MOKA: Open-Vocabulary Robotic Manipulation through Mark-Based Visual Prompting

Fangchen Liu* Kuan Fang* Pieter Abbeel Sergey Levine
Berkeley AI Research, UC Berkeley

https://moka-manipulation.github.io

***Abstract*—Open-vocabulary generalization requires robotic systems to perform tasks involving complex and diverse environments and task goals. While the recent advances in vision language models (VLMs) present unprecedented opportunities to solve unseen problems, how to utilize their emergent capabilities to control robots in the physical world remains an open question. In this paper, we present Marking Open-vocabulary Keypoint Affordances (MOKA), an approach that employs VLMs to solve robotic manipulation tasks specified by free-form language descriptions. At the heart of our approach is a compact point-based representation of affordance and motion that bridges the VLM's predictions on RGB images and the robot's motions in the physical world. By prompting a VLM pre-trained on Internet-scale data, our approach predicts the affordances and generates the corresponding motions by leveraging the concept understanding and commonsense knowledge from broad sources. To scaffold the VLM's reasoning in zero-shot, we propose a visual prompting technique that annotates marks on the images, converting the prediction of keypoints and waypoints into a series of visual question answering problems that are feasible for the VLM to solve. We evaluate and analyze MOKA's performance on a variety of manipulation tasks specified by free-form language descriptions, such as tool use, deformable body manipulation, and object rearrangement.**

I. Introduction

The pursuit of open-vocabulary generalization poses a major challenge for robotic systems: Solving tasks in unseen environments given new user commands necessitate methods that can deal with the vast diversity and complexity of the physical world. An appealing prospect for handling this challenge is to employ large pretrained models by encapsulating extensive prior knowledge from broad data and bringing it to bear on novel problems. Recent advances in large language models (LLMs) and vision-language models (VLMs) provide particularly promising tools in this regard, with their emergent and fast-growing conceptual understanding, commonsense knowledge, and reasoning abilities [8, 39, 40, 4, 7, 1, 41, 22, 42, 23]. However, existing large models pre-trained on Internet-scale data still lack the capabilities to understand 3D space, contact physics, and robotic control, not to mention the knowledge about the embodiment and environment dynamics in each specific scenario, creating a large gap between the promising trend in computer vision and natural language processing and applying them to robotics. It remains an open question how such tools can guide a robotic system to solve manipulation tasks by interacting with the physical world.

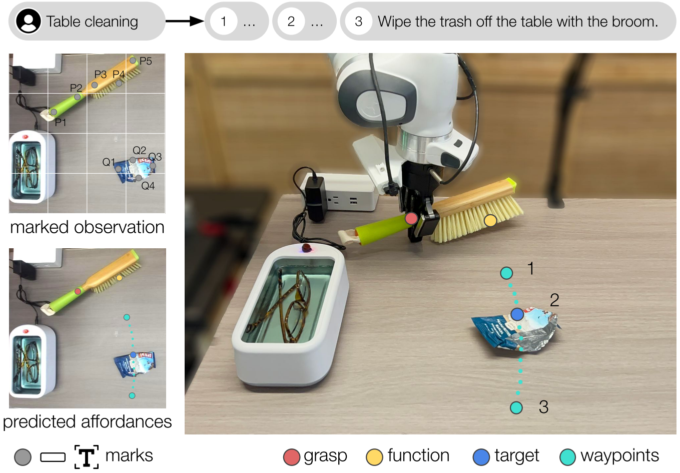


Fig. 1: To solve manipulation tasks with unseen objects and goals, MOKA employs a VLM to generate motions through a point-based affordance representation (plotted as colorful dots on the images). By annotating marks (e.g., candidate points, grids, and captions) on the observed 2D image, MOKA converts the motion generation problem into a series of visual question-answering problems that the VLM can solve.

Recently, a growing body of research has been dedicated to utilizing pre-trained large-scale models for robotic control. By incorporating broad knowledge, these approaches directly prompt or fine-tune the large models to generate plans [2, 16, 15, 5], rewards [28, 20, 30, 56], codes [25, 45, 49], etc. Despite the encouraging results they have demonstrated, these approaches are subject to notable limitations. Since the advances in LLMs precede VLMs, many previous approaches first process the raw sensory inputs to obtain the language description of the environment and then query LLMs to perform reasoning and planning in the language domain. However, relying solely on high-level language descriptions may overlook the nuanced visual details of environments and objects, which are vital for accurately completing tasks. In addition, existing approaches usually require non-trivial effort in designing in-context examples [17, 25, 45] to ensure LLMs can produce desired predictions on similar tasks. As a result, the tasks that can be solved by these approaches are largely constrained by such manual efforts.

