# GRASP to GemBench Challenge at CVPR 2025 Workshop GRAIL

Ziyang Li[1], Yuting Mei[1], Sipeng Zheng[2], Qin Jin[1]✉

[1]AIM[3] Lab, School of Information,Renmin University of China; [2]BeingBeyond
ziyangli33@163.com,{meiyuting1004, qjn}@ruc.edu.cn, zhengsipeng27@gmail.com

## Abstract

*In this report, we present our winning solution to the GemBench Challenge at the CVPR 2025 Workshop GRAIL. Generalization remains a fundamental challenge in robotic manipulation, referring to the ability to perform manipulation tasks involving novel objects in previously unseen scenarios. Existing approaches often follow a planning-grounding-motion pipeline, where a language model decomposes high-level instructions, a vision-language model grounds the target objects, and a motion model executes the corresponding actions. However, a critical bottleneck in this pipeline lies in the mismatch between object grounding and motion control. Inaccurate grounding often results in partial or imprecise object representations, which ultimately degrades manipulation performance. To address this issue, we propose GRASP (Generalization and Robustness across Appearance and Semantic Perturbations), an enhanced planning-grounding-motion pipeline that improves manipulation performance under visual and semantic variability. Specifically, GRASP enhances robustness to partial observations via random view dropout and promotes generalization across diverse object appearances through re-coloring augmentation, a multi-view transformer, and a semantic-aware encoder. Our approach achieved $1^{st}$ place in the GemBench 2025 Challenge, outperforming the previous state-of-the-art methods by $5.9\%$ in average success rate in the public test set.*

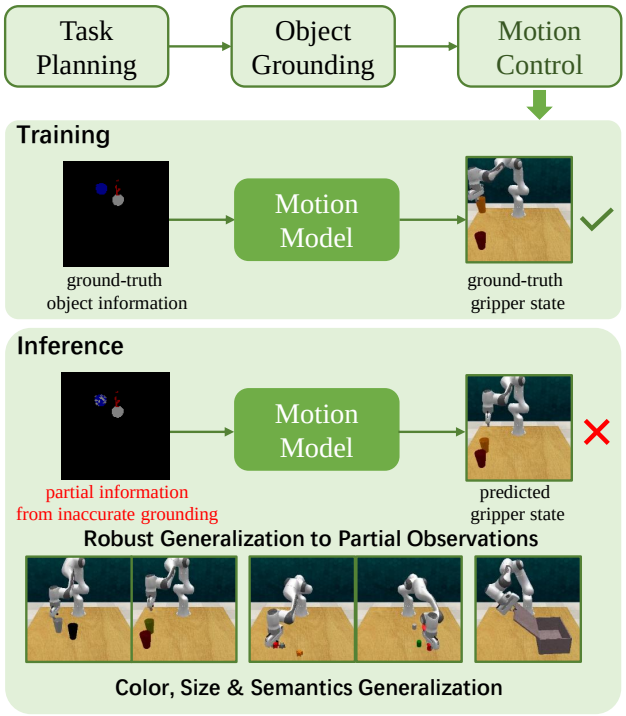


Figure 1. Our approach mainly focuses on two types of generalization: the robustness to partial observations, and the generalization across object color, size and semantics.

## 1. Introduction

Generalizable vision-language robotic manipulation remains a highly challenging task, as it requires language-conditioned policies to handle out-of-distribution scenarios, such as unseen object colors or novel objects. To thoroughly evaluate a model's generalization ability, the GRAIL GemBench Challenge [6] evaluates this across four difficulty levels: novel placements (Level 1), novel rigid objects(Level 2), novel articulated objects (Level 3), and long-horizon tasks (Level 4). The benchmark includes 16 training tasks (31 variations) and 44 testing tasks (92 variations), with 100 trajectory demonstrations per variation for training. Each variation presents a different configuration of the same task, differing in aspects such as object colors, positions, or orientations. The challenge provides two methods [6] as the baseline: i) 3D-LOTUS, which is a vision-language-action model that directly predict gripper action from visual and language inputs in an end-to-end manner, and ii) 3D-LOTUS++, an enhanced version that leverages a large language model [10] for task planning and vision-language models [18, 22] for object grounding, combined with the afore-mentioned mo-

✉ Corresponding author.

tion model in a three-step pipeline, significantly improving generalization in novel scenarios.

