2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

CoPa: General Robotic Manipulation through Spatial Constraints of Parts with Foundation Models

Haoxu Huang[1,2,3,4*], Fanqi Lin[1,2,4*], Yingdong Hu[1,2,4], Shengjie Wang[1,2,4], Yang Gao[1,2,4]

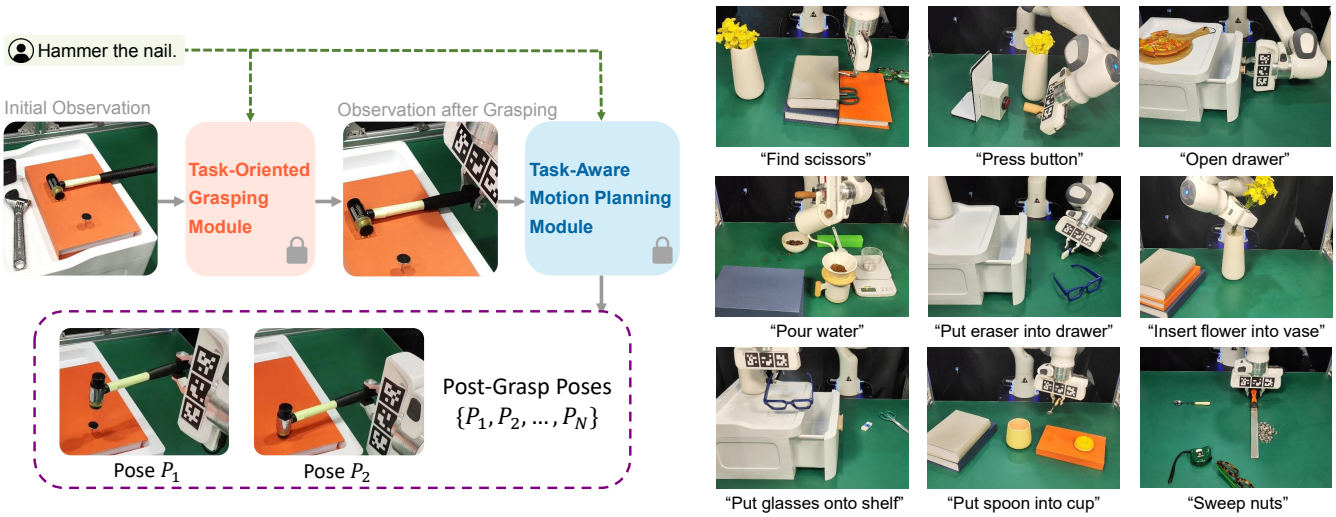

Fig. 1: **Overview.** We present CoPa, a novel framework that utilizes common sense knowledge embedded within VLMs for robotic low-level control. **Left.** Our pipeline. Given an instruction and scene observation, CoPa first generates a grasp pose through Task-Oriented Grasping Module (detailed in Fig. 3). Subsequently, a Task-Aware Motion Planning Module (detailed in Fig. 4) is utilized to obtain post-grasp poses. **Right.** Examples of real-world experiments. Boasting a fine-grained physical understanding of scenes, CoPa can generalize to open-world scenarios, handling open-set instructions and objects with minimal prompt engineering and without the need for additional training.

***Abstract*— Foundation models pre-trained on web-scale data are shown to encapsulate extensive world knowledge beneficial for robotic manipulation in the form of task planning. However, the actual physical implementation of these plans often relies on task-specific learning methods, which require significant data collection and struggle with generalizability. In this work, we introduce Robotic Manipulation through Spatial Constraints of Parts (CoPa), a novel framework that leverages the common sense knowledge embedded within foundation models to generate a sequence of 6-DoF end-effector poses for open-world robotic manipulation. Specifically, we decompose the manipulation process into two phases: task-oriented grasping and task-aware motion planning. In the task-oriented grasping phase, we employ foundation vision-language models (VLMs) to select the object's grasping part through a novel coarse-to-fine grounding mechanism. During the task-aware motion planning phase, VLMs are utilized again to identify the spatial geometry constraints of task-relevant object parts, which are then used to derive post-grasp poses. We also demonstrate how CoPa can be seamlessly integrated with existing robotic planning algorithms to accomplish complex, long-horizon tasks. Our comprehensive real-world experiments show that CoPa possesses a fine-grained physical understanding of scenes, capable of handling open-set instructions and objects with minimal prompt engineering and without additional training. Project page: copa-2024.github.io**

* The first two authors contributed equally.
[1] Institute of Interdisciplinary Information Sciences, Tsinghua University.
[2] Shanghai Qi Zhi Institute.
[3] Shanghai Jiao Tong University.
[4] Shanghai Artificial Intelligence Laboratory.

I. INTRODUCTION

Developing a general-purpose robot necessitates effective approaches in two critical areas: *(i)* high-level task planning, which determines what to do next, and *(ii)* low-level robotic control, focusing on the precise actuation of joints [1], [2]. The emergence of high-capacity foundation models [3], [4], pre-trained on extensive web-scale datasets, has inspired a surge of recent research efforts aimed at integrating these models into robotics [5], [6]. Nonetheless, these methods generally address only the "higher level" aspects of task planning [7]–[10]. In contrast, the prevailing approach for low-level control continues to revolve around crafting task-specific policies via diverse learning methods [11], [12]. Such policies, however, are brittle and prone to failure when encountering unseen scenarios [13]. Even the largest robotics models struggle outside environments they have previously encountered [14], [15].