In this work, we study how to effectively endow robots the ability to solve novel manipulation tasks specified by free-form language instructions using VLMs. Our key insight is to

*: Equal contribution. Correspondance to {fangchenliu, kuanfang}@berkeley.edu

find an intermediate affordance representation that connects the VLM's prediction on images with the robot's motion in the physical world. This affordance representation should satisfy two critical requirements. First, it should be feasible for the VLM to predict given the visual observation of the environment and task description. Second, it should compactly capture the information that well characterizes the important properties of the robot's motion, such that it can be easily executed on the robot.

To this end, we propose Marking Open-vocabulary Keypoint Affordances (MOKA), an approach that employs VLMs for robotic manipulation through mark-based visual prompting. As shown in Fig. 1, MOKA leverages a compact affordance representation consisting of a set of keypoints and waypoints, defined on open sets of objects and tasks. This point-based affordance representation is then used to specify the desired motion for the robot to solve the task. To generate the motions given the free-form language descriptions, MOKA uses hierarchical visual prompting to convert the affordance reasoning problem into a series of visual question answering problems. Drawing inspirations from recent advances in visual question-answering [51], we use mark-based visual prompting to enable the VLM to attend to the important visual cues in the observation image and further simplify the point generating problem into multiple choice questions. As shown in the top-left part of Fig. 1, we plot the keypoints on the image, and query the VLM to select the keypoints that result in the desired motion. The predicted keypoints and waypoints are used for specifying a trajectory through a waypoint-following motion, which can represent a wide range of manipulation skills such as picking, placing, pressing, tool-use, etc.

II. Marking Open-vocabulary Keypoint Affordances

We propose Marking Open-vocabulary Keypoint Affordances (MOKA), an approach that leverages the emergent reasoning capability of Vision-Language Models (VLMs) to guide a low-level motion generator to solve unseen tasks specified by free-form language instructions. As shown in Fig. 2, MOKA uses a point-based affordance representation to connect the VLM's prediction on 2D images with the robot's motion in the physical world.

A. Motion with Point-based Affordances

To leverage VLMs for solving open-vocabulary manipulation tasks, there needs to be an interface that connects the inputs and outputs of the VLM and the motions performed by the robot. To achieve this goal, we design an affordance representation defined on 2D images. Produced as the end result by the VLM, the affordance representation specifies the desired motion.

By extending the definitions in Manuelli et al. [32] and Qin et al. [38], we design a point-based affordance representation for a wide range of manipulation tasks. Instead of separately devising motion primitives for different pre-defined skills, we use a unified set of keypoints and waypoints to specify the motion. These points are predicted by VLMs on 2D images and converted to poses in the $\mathbf{SE}(3)$ space. Then a smooth motion trajectory is generated based on these poses. To perform the task, the robot gripper interacts with the environment by following the generated motion trajectory.