Although the grounding model [22] used in the pipeline baseline is capable of grounding unseen objects in novel scenarios, it often suffers from reduced accuracy in these cases. In addition, the motion model is trained with complete and accurate information about the target object, which becomes unavailable during inference due to inaccuracies in object grounding. In novel scenarios, inaccurate object grounding often results in partial observations—e.g., only the cup body of a cup—being fed to the motion model, which degrades the overall pipeline performance. Considering this, we employ a random view dropout mechanism during motion model training, enabling our model to adapt well to partial observations.

To further enhance generalization across variations in object color, size, and semantics—another key factor that causes failures when the motion model struggles to generalize to unseen objects-we introduce three key improvements to our pipeline: a re-coloring strategy, a two-stage multi-view transformer, and a semantic-aware vision encoder. As illustrated in Figure 1, these components work synergistically to address both partial observations and variations in object attributes. Our approach, **GRASP** (Generalization and Robustness across Appearance and Semantic Perturbations), demonstrates strong performance gains, achieving an average successful rate improvement of **5.9%** across all evaluation levels, with particularly significant improvements in levels including novel placements(L1), novel rigid objects(L2) and long-horizon tasks(L4).

## 2. Related Work

Vision-language robotic manipulation presents a significant challenge, requiring systems to determine appropriate robotic actions based on visual observations and natural language instructions. While several simulators and benchmarks [7, 15, 16, 21, 23, 30] have been developed to support this research area, RL-Bench [15] has emerged as the most widely adopted platform. The ability to generalize is particularly crucial in this domain, as policies must maintain robustness when deployed in environments that differ from their training conditions.

GemBench [6], built upon RL-Bench, specifically focuses on evaluating the generalization capabilities of vision-language manipulation policies. This benchmark features seven fundamental manipulation primitives and comprises 16 training tasks (with 31 variations) for policy deployment. For evaluation, it offers 44 test tasks (with 92 variations) organized into four progressively challenge levels: i) novel placements; ii) novel rigid objects varying in shape and color; iii novel articulated objects differing in instances, categories, and action parts; and iv) long-horizon tasks requiring extended multi-step planning and execution. This structure establishes GemBench as a comprehensive and fine-grained benchmark for assessing generalization in robotic manipulation systems.

Recent advances in generalizable vision-language robotic manipulation have introduced several promising approaches. With the impressive generalization capabilities demonstrated by foundation models [1, 3, 19, 24, 28], a growing number of works in robotic manipulation have incorporated them to enhance generalization, particularly in the domains of planning and object grounding [26]. [13] leverages large language models (LLMs) to decompose high-level tasks into actionable substeps. SayCan [2] further grounds these plans in the physical world by combining LLMs with the value functions of pre-trained skills. ViLa [12] integrates multimodal capabilities by using GPT-4V in place of standard language models, while CaP [20] prompts LLMs to generate executable code that invokes perception and control APIs. VoxPoser [14] proposes using LLMs to generate rich 3D voxel representations to generate execution trajectories. [29] integrates language-reasoning segmentation masks generated by foundation models into end-to-end policy models. FoundationGrasp [27] utilizes foundation models to learn generalizable task-oriented grasping skills by leveraging open-ended knowledge. The GemBench benchmark proposes 3D-LOTUS [6], a motion policy that processes point cloud representations to generate actions from visual observations and language instructions. Its successor, 3D-LOTUS++, enhances this approach through a modular pipeline integrating large language models for task planning and vision-language models for object grounding, significantly improving generalization across novel tasks, objects, and environments. Parallel developments include RVT [8], which employs a multi-view transformer with attention mechanisms to aggregate visual information from multiple viewpoints. This was further refined in RVT-2 [9] through a two-stage mechanism incorporating zoom-in views for enhanced spatial precision in fine-grained robotic manipulation. Another notable approach, SAM2Act [5], demonstrates the effectiveness of semantic segmentation models for robotic manipulation.

Inspired by these advances, our method adopts a pipeline architecture similar to 3D-LOTUS++ while introducing three key innovations: i) a random view dropout strategy to bridge the gap between object grounding and motion control, ii) a re-coloring module for color variation handling, and iii) an enhanced vision encoder combining a two-stage multi-view transformer with semantic segmentation capabilities. These components collectively address challenges in object grounding and motion control while improving generalization across variations in object attributes like color, size, and semantics.