The question then arises: what makes generalizable low-level robotic control so hard? We attempt to answer this question through the lens of human object manipulation. For instance, when an individual is tasked with hammering a nail, regardless of their familiarity with the specific hammer, they intuitively grasp it by the handle (instead of the head), adjust its orientation so the striking surface aligns with the nail, and then execute the strike. This process underscores

the importance of a fine-grained understanding of the physical properties of task-related objects, or more broadly, the extensive common sense knowledge of the world that facilitates generalizable object manipulation. Some pioneering works [16]–[18] have sought to leverage the rich semantic knowledge of Internet-scale foundation models to enhance low-level robotic control. Yet, these approaches are heavily dependent on intricate prompt engineering and suffer from a fundamental limitation: a *coarse* understanding of the scene, leading to failures in tasks requiring *fine-grained* physical understanding. Such a detailed understanding is essential for nearly all real-world robotic tasks of interest.

To endow robots with fine-grained physical understanding, we propose Robotic Manipulation through Spatial **Co**nstraints of **Pa**rts (**CoPa**), a novel framework that incorporates common sense knowledge embedded within foundation vision-language models (VLMs), such as GPT-4V, into the robotic manipulation tasks. Common sense knowledge incorporates understanding of physical properties and reasoning for complex tasks. We observe that most manipulation tasks require a part-level, fine-grained physical understanding of objects within the scene. Hence, we design a coarse-to-fine grounding module to identify task-relevant parts. Then, to leverage VLMs for aiding the robotic low-level control, it is necessary to design an interface that not only allows VLMs to reason in the form of language but also facilitates robot's object manipulation. Therefore, we propose utilizing *spatial constraints* as a bridge between VLMs and robots. Specifically, we utilize VLMs to generate the spatial constraints that task-relevant parts must meet to accomplish the task, and then employ a solver to determine the robot's poses based on these constraints. Finally, to ensure the precise execution of the robot's actions, transitions between adjacent poses are achieved through traditional motion planning methods.

We demonstrate that CoPa is capable of completing everyday manipulation tasks with a high success rate through extensive real-world experiments. Attributed to the innovative design of coarse-to-fine grounding and constraint generation module, CoPa possesses a profound physical understanding of the environment and can generate precise 6-Dof poses to complete complex manipulation tasks, significantly surpassing a strong baseline VoxPoser [16].

Our contributions are summarized as follows:

- We propose CoPa, a novel framework that utilizes the common sense knowledge of VLMs for low-level robotic control, which can handle open-set instructions and objects with minimal prompt engineering and without additional training.
- Through extensive real-world experiments, CoPa is demonstrated to possess the capability to complete manipulation tasks that require a fine-grained understanding of physical properties of task-relevant objects, significantly surpassing baselines.
- We show that CoPa can be seamlessly integrated with high-level planning methods to accomplish complex, long-horizon tasks (e.g. make pour-over coffee and set up romantic table).

II. RELATED WORK

Learning for Robotic Manipulation. Manipulation is a critical and challenging aspect in the robotic field. Numerous studies harness imitation learning (IL) from expert demonstrations to acquire manipulation skills [14], [15], [19]–[23]. Despite IL's conceptual simplicity and its notable success across a broad spectrum of real-world tasks, it struggles with out-of-distribution samples and demands considerable effort in collecting expert data. Reinforcement learning (RL) [12] emerges as another principal approach [7], [24]–[27], enabling robots to develop manipulation skills via trial-and-error interactions with their environment. However, RL's sample inefficiency limits its applicability in real-world settings, leading most robotic systems to rely on sim-to-real transfers [28]–[30]. Nonetheless, sim-to-real approaches necessitate the construction of specific simulators and confront the sim-to-real gap. Furthermore, policies learned via these end-to-end learning methods often lack generalization to new tasks. In contrast, by leveraging foundation models' common sense knowledge, our CoPa can generalize to open-world scenarios without additional training.

Foundation Models For Robotics. In recent years, foundation models have dramatically transformed the landscape of robotics [5]. Many works employ vision models, pre-trained on large-scale image datasets, to generate visual representations for visuomotor control tasks [21], [31]–[34]. Some other studies utilize foundation models for reward specification in reinforcement learning [35]–[40]. Furthermore, numerous studies have leveraged foundation models for robotic high-level planning, achieving remarkable success [7], [8], [10], [41]–[49]. There is also a body of works that employs foundation models for low-level control [14], [15], [22], [23]. Some works fine-tune vision-language models (VLMs) to directly output robot actions. However, such fine-tuning approaches require extensive amounts of expert data. To address this issue, Code as Policies [17] applies large language models (LLMs) to write code to control robots, and VoxPoser [16] generates robot trajectories by producing value maps based on foundation models. Nevertheless, these methods rely on complex prompt engineering and possess only a coarse understanding of the scene. In stark contrast, benefiting from the rational use of common sense knowledge within VLMs, our method exhibits a fine-grained understanding of scenarios and generalizes to open-world scenarios without additional training, requiring only minimal prompt engineering.