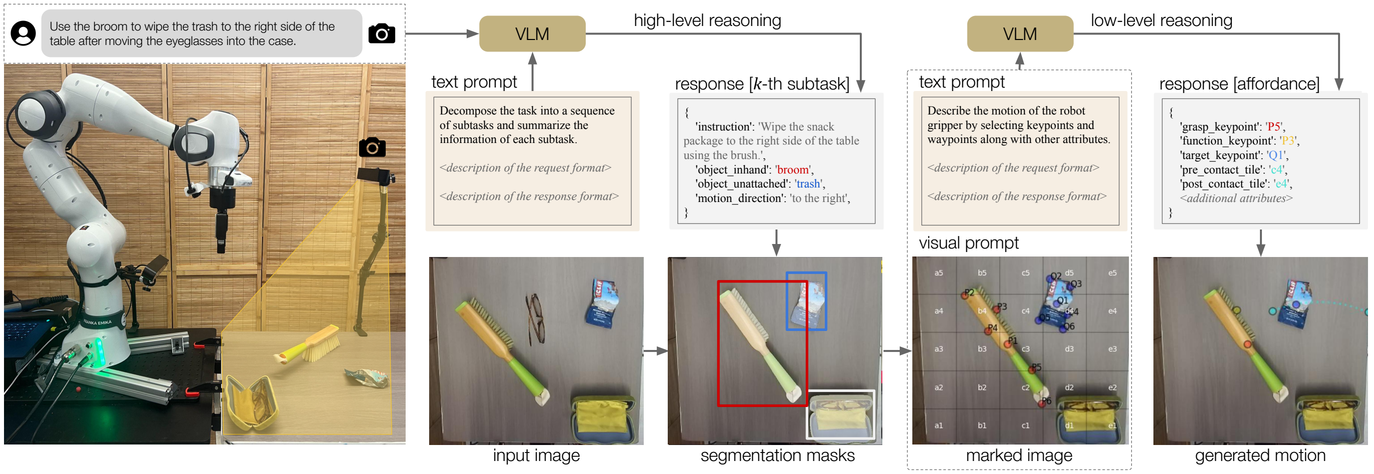
We specify the robot's motion in an object-centric manner as shown in Fig. 2. We would like this representation to be applicable to different types of interactions with objects in the environment. Therefore, we consider two types of objects, $o_{\text{in-hand}}$ (e.g., the broom) and $o_{\text{unattached}}$ (e.g., the trash), and specify the motion with a grasping phase and a manipulation phase. In the grasping phase, the robot reaches and grasps an object $o_{\text{in-hand}}$ from the environment. Then in the manipulation phase, the robot performs a motion and makes contact with another object $o_{\text{unattached}}$, either directly or using $o_{\text{in-hand}}$ as a tool. In some scenarios, only one of these two types of objects is interacted with by the robot, either $o_{\text{in-hand}}$ (e.g., unplugging a cable, opening a drawer) or $o_{\text{unattached}}$ (e.g., pressing a button), and one of the two phases can be skipped accordingly.

We now describe the definition of the keypoints and waypoints as well as how they are used to specify the motions in both phases. These points are illustrated in Fig. 2. Following the practice of Manuelli et al. [32] and Qin et al. [38], we use the **grasping keypoint** x_{grasp} to specify the position on $o_{\text{in-hand}}$ where the robot gripper should hold the object. If $o_{\text{in-hand}}$ is not involved in a task, the grasping phase will be skipped. For the manipulation phase, the robot's gripper follows a motion trajectory specified by an additional set of points. The **function keypoint** x_{function} specifies the part of $o_{\text{in-hand}}$ that will make contact with $o_{\text{unattached}}$ in the manipulation phase. If $o_{\text{in-hand}}$ is not specified, x_{function} will be on the robot gripper and the contact will directly be made between the robot and $o_{\text{unattached}}$. Correspondingly, the **target keypoint** x_{target} is the part of $o_{\text{unattached}}$ that will be contacted by x_{function} during the manipulation phase. We also introduce the **pre-contact waypoints** $x_{\text{pre-contact}}$ and the **post-contact waypoints** $x_{\text{post-contact}}$ defined in free space, which dictates the manipulation motion along with the keypoints defined on the objects.

During the manipulation phase, the robot moves the gripper such that x_{function} follows the path sequentially connecting the $x_{\text{pre-contact}}$, x_{target}, and $x_{\text{post-contact}}$. Besides following the path, we also require the robot gripper to follow the specified **grasping orientation** R_{grasp} and **manipulation orientation** $R_{\text{manipulate}}$ during the two phases respectively. To better illustrate the design of our point-based motion, we provide examples of the predicted point specifications and the resultant motions from our experiments in Appendix D. In MOKA, this set of keypoints and **additional attributes** (described in Sec. II-C) are summarized in a dictionary as the affordance representation (see Fig. 2).

B. Affordance Reasoning with Vision-Language Models

To predict the defined affordance representations, we employ the VLM $\mathcal{M}(\cdot)$, which is pre-trained on Internet-scale data for solving general visual question answering (VQA) problems. Using a hierarchical prompting framework as shown



Use the broom to wipe the trash to the right side of the table after moving the eyeglasses into the case.

Decompose the task into a sequence of subtasks and summarize the information of each subtask.

<description of the request format>

<description of the response format>

```
{
   'instruction': 'Wipe the snack package to the right side of the table using the brush.',
   'object_inhand': 'broom',
   'object_unattached': 'trash',
   'motion_direction': 'to the right',
}
```

Describe the motion of the robot gripper by selecting keypoints and waypoints along with other attributes.