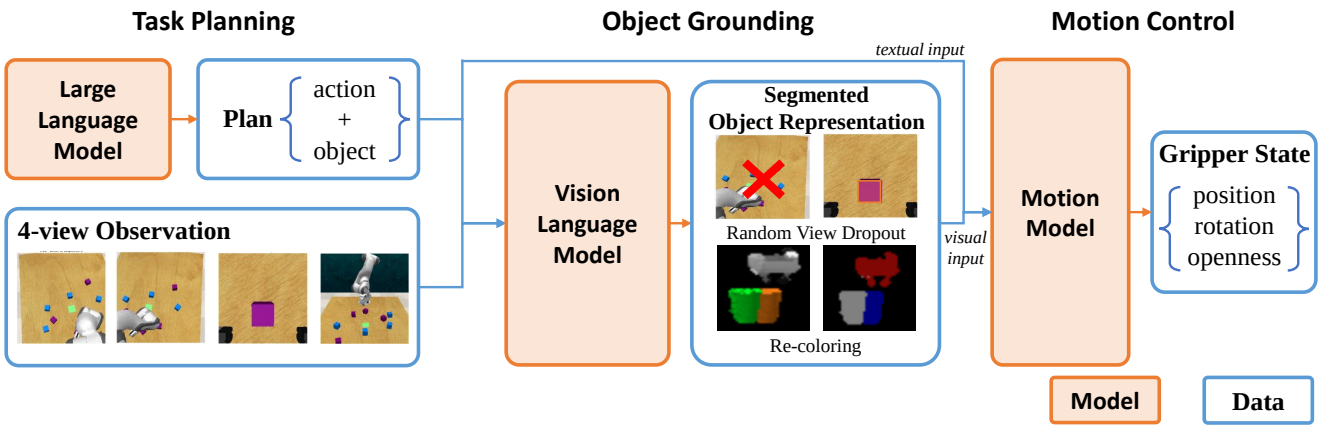


Figure 2. Illustration of our pipeline GRASP. Similar to 3D-LOTUS, it includes three steps: task planning, object grounding, and motion control.

# 3. GRASP

## 3.1. Overview

As illustrated in Figure 2, the pipeline of our method GRASPfollows a similar structure introduced by 3D-LOTUS++ [6], which finish a robotic tasks via three major steps: LLM Planning, Object Grounding, and Motion Control.

**LLM Planning.** This step is achieved based on a Large Language Model (LLM) [10], using a natural language instruction as input. The LLM decomposes the instruction into a sequence of sub-tasks each paired with the target object it operates on by selecting from a predefined set of actions.

**Object Grounding.** This step involves using a Vision-Language Model (VLM) to perform segmentation on RGB observations from four distinct viewpoints (global, left, right, and wrist) and merges the data across views to generate object-centric representations for target object matching. Specifically, We first use the object detection model Owlv2 [22] on four depth views to obtain candidate bounding boxes, along with CLIP-aligned embeddings for each detected object. For each bounding box, we extract the corresponding point cloud in world coordinates using depth information. We then merge identical objects across different views by computing the chamfer distance between point clouds and the cosine similarity between semantic embeddings. Two candidates are merged if both metrics fall below predefined thresholds. Finally, we encode the referenced object in the instruction using the CLIP text encoder and select the most similar object in the merged object set as the target object based on cosine similarity.

**Motion Control.** The final step involves a motion model that takes as input the LLM-predicted action and the target object point cloud grounded by the VLM. Using this information, the model predicts the appropriate gripper state, including position, rotation, and openness.

Table 1. The success rates of different pipeline components are compared when using either ground truth (GT) information or predictions from LLMs/VLMs [6]. Here, GT indicates the use of ground truth information, and LLM or VLM refers to predictions generated by large language models (LLMs) or vision-language models (VLMs).

| task plan | object ground | L1 | L2 | L3 | L4 |
|---|---|---|---|---|---|
| GT | GT | **92.6** | **80.1** | **47.8** | **31.5** |
| GT | VLM | 71.0 | 66.3 | 46.0 | 19.4 |
| LLM | VLM | 68.7 | 64.5 | 41.5 | 17.4 |

Next, we detail two challenges during this competition along with our corresponding solutions.