III. METHOD

In this section, we first introduce the formulation of manipulation tasks in Section III-A. Then, we describe two critical components within our framework — the task-oriented grasping in Section III-B and the task-aware motion planning in Section III-C.

A. Promblem Formulation

Most manipulation tasks can be decomposed into two phases: the initial grasp of the object and the subsequent motion required to complete the task. For example, opening

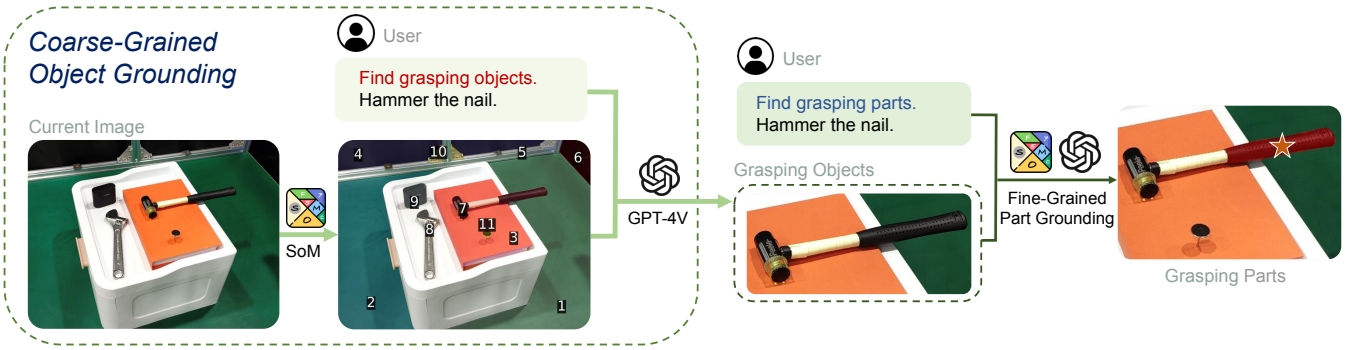

Fig. 2: Grounding Module. This module is utilized to identify the grasping part for task-oriented grasping or task-relevant parts for task-aware motion planning. The grounding process is divided into two stages: coarse-grained object grounding and fine-grained part grounding. Specifically, we first segment and label objects within the scene using SoM. Then, in conjunction with the instruction, we employ GPT-4V to select the grasping/task-relevant objects. Finally, similar fine-grained part grounding is applied to locate the specific grasping/task-relevant parts.

a drawer involves grasping the handle and pulling it in a straight line, while picking up a water glass requires first seizing the glass and then lifting it. Motivated by this observation, we structure our approach into two modules: **task-oriented grasping** and **task-aware motion planning**. Additionally, we posit that the execution of robotic tasks essentially entails generating a series of target poses for the robot's end-effector. The transition between adjacent target poses can be achieved through motion planning.

Given a language instruction l and the initial scene observation O_0 (RGB-D images), our objective in the task-oriented grasping module is to generate the appropriate grasp pose for the specified objects of interest. This process is represented as $P_0 = f(l, O_0)$. We denote the observation after the robot reaches P_0 as O_1. For the task-aware motion planning module, our goal is to derive a sequence of post-grasp poses, expressed as $g(l, O_1) \longrightarrow \{P_1, P_2, ..., P_N\}$, where N is the total number of poses required to complete the task. After acquiring the target poses, the robot's end-effector can reach these poses utilizing motion planning algorithms such as RRT* [50] and PRM* [51].

B. Task-Oriented Grasping

To generate the task-oriented grasp pose, our approach initially employs a grasping model to produce grasp pose proposals, and filter out the most feasible one through our novel grasping part grounding module. The entire process is depicted in Fig. 3.

Grasp Pose Proposals. We leverage a pre-trained grasping model for generating grasp pose proposals. To achieve this, we first convert RGB-D images into point clouds by back-projecting them into 3D space. These point clouds are then input into GraspNet [52], a model trained on a vast dataset comprising over one billion grasp poses. GraspNet outputs 6-DOF grasp candidates, including information on grasp point, width, height, depth, and a "graspness score," which indicates the likelihood of a successful grasp. However, given that GraspNet yields all potential grasps within a scene, it is necessary for us to employ a filtering mechanism that selects

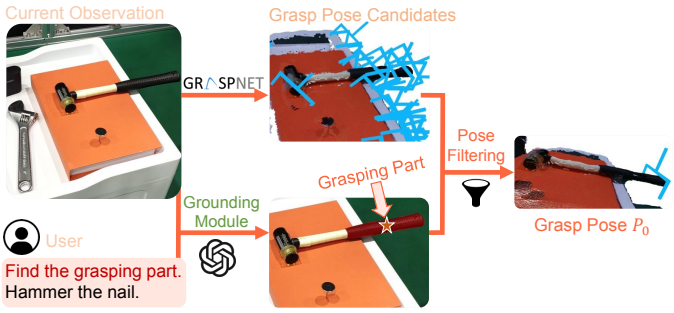

Fig. 3: Task-Oriented Grasping Module. This module is employed to generate grasp poses. Grasp pose candidates are generated from the scene point cloud using GraspNet. Concurrently, given the instruction and the scene image, the grasping part is identified by a grounding module (detailed in Fig. 2). Ultimately, the final grasp pose is selected by filtering candidates based on the grasping part mask and GraspNet scores.

the optimal grasp based on the specific task outlined by the language instruction.