<description of the request format>

<description of the response format>

```
{
   'grasp_keypoint': 'P5',
   'function_keypoint': 'P3',
   'target_keypoint': 'Q1',
   'pre_contact_tile': 'c4',
   'post_contact_tile': 'e4',
   <additional attributes>
}
```

Fig. 2: **Overview of MOKA.** We propose a hierarchical approach to prompt the VLM to perform affordance reasoning. On the high-level, we query the VLM to decompose the free-form language description of the task into a sequence of subtasks and summarize the subtask information. On the low-level, the VLM is prompted to produce the keypoints and additional attributes.

in Fig. 2, MOKA converts this affordance reasoning problem into a series of VQA problems that are solvable by the pre-trained VLM.

The hierarchical prompting framework takes as input the free-form language description l of the task and an RGB image observation of the environment s_t. MOKA examines the initial observation s_0 and decomposes the task l into a sequence of subtasks using the VLM. For each of the subtasks, the VLM is asked to provide the summary of the subtask instruction, the description of the corresponding $o_{\text{in-hand}}$, $o_{\text{unattached}}$, as well as the description of the motion (e.g., ”from left to right”). On the low level, given the response from the high-level reasoning and the visual observation $s_{t(k)}$ at the beginning of k-th subtask (at the time step $t(k)$), the VLM is queried again with a different prompt to produce the affordance representation defined in Sec. II-A. In the remainder of this section, we will describe the input formats, output formats, and prompt designs that we use to instantiate this method. Further details, including the complete prompts, can be found in the Appendix C.

High-level reasoning. Given the initial observation s_0 and the language description l, we first query the VLM $\mathcal{M}$ with the language prompt p_{task} to produce the response y_{high}:

$$y_{\text{high}} = \mathcal{M}([p_{\text{high}}, l, s_0]). \tag{1}$$

The representation y_{high} is a string that contains structured information for the K subtasks that the VLM infers are needed to solve the task. We design the prompt so as to require the VLM to produce y_{task} as a list of dictionaries. As shown in Fig. 2, each dictionary contains the language description of a subtask (e.g., *"Wipe the snack package to the right side of the table using the broom."*), as well as detailed information to facilitate motion generation, including the description of $o_{\text{in-hand}}$ (e.g., *"broom"*), the object name of $o_{\text{unattached}}$ (e.g., *"snack package"*), and the description of the motion (e.g., *"from left to right"*). This high-level plan will be used as an intermediate result for producing the detailed affordance representation through the low-level reasoning with the VLM.

Low-level reasoning. Next, we prompt the VLM once again to produce the affordance representation defined in Sec. II-A as y^k_{low}, conditioning on the high-level representation y_{high} and the visual observation $s_{t(k)}$ at the beginning of the k-th subtask. Instead of directly predicting 3D coordinates on 2D images, which is challenging and even ill-defined, we query the VLM to output 2D coordinates on the images and deproject them back to the 3D space. The three keypoints $x_{\text{grasp}}, x_{\text{function}}$ and x_{target} are defined on the object surface, and thus we can compute the 3D coordinates using the corresponding depth value of the 2D location based on the RGB image and camera parameters. For the waypoints in free space, we query the VLM to predict the desired height in text. To produce such an affordance representation y^k_{motion}, we query the VLM again by

$$y^k_{\text{low}} = \mathcal{M}([p_{\text{low}}, y^k_{\text{task}}, f(s_{t(k)})]), \tag{2}$$

where y^k_{high} is the substring corresponding to the k-th subtask extracted from y_{high}, and $f(\cdot)$ is a function that process the raw visual observation $s_{t(k)}$. We will explain the motivation and detailed implementation of $f(\cdot)$ in the next section and Appendix C. Through our ablation study in Appendix D, the hierarchical prompting strategy is essential for VLM to successfully perform the affordance reasoning for solving the tasks.

C. Mark-Based Visual Prompting

To perform the low-level reasoning mentioned in the previous section, we need the VLM to generate keypoints and waypoints on 2D images in order to execute a specific motion for a subtask. Since VLMs are better at multiple-choice problems than directly producing continuous-valued locations, we employ a mark-based visual prompting strategy to extract the desired output from VLMs, which we will describe in this subsection.