## 3.2. Key Challenge I: Generalization to Partial Observations

As shown in Table 1, the 3D-LOTUS++ [6] pipeline achieves significantly better performance when ground truth object grounding information is provided, compared to cases where such information is unavailable. Our analysis reveals that this performance gap stems from errors in the object grounding process during inference. Specifically, the grounding model may fail to associate the same object across different views, wrongly treating them as distinct instances. Consequently, the motion model receives incomplete object information, despite being trained on complete object data. This discrepancy between the training conditions and the inference-time inputs explains the motion model's lack of robustness to incomplete object representations.

We identify this discrepancy as the key factor limiting the pipeline's performance. To address this issue, we introduce a random view dropout training strategy for the motion model. During training, after generating the object-centric 3D point

representation from the grounding module, we randomly discard part of the object-centric point cloud, either by dropping points from randomly selected views or by applying a probabilistic dropout across all points. This approach emulates the partial and noisy observations encountered during inference, thereby improving the motion model's robustness to incomplete object representations. Experimental results confirm that our method significantly enhances the model's ability to generalize to partial observations, leading to significant performance gains across the pipeline.

### 3.3. Key Challenge II: Generalization to Diverse Color, Size, Semantics

To improve the pipeline's robustness to variations in object color, size, and semantics, we implement three key strategies:

**Re-coloring.** We introduce this strategy during both training and inference by adopting color assignments as follows: the object to be manipulated is colored blue, and the target object is colored yellow. In addition, the robotic arm is colored red and obstacles are colored gray. This enforced color scheme reduces appearance-based biases, directing the model's attention to object functionality rather than visual attributes.

**Two-stage Multi-view Transformer.** Our motion model, SAM2Act [5], utilizes a two-stage multi-view transformer same as RVT-2 [9]. This model consists of two stages: a coarse-grained stage and a fine-grained stage. The multi-view transformer processes input from multiple virtual views to predict the gripper's actions with increasing levels of detail, improving both accuracy and robustness across different object sizes and orientations.

**Semantic-aware Vision Encoder.** Our motion model [5] incorporates the semantic segmentation vision encoder from SAM2 [25] to extract high-quality visual features. Specifically, the object-centric point cloud are first rendered into virtual images from multiple viewpoints. These rendered views are then encoded using the vision encoder of SAM2 [25] to obtain semantic-aware visual features, which serve as input to the two-stage multi-view transformer. This design enhances the model's ability to understand object semantics and spatial configurations.

## 4. Experiment

### 4.1. Submission Results

We submitted five runs to the GemBench benchmark, with Table 2 comparing our best-performing run against previous state-of-the-art approaches. Our method demonstrates superior performance over both end-to-end methods [4–6, 9, 11, 17] and the modular baselines like 3D-LOTUS++ [6]. These results confirm that our proposed strategies effectively address two critical challenges: i) robust generalization to partial observations and ii) adaptation to variations in object color, size, and semantics.

Table 2. Comparison of our approach with previous works, including modular pipelines 3D-Lotus++ and vision-language manipulation policies. For SAM2Act [5], we evaluate performance using subtasks derived from ground truth planning as direct input to the motion model, bypassing the object grounding stage.

| Method | L1 | L2 | L3 | L4 | Avg. |
|---|---|---|---|---|---|
| Hiveformer [11] | 60.3 | 26.1 | 35.1 | 0.0 | 30.4 |
| PolarNet [4] | 77.7 | 37.1 | **38.5** | 0.1 | 38.4 |
| SAM2Act [5] | 83.5 | 48.2 | 37.1 | 0.9 | 42.4 |
| 3D diffuser actor [17] | 91.9 | 43.4 | 37.0 | 0.0 | 43.1 |
| RVT-2 [9] | 89.1 | 51.0 | 36.0 | 0.0 | 44.0 |
| 3D-LOTUS [6] | **94.3** | 49.9 | 38.1 | 0.3 | 45.7 |
| 3D-LOTUS++ [6] | 68.7 | 64.5 | 41.5 | 17.4 | 48.0 |
| Ours | 76.5 | **78.3** | 31.9 | **29.0** | **53.9** |

Table 3. Evaluation results of all runs on the public test set of GemBench[6]. Grayed-out numbers represent baseline results reported in [6].

