mouth spray box	----	Manipulation failure
blue bag	brown box	Success
black burn ointment box	brown box	Success
McDonald's paper bag	----	Navigation failure
grey rag	sink	Success
sparkling water can [sink]	chair	Success
lemon tea carton	stove	Success
metallic golden beverage can	----	Navigation failure
red bottle	brown box	Success
nyu water bottle [sink]	table	Success
white hand wash	sink	Success
	Cleanup level: high	
blue bag	brown box	Success
black burn ointment box	----	Manipulation failure
grey rag	sink	Success
sparkling water can [sink]	chair	Success
lemon tea carton	table	Success
metallic golden beverage can	stove	Success
red bottle	----	Navigation failure
nyu water bottle [sink]	----	Manipulation failure
white hand wash	----	Manipulation failure
	Home 7	
	Cleanup level: none	
blue plastic bag roll	-----	Navigation failure
green bag	basket[window]	Success
toy cactus	desk	Success
toy van	chair	Success

Continued on the next page

Pick object	Place location	Result
brown medical bandage	chair	Success
power adapter	_____	Navigation failure
red herbal tea	brown cardboard box	Success
apple juice box	brown cardboard box	Success
paper towel	blue cardboard box	Success
toy bear	bed blanket	Success
yellow ball	bed blanket	Success
black pants	basket[window]	Success
purple water bottle	desk	Success
blue eyeglass case	_____	Manipulation failure
brown toy monkey	_____	Navigation failure
blue hardware box [table]	blue cardboard box	Success
green zandu balm container	blue cardboard box	Success
	Cleanup level: low	
green bag	basket	Success
toy cactus	basket	Success
toy van	chair	Success
brown medical bandage	_____	Manipulation failure
red herbal tea	brown box	Success
apple juice box	brown box	Success
paper towel	basket	Success
toy bear	desk	Success
purple water bottle	desk	Success
blue eyeglass case	_____	Manipulation failure
green zandu balm container	blue cardboard box	Success
	Cleanup level: high	
green bag	stool [window]	Success
toy cactus	table	Success
toy van	white basket	Success
red herbal tea	brown cardboard box	Success
apple juice box	brown cardboard box	Success
paper towel	blue cardboard box	Success
toy bear	white basket	Success
yellow ball	bed	Success
purple water bottle	black tote bag	Success
green zandu balm container	blue cardboard box	Success
	Home 8	
	Cleanup level: none	
cyan air spray	brown shelf [sink]	Success
blue gloves	kitchen sink	Success
blue peanut butter	black stove	Success
nutella	table	Success
green bag	brown shelf [sink]	Success
green bandage box	trash can	Success
green detergent	kitchen sink	Success
black ‘red pepper sauce’	____	Manipulation failure
red bag	chair	Success
black bag	chair	Success
red spray [brown shelf]	kitchen countertop	Success
steel wool	_____	Manipulation failure
white aerosol	trash can	Success
white pretzel	black stove	Success

Continued on the next page

Pick object	Place location	Result
purple crisp	kitchen countertop	Success
plastic bowl	______	Manipulation failure
playing card	microwave	Success
	Cleanup level: low	
cyan air apray	chair	Success
blue gloves	sink	Success
blue peanut butter	____	Navigation failure
green bag	brown shelf	Success
green bandage box	brown shopping bag	Success
green detergent	microwave	Success
red bag	____	Manipulation failure
black bag	chair	Success
white aerosol	trash can	Success
white pretzel	black stove	Success
purple crisp	kitchen countertop	Success
plastic bowl	______	Manipulation failure
playing card	microwave	Success
	Cleanup level: high	
cyan air apray	brown shelf [sink]	Success
blue gloves	stove	Success
blue peanut butter	black stove	Success
green bag	brown shelf [sink]	Success
green bandage box	microwave	Success
green detergent	____	Manipulation failure
black bag	chair	Success
white aerosol	table	Success
purple crisp	chair	Success
playing card	microwave	Success
	Home 9	
	Cleanup level: none	
toy grapes	black laundry bag	Success
purple strap	_____	Manipulation failure
red foggy body spray	_____	Manipulation failure
arm smartphone holder	bed	Success
medicine bottle	_____	Manipulation failure
yogurt beverage	_____	Navigation failure
blue shaving cream can	____	Navigation failure
blue cup	table	Success
purple tape	_____	Manipulation failure
black shoe brush	_____	Navigation failure
fluffy headband	_____	Manipulation failure
black water bottle	brown shopping bag	Placing failure
yellow eyeglass case	black chair	Success
paper cup	_____	Manipulation failure
lotion pump	_____	Manipulation failure
nasal spray	_____	Manipulation failure
plastic bag	trash basket	Success
	Cleanup level: low	
toy grapes	_____	Manipulation failure
red foggy body spray	brown paper bag	Success
arm smartphone holder	brown paper bag	Success

Continued on the next page

Pick object	Place location	Result
yogurt beverage	desk	Success
blue shaving cream can	black bag	Success
blue cup	black chair	Success
black shoe brush	-----	Manipulation failure
fluffy headband	-----	Navigation failure
black water bottle	folded chair	Success
nasal spray	-----	Navigation failure
plastic bag	trash basket	Success
Cleanup level: high		
red foggy body spray	brown paper bag	Success
arm smartphone holder	-----	Manipulation failure
yogurt beverage	desk	Success
blue shaving cream can	black bag	Success
blue cup	black chair	Success
black water bottle	white bed	Success
nasal spray	folded chair	Success
plastic bag	trash basket	Success
Home 10		
Cleanup level: none		
grey toy dragon	bed	Success
purple body spray	----	Manipulation failure
hand sanitizer	shelf	Success
toy plant	bed [shelf]	Success
brown trail mix bag	----	Manipulation failure
hanging blue shirt	cloth bin	Success
white apple bag	----	Manipulation failure
white and pink powder bottle	table	Success
cough syrup bottle	shelf	Success
tangled ear phones	office chair	Success
red deodrant stick[table]	chair	Success
black body spray	chair	Success
hair treatment medicine bottle	----	Manipulation failure
green tea package	chair	Success
portable speaker [green tea package]	office chair	Success
wooden workout gripper	-----	Navigation failure
brown box	-----	Navigation failure
blue bulb adapter	office chair	Success
game controller	office chair	Success
Cleanup level: low		
grey toy dragon	orange bag	Success
purple body spray	table	Success
hand sanitizer	-----	Navigation failure
toy plant	bed	Success
brown trail mix bag	----	Manipulation failure
white and pink powder bottle	black chair [bed]	Success
cough syrup bottle	shelf [bed]	Success
red deodrant stick[table]	bed [rack]	Success
black body spray	rack [bed]	Placing failure
green tea package	orange bag	Success
brown box	black chair [bed]	Success
blue bulb adapter	-----	Manipulation failure

Continued on the next page

Pick object	Place location	Result
	Cleanup level: high	
purple body spray	orange bag	Success
toy plant	bed	Success
white and pink powder bottle	-----	Navigation failure
cough syrup bottle	shelf [bed]	Success
red deodrant stick[table]	-----	Navigation failure
black body spray	black chair	Success
green tea package	table	Success
blue bulb adapter	shelf	Success