	Table Wiping		Watch Cleaning		Gift Preparation		Laptop Packing	
Methods	Subtask I	Subtask II	Subtask I	Subtask II	Subtask I	Subtask II	Subtask I	Subtask II
Code-as-Policies [25]	0.7	0.6	0.6	1.0	1.0	0.7	0.4	0.8
VoxPoser [17]	0.6	0	0.6	0.8	1.0	0.6	0.5	0.8
MOKA (Ours)	0.6	0.6	0.7	1.0	1.0	0.7	0.5	0.8

TABLE I: Success rate of our method and baselines. Across 4 tasks, MOKA consistently achieves superior performances.

Inspired by Yang et al. [52], MOKA uses a set of marks as visual prompts to enable VLM to apply its reasoning capability to predict the point-based affordance representation as shown in Fig. 2. Consisting of dots, grids, and text notations annotated on the image observation, these marks play an important role in the reasoning process. Proposed by open-vocabulary object detection and segmentation algorithms, these marks facilitate visual reasoning by encouraging the VLM to attend to the target objects and other task-relevant information in the image. We annotate marks as candidate parts and regions for the VLM to choose the points from, converting the original problem of directly generating coordinates into multiple-choice questions, which is usually more tractable for existing VLMs.

To select keypoints, which are defined on the in-hand object $o_{\text{in-hand}}$ and the unattached object $o_{\text{unattached}}$ suggested by the high-level reasoning in Sec. II-B, we propose and plot candidate keypoints on these objects. Given the names of $o_{\text{in-hand}}$ and $o_{\text{unattached}}$, we first segment these two objects using GroundedSAM [43], which combines GroundingDINO [27] and SAM [58] to extract segmentation masks of objects specified by a text prompt. After we obtain the segmentation masks of $o_{\text{in-hand}}$ and $o_{\text{unattached}}$, we perform farthest point sampling [37] on the object contour to obtain K boundary points. Together with its geometric center, and overlay the $K + 1$ candidate keypoints on each object. Each candidate keypoint is assigned an index, which is annotated next to it as a reference. To avoid confusion, we use different colors for candidate keypoints on $o_{\text{in-hand}}$ and $o_{\text{unattached}}$ and use the caption in the format of P_i and Q_j respectively, where i and j are integers. More implementation details can be found in Appendix C-B.

Selecting waypoints in free space involves searching over a much larger region. Instead of directly sampling points in the entire workspace, we divide the observed RGB image into an $M \times n$ grid, where m and n are integers. Both m and n are set to 5 for our evaluation tasks. The VLM is prompted to choose the tiles in which the pre-contact and post-contact keypoints are supposed to locate in and then the exact waypoints are sampled uniformly within the tile. For this purpose, we overlay the grid along with the name of each tile on the image. The tile names follow chess notation, which uses letters to specify the columns and integers for the rows.

III. Experiments

A. Experimental Setup

We compare MOKA with **Code-as-Policies** [25] and **VoxPoser** [17], two baselines that also enable zero-shot execution of open-vocabulary tasks:. Code-as-Policies provides a framework for language model-generated programs executed on robotic systems by prompting with code examples. For a fair comparison, we provide the two baselines with the task description in the code comments, with an additional 40 lines of code prompts providing example usage, as in the original implementation [25]. Similarly, VoxPoser [17] also provides code examples to large language models to build a 3D voxel map of value functions. For VoxPoser, we reuse the example prompts and planning pipeline in the original implementation, and use the same hyper-parameters to create the voxel value map. For both baselines, we only adapt the perception modules for fair comparisons, while retaining the functionality of the other components.

B. Evaluation

Our quantitative evaluation results across 4 tasks are illustrated in Tab. I. For each task, we report the number of successes out of 10 trials. As shown in the the table, MOKA achieves state-of-the-art performance at each subtask of the 4 tasks (totally 8 subtasks), with consistent improvements using in-context learning. On most of the tasks, VoxPoser [17] has similar performance with MOKA (zero-shot), except for subtask 2 of table wiping (which is a tool-use task). Additionally, the task success rates can be sensitive to the resolution of the voxel map, which requires some hyperparameter tuning. Unlike the baselines, MOKA can work well without example prompts. For more qualitative results, please refer to Appendix D-A.