| Motion Model | L1 | L2 | L3 | L4 | Avg. |
|---|---|---|---|---|---|
| 3D-LOTUS [6] | 94.3 | 49.9 | 38.1 | 0.3 | 45.7 |
| 3D-LOUTS++ [6] | 68.7 | 64.5 | 41.5 | 17.4 | 48.0 |
| GRASP(run1) | 78.9 | 73.3 | 39.1 | 17.0 | 52.1 |
| GRASP(run2) | 76.2 | 71.9 | 42.5 | 14.3 | 51.2 |
| GRASP(run3) | 82.3 | 64.1 | 45.6 | 9.4 | 50.4 |
| GRASP(run4) | 72.2 | 67.6 | 36.4 | 20.5 | 49.2 |
| GRASP(run5) | 76.5 | 78.3 | 31.9 | 29.0 | 53.9 |

Table 4. Evaluation results of all runs on the private test set of GemBench[6].

| Motion Model | L2 | L3 | L4 | Avg. |
|---|---|---|---|---|
| 3D-LOTUS [6] | 13.0 | 52.2 | 0.0 | 21.7 |
| 3D-LOUTS++ [6] | 58.0 | 41.1 | 17.2 | 38.8 |
| GRASP(run1) | 63.3 | 55.6 | 20.0 | 46.3 |
| GRASP(run2) | 63.0 | 56.1 | 15.0 | 44.7 |
| GRASP(run3) | 42.0 | 50.6 | 15.0 | 35.9 |
| GRASP(run4) | 62.3 | 55.6 | 16.1 | 44.7 |
| GRASP(run5) | 59.3 | 45.6 | 10.9 | 38.6 |

Table 3 and Table 4 show our submitted results on the public and private test sets, respectively. We consider two motion models in our pipeline: 3D-LOTUS [6] and SAM2Act [5]. 3D-LOTUS is a 3D transformer-based motion model, and we evaluate it under four training variations(run1-4) as follows: In the view dropout (run1) setting, we remove points from a random subset of input views during training. In the view dropout + RGB input (run2) setting, we additionally concatenate visual features extracted from the view image as input to the motion model. In the view dropout + object names (run3) variation, we further append the object

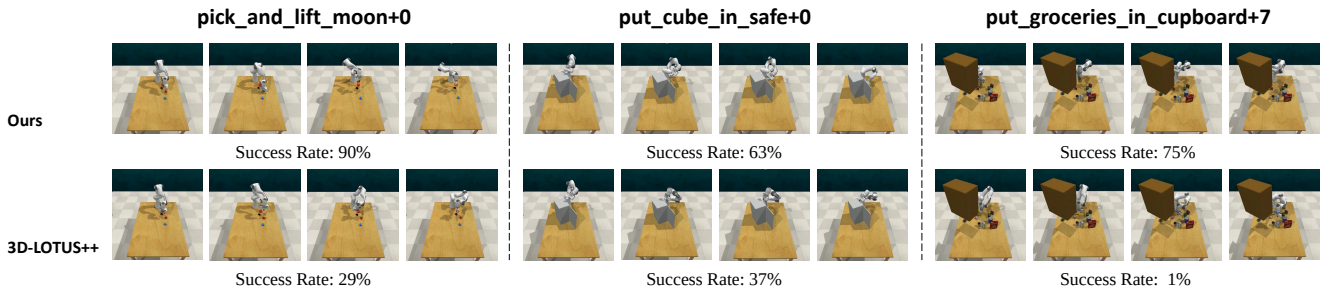


Figure 3. Qualitative comparisons on three challenging cases, where our approach demonstrates improved generalization compared to 3D-LOTUS++.

Table 5. Ablation study of "using the motion model alone" (with ground truth plans as input) vs. "using the full pipeline with different strategies". Here, RC refers to re-coloring and RVD refers to random view dropout.

| RC | RVD | L1 | L2 | L3 | L4 | Avg. |
|---|---|---|---|---|---|---|
| ✗ | ✗ | 83.5 | 48.2 | 37.1 | 0.9 | 42.4 |
| ✓ | ✗ | 75.0 | 66.2 | 34.0 | 21.3 | 50.7 |
| ✓ | ✓ | 76.5 | 78.3 | 31.9 | 29.0 | 53.9 |

name predicted by the LLM as a textual input. All three settings aim to enhance the model's robustness to missing view information by leveraging auxiliary modalities. In the random point dropout (run4) setting, instead of dropping entire views, we randomly remove individual points across the full point cloud with a fixed probability, simulating more diverse patterns of partial observations. SAM2Act [5] is a two-stage multi-view transformer enhanced with a semantic segmentation vision encoder. For this model, we apply both the random view dropout strategy and the re-coloring strategy to improve robustness and generalization (run5). Notably, all variants outperform the 3D-LOTUS++ baseline in the public test set, validating our methodological improvements.