Grasping Part Grounding. Humans grasp specific parts of an object corresponding to the intended use. For instance, when grasping a knife for cutting, we hold onto the handle rather than the blade; similarly, when picking up glasses, we grasp the frame instead of the lenses. This process essentially represents the application of common sense knowledge by humans. To mimic this ability, we utilize vision-language models (VLMs), such as GPT-4V [53], which incorporate vast amounts of common sense knowledge [10], [54], to identify the appropriate part of an object to grasp.

We employ a two-stage process to ground language instructions to the specific parts of objects intended for grasping: *coarse-grained object grounding* and *fine-grained part grounding*. The entire grounding process is shown in Fig. 2. At both stages, we incorporate a recent visual prompting mechanism known as Set-of-Mark (SoM) [55]. SoM leverages segmentation models to partition an image into distinct regions, assigning a numeric marker to each, significantly

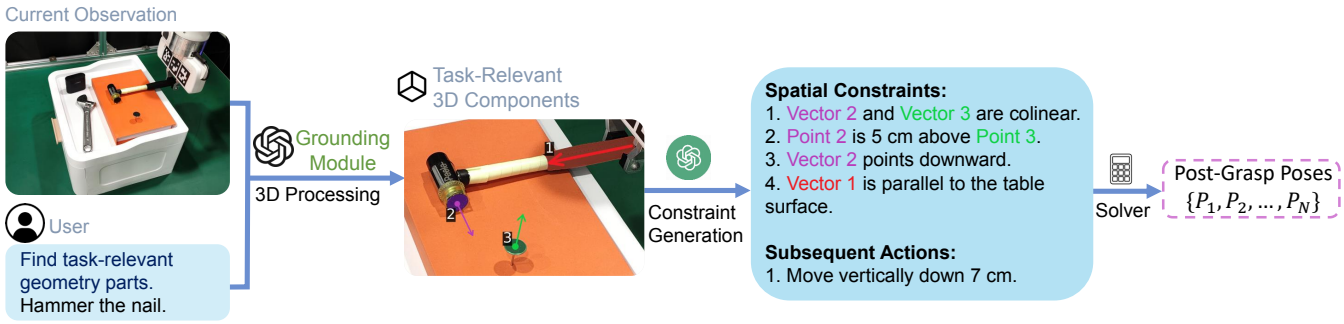

Fig. 4: Task-Aware Motion Planning Module. This module is used to obtain a series of post-grasp poses. Given the instruction and the current observation, we first employ a grounding module (detailed in Fig. 2) to identify task-relevant parts within the scene. Subsequently, these parts are modeled in 3D, and are then projected and annotated onto the scene image. Following this, VLMs are utilized to generate spatial constraints for these parts. Finally, a solver is applied to calculate the post-grasp poses based on these constraints.

boosting the visual grounding capabilities of VLMs. During the *coarse-grained object grounding* phase, SoM is utilized at the object level to detect and label all objects within the scene. Following this, VLMs are tasked with pinpointing the target object for grasping (e.g., a hammer), guided by the user's instructions. The selected object is then cropped from the image, upon which *fine-grained part grounding* is applied to determine the specific part of the object to be grasped (e.g., the handle of the hammer). This coarse-to-fine design endows our method with fine-grained physical understanding ability, enabling generalization across complex scenarios. Finally, we filter the grasp pose candidates, projecting all the grasp points onto the image and retaining only those within the grasping part mask. From these, the pose with the highest confidence scored by GraspNet is selected as the ultimate grasp pose P_0 for execution.

C. Task-Aware Motion Planning

After successfully executing task-oriented grasping, now we aim to obtain a series of post-grasp poses. We divide this step into three modules: task-relevant part grounding, manipulation constrains generation and target pose planning. The entire process is shown in Fig. 4.

Task-Relevant Part Grounding. Similar to the previous grasp part grounding module, we use *coarse-grained object grounding* and *fine-grained part grounding* to locate task-relevant parts. Here we need to identify multiple task-relevant parts (e.g. the hammer's striking surface, handle and the nail's surface). Additionally, we observe that numeric marks on the robotic arm may affect VLMs' selection, so we filter out the masks on the robotic arm (detailed in the Appendix).

Manipulation Constraints Generation. During the execution of tasks, task-relevant objects are often subject to various spatial geometric constraints. For instance, when charging a phone, the charger's connector must be aligned with the charging port; similarly, when capping a bottle, the lid must be positioned directly above the mouth of the bottle. These constraints inherently necessitate common sense knowledge, which includes a profound comprehension of the physical properties of objects. We aim to leverage VLMs to generate spatial geometric constraints for the object manipulated by the robot.

We first model identified task-relevant parts as simple geometric elements. Specifically, we represent slender parts (e.g. hammer handle) as vectors, while other parts are modeled as surfaces. For the parts modeled as vectors, we directly draw them on the scene image; for those modeled as surfaces, we ascertain their center points and normal vectors, which are then projected and marked on the 2D scene image. The annotated image is used as input for VLMs, which are prompted to generate spatial constraints for these geometric elements. We craft a set of descriptions for spatial constraints, such as collinearity between two vectors, perpendicularity between a vector and a surface, and so forth. We instruct the VLMs to first generate the constraints necessary for the first target pose, followed by the subsequent actions required after reaching that pose. Fig. 4 provides an illustrative example of this process. Implementation details of this process are provided in the Appendix.