IV. Conclusion and Discussion

In this paper, we proposed MOKA, a simple and effective visual prompting method that leverages VLMs for robot manipulation. By representing manipulation tasks with point-based affordances, we convert the motion generation process to a visual question-answering problem that VLMs can solve. MOKA provides a general and flexible framework that can intuitively and effectively harness VLM to generate point-based motion for a wide range of open-vocabulary tasks, while preserving the visual reasoning capabilities of VLMs. Our experiments demonstrate the effectiveness and robustness of MOKA across multiple tasks in both zero-shot and in-context learning manners. As far as we know, MOKA is the very first method that leverages visual prompting on VLMs for open-vocabulary robot manipulation.

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] 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.

[3] Aitor Aldoma, Federico Tombari, and Markus Vincze. Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes. In *2012 IEEE international conference on robotics and automation*, pages 1732–1739. IEEE, 2012.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] 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 *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11509–11522. IEEE, 2023.

[6] Changhyun Choi and Henrik I Christensen. Real-time 3d model-based tracking using edge and keypoint features for robotic manipulation. In *2010 IEEE International Conference on Robotics and Automation*, pages 4048–4055. IEEE, 2010.

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24 (240):1–113, 2023.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[10] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23171–23181, 2023.

[11] James J Gibson. *The ecological approach to visual perception: classic edition*. Psychology press, 2014.

[12] Tucker Hermans, James M Rehg, and Aaron Bobick. Affordance prediction via learned object attributes. In *IEEE international conference on robotics and automation (ICRA): Workshop on semantic perception, mapping, and exploration*, pages 181–184. Citeseer, 2011.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[14] 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. *arXiv preprint arXiv:2311.17842*, 2023.

[15] 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.

[16] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[17] 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. *arXiv preprint arXiv:2307.05973*, 2023.

[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. *arXiv preprint arXiv:2304.02643*, 2023.

[19] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[20] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. *arXiv preprint arXiv:2303.00001*, 2023.

[21] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023.

[22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[24] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded

language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

[25] 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.

[26] Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2motion: From natural language instructions to feasible plans. *arXiv preprint arXiv:2303.12153*, 2023.

[27] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[28] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

[29] 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. *arXiv preprint arXiv:1703.09312*, 2017.

[30] Parsa Mahmoudieh, Deepak Pathak, and Trevor Darrell. Zero-shot reward specification via grounded natural language. In *International Conference on Machine Learning*, pages 14743–14752. PMLR, 2022.

[31] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*, pages 2308–2315. IEEE, 2010.

[32] Lucas Manuelli, Wei Gao, Peter R. Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *International Symposium of Robotics Research*, 2019. URL https://api.semanticscholar.org/CorpusID:80628296.

[33] Stephen Miller, Mario Fritz, Trevor Darrell, and Pieter Abbeel. Parametrized shape models for clothing. In *2011 IEEE International Conference on Robotics and Automation*, pages 4861–4868. IEEE, 2011.

[34] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022.

[35] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. https://octo-models.github.io, 2023.

[36] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[38] Zengyi Qin, Kuan Fang, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Keto: Learning keypoint representations for tool manipulation, 2019.

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[40] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9, 2019.

[41] 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.

[42] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[43] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[44] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023.

[45] 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 *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.

[46] Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Brianna Zitkovich, Fei Xia, Chelsea Finn, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

[47] Jie Sun, Joshua L Moore, Aaron Bobick, and James M

Rehg. Learning visual object categories for robot affordance prediction. *The International Journal of Robotics Research*, 29(2-3):174–197, 2010.

[48] Jur Van Den Berg, Stephen Miller, Ken Goldberg, and Pieter Abbeel. Gravity-based robotic cloth folding. In *Algorithmic Foundations of Robotics IX*, pages 409–424. Springer, 2010.

[49] 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. *arXiv preprint arXiv:2305.16291*, 2023.

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[51] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

[52] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chun yue Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *ArXiv*, abs/2310.11441, 2023. URL https://api.semanticscholar.org/CorpusID:266149987.

[53] Lingfeng Yang, Yueze Wang, Xiang Li, Xinlong Wang, and Jian Yang. Fine-grained visual prompting. *arXiv preprint arXiv:2306.04356*, 2023.

[54] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023.

[55] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

[56] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.

[57] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

[58] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv preprint arXiv:2304.06718*, 2023.