### 4.2. Ablation Study

We conduct an ablation study with SAM2Act baseline [5] as shown in Table 5. Specifically, we compare SAM2Act [5] using ground truth task plans as step-by-step textual inputs against our three-step pipeline setup. For the pipeline, we compare the same configuration as 3D-LOTUS++ [6] only replacing the motion model with SAM2Act with pipeline equipped with our proposed generalization strategies—random view dropout and re-coloring—applied during training. The results show a significant performance improvement, confirming that our proposed methods effectively enhance both generalization capability and robustness.

### 4.3. Qualitative Analysis

Figure 3 provides a qualitative comparison between our method and 3D-LOTUS++ across challenging scenarios. For example, our approach demonstrates better generalization by successfully handling: i) grasping a moon-shaped cube, ii) precisely placing a cube into the correct layer, and iii) manipulating a non-standard grocery item. These results not only showcase our method's robustness with novel object shapes and complex tasks, but also correlate with the quantitative improvements in task successful rates.

## 5. Conclusion

In the GemBench Challenge, we tackle two key generalization issues: i) robustness to partial observations and ii) adaption to variations in object color, size, and semantics. Our solution integrates three innovations: a random view dropout training strategy, a standardized re-coloring approach, and a unified three-step pipeline featuring a two-stage multi-view transformer as motion model, and a semantic-aware vision encoder. This integrated approach achieved state-of-the-art performance on the benchmark. Looking ahead, we plan to optimize inference efficiency and enhance robustness in dynamic real-world scenarios to enable practical deployment.

## References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2

[3] 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. *arXiv preprint arXiv:2307.15818*, 2023. 2

[4] Shizhe Chen, Ricardo Garcia Pinel, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. In *Conference on Robot Learning*, pages 1761–1781. PMLR, 2023. 4

[5] Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation. *arXiv preprint arXiv:2501.18564*, 2025. 2, 4, 5

[6] Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. *arXiv preprint arXiv:2410.01345*, 2024. 1, 2, 3, 4, 5

[7] Ran Gong, Jiangyong Huang, Yizhou Zhao, Haoran Geng, Xiaofeng Gao, Qingyang Wu, Wensi Ai, Ziheng Zhou, Demetri Terzopoulos, Song-Chun Zhu, et al. Arnold: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20483–20495, 2023. 2

[8] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. *CoRL*, 2023. 2

[9] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt2: Learning precise manipulation from few demonstrations. *RSS*, 2024. 2, 4

[10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024. 1, 3

[11] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In *Conference on Robot Learning*, pages 175–187. PMLR, 2023. 4

[12] Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*. 2

[13] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022. 2

[14] 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 *Conference on Robot Learning*, pages 540–562. PMLR, 2023. 2

[15] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 2

[16] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, pages 14975–15022, 2023. 2

[17] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *8th Annual Conference on Robot Learning*. 4

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 1

[19] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. In *ICLR*, 2024. 2

[20] 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 *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023. 2

[21] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. 2

[22] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023. 1, 2, 3

[23] Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint arXiv:2402.08191*, 2024. 2

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[25] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4

[26] Xiuchao Sui, Daiying Tian, Qi Sun, Ruirui Chen, Dongkyu Choi, Kenneth Kwok, and Soujanya Poria. From grounding to manipulation: Case studies of foundation model integration in embodied robotic systems. *arXiv preprint arXiv:2505.15685*, 2025. 2

[27] Chao Tang, Dehao Huang, Wenlong Dong, Ruinian Xu, and Hong Zhang. Foundationgrasp: Generalizable task-oriented grasping with foundation models. *IEEE Transactions on Automation Science and Engineering*, 2025. 2

[28] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis

Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. 2

[29] Jiange Yang, Wenhui Tan, Chuhao Jin, Keling Yao, Bei Liu, Jianlong Fu, Ruihua Song, Gangshan Wu, and Limin Wang. Transferring foundation models for generalizable robotic manipulation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1999–2010. IEEE, 2025. 2

[30] Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Wang. Vlmbench: A compositional benchmark for vision-and-language manipulation. *Advances in Neural Information Processing Systems*, 35:665–678, 2022. 2