Target Pose Planning. Upon obtaining manipulation constraints, we proceed to derive the sequence of post-grasp poses. This is equivalent to computing a sequence of SE(3) matrices such that, when applied to the parts of the object manipulated by the robotic arm, these parts satisfy the spatial geometric constraints. We operate under the assumption that the object part under manipulation and the robotic end-effector together constitute a rigid body. Consequently, these calculated SE(3) transformations can be directly applied to the robotic end-effector. We formalize the computation of the SE(3) matrix as a constrained optimization problem. Specifically, we compute a loss for each constraint, and then a nonlinear constraint solver is used to find the SE(3) matrix that minimizes the sum of these losses. Taking the constraint "Vector 2 points downward" from Fig. 4 as an example, the loss can be defined as the negative dot product of the normalized Vector 2 after SE(3) transformation and the vector $(0, 0, -1)$. After obtaining the first target pose, we solve for subsequent poses in alignment with the actions specified by VLMs. Concretely, we sequentially compute a new pose corresponding to each subsequent action. For example, for the action "Move vertically down 7 cm," we simply subtract 7 cm from the current pose on the z-

axis. This process results in a complete set of post-grasp poses $\{P_1, P_2, ..., P_N\}$, with the transitions between adjacent poses facilitated by motion planning algorithms. The detailed process for solving the SE(3) matrix and a comprehensive description of the subsequent actions can be found in the Appendix.

IV. Experiments

We first introduce the experimental setup in Section IV-A. Subsequently, we evaluate the performance of CoPa in real-world manipulation tasks in Section IV-B. Then we highlight CoPa's intriguing properties by comparing it with the baseline VoxPoser [16] in Section IV-C. We further present an ablation study to analyze the contribution of key modules within our framework in Section IV-D. Finally, we demonstrate that CoPa can be seamlessly integrated with high-level task planning methods to accomplish complex long-horizon tasks in Section IV-E.

A. Experimental Setup

Hardware. We set up a real-world tabletop environment. We use a Franka Emika Panda robot (a 7-DoF arm) and a 1-DoF parallel jaw gripper. For perception, we mount two RGB-D cameras (Intel RealSense D435) at two opposite ends of the table and calibrate them.

Tasks and Evaluations. We design 10 real-world manipulation tasks, each demanding a comprehensive understanding of the physical properties of objects. See Fig. 1 for illustrations of the tasks. For each task, we evaluate all methods across 10 different variations of the environment, which encompass alterations in object types and their arrangements. Detailed descriptions of the tasks are provided in the Appendix.

VLMs and Prompting. We employ GPT-4V from OpenAI API as the VLM. CoPa involves minimal few-shot prompts to aid VLMs in comprehending their roles. Additionally, the chain-of-thought technique [56] is utilized to facilitate a deeper understanding of the scene by VLMs. The full prompt is provided in the Appendix.

Baselines. We compare with Voxposer [16], a method capable of synthesizing closed-loop robot trajectories without necessitating additional training through the utilization of a series of foundational models. Following Huang et al [16], we employ GPT-4 from OpenAI API as the LLM, and utilize the open-vocabulary detector Owl-ViT [57] and Segment Anything [58] for perception.

B. CoPa for Real-World Manipulation

We study whether CoPa can generate robot trajectories to perform real-world manipulation tasks. The quantitative results are detailed in Table I, while the Appendix showcases additional qualitative outcomes, including visualizations of part grounding results and manipulation constraints. We find that CoPa achieves a remarkable success rate of 63% across ten different tasks, significantly outperforming the VoxPoser baseline and various ablation variants (detailed in the following sections). A key factor in CoPa's superior performance is its leverage of common sense knowledge embedded in VLMs, which enables a fine-grained understanding of objects' physical properties during both part grounding and constraint generation phases. For example, in the part grounding phase, CoPa accurately identifies the need to grasp the protective cover of an eraser in the `Put eraser on shelf` task, and recognizes the stem of the flower and the rim of the vase as critical parts in the `Insert flower into vase` task. During the constraint generation phase, CoPa comprehends that the spoon can be inserted vertically down into the cup in the `Put spoon into cup` task, and that the wooden stick needs to be aligned directly with the button in the `Press button` task.

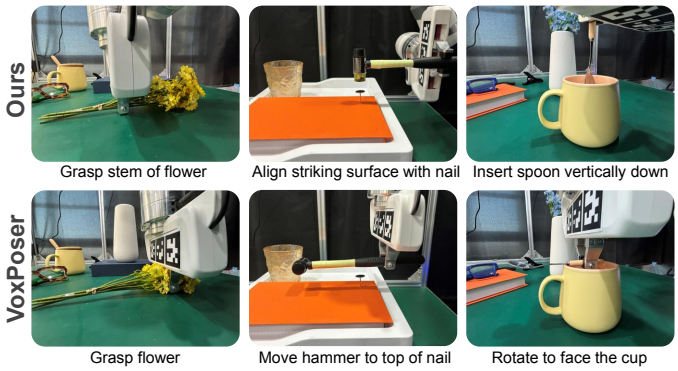

Fig. 5: **Comparison with VoxPoser.** We illustrate the execution of CoPa (top) and VoxPoser (bottom), demonstrating that CoPa possesses a fine-grained physical understanding of scenes and can effectively handle rotation DoF. The tasks from left to right are sequentially `Insert flower into vase`, `Hammer nail`, `Put spoon into cup`.

C. Understanding Properties of CoPa

In this section, we delve deeper into CoPa, shedding light on its intriguing properties through a comparative analysis with Voxposer, another method that utilizes the common sense knowledge embedded in foundation models to synthesize robot trajectories. CoPa exhibits significant advantages in the following three aspects:

Fine-Grained Physical Understanding. Many manipulation tasks require a nuanced physical understanding of the scene, which necessitates not only identifying object parts with fine granularity but also comprehending their intricate attributes. CoPa excels in this aspect, employing a coarse-to-fine part grounding module to select grasping/task-relevant object parts, and then utilizing VLMs to provide their spatial geometry constraints. In contrast, Voxposer only perceives objects in the scene as a whole. This coarse-grained level of comprehension often leads to failure in tasks that require precise operations. For instance, in the `Insert flower into vase` task (shown in Fig. 5 left), CoPa grasps the stem of the flower, whereas Voxposer seizes the petals. In the `Hammer nail` task (shown in Fig. 5 middle), CoPa orients the hammer to align precisely with the nail, while Voxposer overlooks this fine-grained physical constraint, treating the hammer as a single rigid body.

Simple Prompt Engineering. CoPa demonstrates remarkable generalizability across a wide range of scenarios with

Tasks	CoPa (Ours)	Voxposer	CoPa w/o foundation	CoPa w/o coarse-to-fine	CoPa w/o constraint
Hammer nail	30%	0%	0%	0%	10%
Find scissors	70%	50%	10%	70%	70%
Press button	80%	10%	10%	60%	20%
Open drawer	80%	40%	10%	70%	30%
Pour water	30%	0%	0%	10%	0%
Put eraser into drawer	80%	30%	30%	60%	80%
Insert flower into vase	70%	0%	0%	60%	0%
Put glasses onto shelf	60%	20%	30%	50%	60%
Put spoon into cup	60%	10%	0%	30%	30%
Sweep nuts	70%	20%	20%	50%	70%
Total	**63%**	18%	11%	46%	37%

TABLE I: Quantitative results in real-world experiments. CoPa successfully complete everyday manipulation tasks with a high success rate, demonstrating a profound physical understanding of scenes, significantly surpassing the baseline VoxPoser. Furthermore, we conduct ablation study to validate the importance of foundation models in our algorithm, as well as the design of coarse-to-fine grounding and constraint generation.

minimal prompt engineering. In our CoPa experiments, we employ just three examples to aid the VLMs in comprehending their roles. In contrast, Voxposer relies on highly complex prompts containing 85 hand-crafted examples. Its capability for reasoning predominantly stems from the provided prompts, thereby limiting its generalizability to new scenarios. When we attempt to simplify Voxposer's prompts, reducing the example count to three for each module, the system's performance drastically declines, resulting in almost complete failure across all evaluated tasks.

Handling Rotation DoF. Robotic manipulation requires not just the movement of the end-effector to a specified location but also the precise control of its rotation. For example, in the `Pour water` task, it is essential to rotate the kettle to a certain angle to enable the water to flow out through the spout. CoPa calculates the end-effector's 6-DoF pose by considering the spatial geometric constraints of key object parts within the scene, allowing for accurate and continuous control over rotation DoF. Conversely, Voxposer attempts to have LLMs directly specify the end-effector's rotation DoF based on simple examples in prompts, causing the output rotation values to be selected from a limited set of discrete options. This approach often overlooks the dynamic interactions and constraints between objects. For example, in the `Put spoon into cup` (shown in Fig. 5 right), CoPa rotates the spoon to a vertical orientation, whereas Voxposer positions the robot's end-effector to face the cup, resulting in a collision between the spoon and the cup.

D. Ablation Study

We next conduct a series of ablation studies to demonstrate the significance of the foundation model within our framework, as well as the design of coarse-to-fine grounding and constraint generation. The results are shown in Table I.

1) CoPa w/o foundation: We eliminate the use of foundation vision-language models (GPT-4V). Specifically, we substitute grasping/task-relevant parts grounding module with an open-vocabulary detector, Owl-ViT. Additionally, we remove the constraint generation phase and instead compute post-grasp poses in a predefined rule-based manner (detailed in the Appendix). The results, as presented in Table I, reveal that this approach encounters significant challenges, with a success rate of merely 11% across all the tasks. This underscores the crucial role of the common sense knowledge embedded within VLMs. For example, in the `Sweep nuts` task, it becomes challenging to determine which tool in the scene is suitable for sweeping without the aid of VLMs.

2) CoPa w/o coarse-to-fine: We eliminate the coarse-to-fine design in the grounding module, opting instead for direct utilization of fine-grained SoM and GPT-4V to select object parts within scenes. Experimental results indicate that removing coarse-to-fine design leads to a performance decline, especially in tasks where identifying important parts accurately is challenging. For example, in the `Hammer nail` tasks, the absence of the coarse-to-fine design makes this variant impossible to accurately identify the hammer's striking surface, leading to zero success rate for this task.

3) CoPa w/o constraint: In this ablation study, we have the VLMs directly output numerical values for the post-grasp poses of the end-effector, instead of the constraints that need to be satisfied by the object being manipulated. Experiments demonstrate that, for most manipulation tasks, directly deriving precise pose values from scene images is extremely challenging. For instance, in the `Pour water` task, it's almost impossible for this variant to generate precise pose values to tilt the kettle to the correct pose. In contrast, utilizing constraints given by VLMs to solve for post-grasp poses presents a more viable option.

E. Integration with High-Level Planning

High-level planning and low-level control are two critical and decoupled aspects of robotic task execution. Our low-level control framework can be seamlessly integrated with high-level planning methods to accomplish complex long-horizon tasks. We design two long-horizon tasks, `Make pour-over coffee` and `Set up romantic table`, to validate the effectiveness of this combination. Not only do these two tasks need to be accurately decomposed into reasonable and actionable steps, but the execution of each step requires a profound understanding of the physical properties of the task-relevant objects. Specifically, we employ

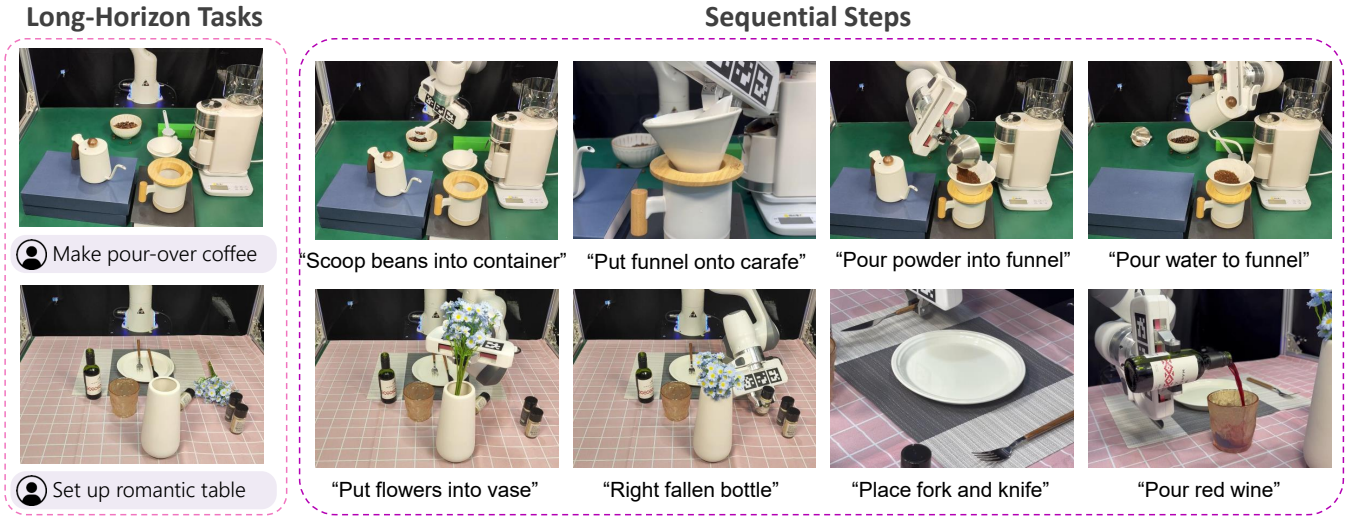

Fig. 6: **Intergration with High-Level Planning.** We show the execution process of two long-horizon tasks: `Make pour-over coffee` and `Set up romantic table`. We demonstrate that CoPa can be seamlessly integrated with high-level planning methods to accomplish complex long-horizon tasks.

VILA [10] as the high-level planning method to decompose the high-level instruction into a sequence of low-level control tasks. Subsequently, these low-level control tasks are executed sequentially using CoPa. Fig. 6 shows some environment rollouts. Experiments demonstrate that CoPa, combined with high-level planning methods, can effectively complete long-horizon tasks, showcasing the potential of this combination for real-world applications.

V. DISCUSSION & LIMITATIONS

In this work, we present CoPa, a novel framework that leverages the common sense knowledge of foundation vision-language models to generate pose sequences for robotic manipulation tasks. CoPa operates effectively with simple prompt engineering without requiring any training. Boasting a fine-grained physical understanding of scenes, CoPa can generalize to open-world scenarios, handling open-set instructions and objects. Moreover, CoPa can be naturally combined with high-level planning algorithms to accomplish complex, long-horizon tasks.

CoPa has a few limitations that future work can improve. First, CoPa's capability to process complex objects is constrained by its reliance on simplistic geometric elements such as surfaces and vectors. This can be improved by incorporating more geometric elements into our modeling process. Second, the VLMs currently in use are pre-trained on large-scale 2D images and lack a genuine grounding in the *3D physical world*. This limitation hampers their ability to perform accurate spatial reasoning. Integrating 3D inputs, like point clouds, into the training phase of VLMs may alleviate this challenge. Lastly, the existing VLMs produce only discrete textual outputs, whereas our framework essentially necessitates *continuous* output values, like the coordinates of object parts. The development of foundation models that incorporate these capabilities remains a highly anticipated advancement.

REFERENCES

[1] S. Cambon, R. Alami, and F. Gravot, "A hybrid approach to intricate motion, manipulation and task planning," *The International Journal of Robotics Research*, vol. 28, no. 1, pp. 104–126, 2009.

[2] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1470–1477.

[3] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[4] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[5] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, Z. Zhao, *et al.*, "Toward general-purpose robots via foundation models: A survey and meta-analysis," *arXiv preprint arXiv:2312.08782*, 2023.

[6] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023.

[7] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[8] W. Huang, F. Xia, D. Shah, D. Driess, A. Zeng, Y. Lu, P. Florence, I. Mordatch, S. Levine, K. Hausman, *et al.*, "Grounded decoding: Guiding text generation with grounded models for robot control," *arXiv preprint arXiv:2303.00855*, 2023.

[9] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[10] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao, "Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning," *arXiv preprint arXiv:2311.17842*, 2023.

[11] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[13] A. Xie, L. Lee, T. Xiao, and C. Finn, "Decomposing the generalization gap in imitation learning for visual robotic manipulation," *arXiv preprint arXiv:2307.03659*, 2023.

[14] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1:

Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.
[15] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.
[16] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.
[17] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.
[18] R. Mirjalili, M. Krawez, S. Silenzi, Y. Blei, and W. Burgard, "Langrasp: Using large language models for semantic object grasping," *arXiv preprint arXiv:2310.05239*, 2023.
[19] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.
[20] ——, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.
[21] T. Zhang, Y. Hu, H. Cui, H. Zhao, and Y. Gao, "A universal semantic-geometric representation for robotic manipulation," *arXiv preprint arXiv:2306.10474*, 2023.
[22] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.
[23] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, *et al.*, "Octo: An open-source generalist robot policy," 2023.
[24] Y. Liu, W. Dong, Y. Hu, C. Wen, Z.-H. Yin, C. Zhang, and Y. Gao, "Imitation learning from observation with automatic discount scheduling," *arXiv preprint arXiv:2310.07433*, 2023.
[25] W. Ye, Y. Zhang, M. Wang, S. Wang, X. Gu, P. Abbeel, and Y. Gao, "Foundation reinforcement learning: towards embodied generalist agents with foundation prior assistance," *arXiv preprint arXiv:2310.02635*, 2023.
[26] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.
[27] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.
[28] Z. Xu, Z. Xian, X. Lin, C. Chi, Z. Huang, C. Gan, and S. Song, "Roboninja: Learning an adaptive cutting policy for multi-material objects," *arXiv preprint arXiv:2302.11553*, 2023.
[29] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 734–743.
[30] R. Jeong, Y. Aytar, D. Khosid, Y. Zhou, J. Kay, T. Lampe, K. Bousmalis, and F. Nori, "Self-supervised sim-to-real adaptation for visual robotic manipulation," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 2718–2724.
[31] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.
[32] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in *Conference on Robot Learning*. PMLR, 2023, pp. 416–426.
[33] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022.
[34] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik, *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *arXiv preprint arXiv:2303.18240*, 2023.
[35] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," *arXiv preprint arXiv:2210.00030*, 2022.
[36] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.
[37] M. Alakuijala, G. Dulac-Arnold, J. Mairal, J. Ponce, and C. Schmid, "Learning reward functions for robotic manipulation by observing humans," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5006–5012.
[38] P. Mahmoudieh, D. Pathak, and T. Darrell, "Zero-shot reward specification via grounded natural language," in *International Conference on Machine Learning*. PMLR, 2022, pp. 14 743–14 752.
[39] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran, "Can foundation models perform zero-shot task specification for robot manipulation?" in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 893–905.
[40] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," *arXiv preprint arXiv:2306.00958*, 2023.
[41] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.
[42] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," *arXiv preprint arXiv:2309.02561*, 2023.
[43] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," *arXiv preprint arXiv:2303.06247*, 2023.
[44] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*. PMLR, 2022, pp. 9118–9147.
[45] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: From natural language instructions to feasible plans," *arXiv preprint arXiv:2303.12153*, 2023.
[46] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "Llm+ p: Empowering large language models with optimal planning proficiency," *arXiv preprint arXiv:2304.11477*, 2023.
[47] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, *et al.*, "Robots that ask for help: Uncertainty alignment for large language model planners," *arXiv preprint arXiv:2307.01928*, 2023.
[48] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2998–3009.
[49] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *arXiv preprint arXiv:2305.05658*, 2023.
[50] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
[51] L. Gang and J. Wang, "Prm path planning optimization algorithm research," *Wseas Transactions on Systems and control*, vol. 11, pp. 81–86, 2016.
[52] H.-S. Fang, C. Wang, M. Gou, and C. 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. 11 444–11 453.
[53] OpenAI, "Gpt-4v(ision) system card," https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023.
[54] H. Geng, S. Wei, C. Deng, B. Shen, H. Wang, and L. Guibas, "Sage: Bridging semantic and actionable parts for generalizable articulated-object manipulation under language instructions," *arXiv preprint arXiv:2312.01307*, 2023.
[55] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao, "Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v," *arXiv preprint arXiv:2310.11441*, 2023.
[56] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.
[57] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, *et al.*, "Simple open-vocabulary object detection with vision transformers. arxiv 2022," *arXiv preprint arXiv:2205.06230*.
[58] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.