Which objects help me to act effectively? Reasoning about physically-grounded affordances

Anne Kemmeren*[1], Gertjan Burghouts*[1], Michael van Bekkum[1], Wouter Meijer[1], Jelle van Mil[1]

***Abstract*—For effective interactions with the open world, robots should understand how interactions with known and novel objects help them towards their goal. A key aspect of this understanding lies in detecting an object's affordances, which represent the potential effects that can be achieved by manipulating the object in various ways. Our approach leverages a dialogue of large language models (LLMs) and vision-language models (VLMs) to achieve open-world affordance detection. Given open-vocabulary descriptions of intended actions and effects, the useful objects in the environment are found. By grounding our system in the physical world, we account for the robot's embodiment and the intrinsic properties of the objects it encounters. In our experiments, we have shown that our method produces tailored outputs based on different embodiments or intended effects. The method was able to select a useful object from a set of distractors. Finetuning the VLM for physical properties improved overall performance. These results underline the importance of grounding the affordance search in the physical world, by taking into account robot embodiment and the physical properties of objects.**

I. Introduction

To enable an intelligent robot to operate in the open-world, it needs to reason about how interacting with objects in the environment could contribute to its goal [9]. A crucial aspect of this capability is the robot's awareness of object affordances: understanding which actions can be executed on an object and what effects those actions produce. The encountered objects could be both novel or well-known.

Various previous works created models that endow robots with these reasoning skills by training on affordance datasets, where (regions of) objects are annotated with the actions that it allows [13, 22, 21]. However, the datasets are annotated from a human point-of-view, and these approaches thus fail to address how the embodiment of the robot affects affordances. A door handle can only be turned if the robot has the correct type of manipulator. Moreover, these datasets solely consider object-action pairs and do not take the effects into account. For example, in Padv2 both a surfboard and a bed have the affordance to *lie on* [21], but if the intended effect is to have the robot float on water only the surfboard is useful. Therefore, we believe that inclusion of the intended effect provides relevant task context, that is vital to distinguish which objects afford useful actions to the robot.

Embodied AI addresses both issues since it takes task context and robot embodiment into account. State-of-the-art models such as DreamerV3 [10], Octopus [19] and SayCan [1] have shown impressive performance to resolve what actions a robot should take to complete a (potentially complex) task, including situations where it needs to find and manipulate objects in the environment. However, these models consider only a very limited number of actions or skills that the robots can execute to keep the planning tractable. For example, DreamerV3, Octopus and SayCan would not be able to rotate the cap of a water bottle to retrieve water, since this specific grasp-and-rotation action is not in the list of possible actions.

Our approach to open-vocabulary affordance detection considers a much wider range of actions. Yet, it keeps planning tractable by only returning an object that affords the action if it (1) contributes to the intended effect, and (2) the object can be manipulated given the robot embodiment. Following the work on Socratic Models [4], we propose a dialogue between a Large Language Model (LLM) and Vision Language Model (VLM). Given an open-vocabulary action and task, the dialogue finds the objects in the given image that can help the robot reach its goal. When reasoning about relevant objects, we were inspired by [17] to have the dialogue explicitly take physical properties of the object into account. The novelty is that we prompt the LLM which objects provide the required affordance, while taking into account the limitations posed by the robot embodiment and the physical properties of object candidates as found by the VLM.

The contributions of this paper are as follows. (1) We develop a method for open-world affordance detection, where action, object and effect are all open-vocabulary. (2) We leverage out-of-the-box foundation models to reason about how embodiment of the robot and physical properties of the objects affect affordances. (3) We validate the efficacy for physical properties, show the limitations, and how finetuning can improve this. (4) In the experiments we show that the dialogue is able to more accurately select useful items from a set of distractors when the LLM takes physical properties of the object into account, and the VLM is finetuned on detecting objects with these properties, in an experiment with a real-life robot.

II. Related Work

Foundation models provide a generalized ability to reason about the physical world using everyday knowledge, understanding the behaviour and properties of objects (physical commonsense reasoning, [6]). However, physical grounding remains a challenge, while it is essential: a robot should know about the physical conditions of the objects of interest. For

*These two authors contributed equally

[1]The authors are with TNO, The Hague, Oude Waalsdorperweg 63, 2597 AK, Netherlands, {anne.kemmeren, gertjan.burghouts, michael.vanbekkum, wouter.meijer, jelle.vanmil}@tno.nl

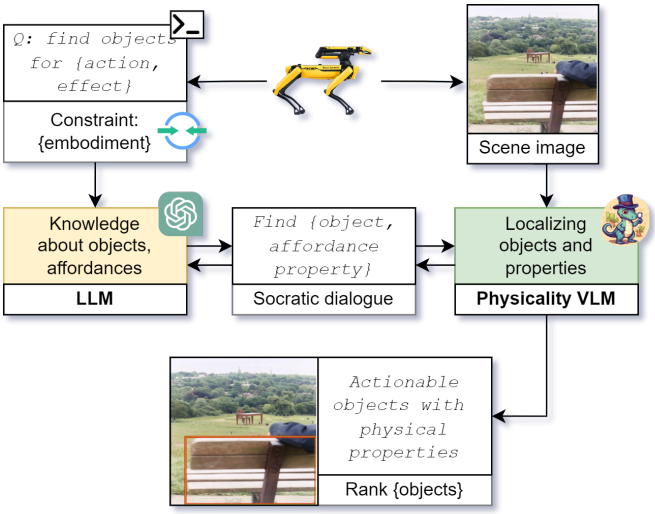

Fig. 1: A socratic dialogue enriched with a physicality-grounded VLM can reason about relevant objects for the given task and action.

instance, the weight of the objects determines which ones can be picked up, whereas the material determines how to manipulate them (e.g. amount of exerted force). For this purpose, PG-VLM [8] improved the prediction of physical properties of BLIP-2 [11]. BLIP2 is a foundation VLM, trained on a huge set of text-image pairs. However, it was shown that it could not estimate physical properties yet; the current very broad pretraining does not provide specialized knowledge about the object. PG-VLM incorporates such knowledge by instruction-tuning on physically grounded annotations.

LLMs specifically have previously displayed semantic reasoning capabilities [18]. Affordance learning usually takes place via detecting visual representations or imitation learning [2]. Most robotic foundation models are however tailored to a specific embodiment [7] and focus on movement or motion physics of objects [16].

Dialogue models or Socratic Models are employed to support exploratory thinking through questioning [4]. They allow back and forth interaction between foundation models as a dialogue in order to retrieve open-vocabulary affordances without fine-tuning [20].

The approach outlined in this paper also bolsters recent advancements in vision-language-action models such as RT-2 [3] and RT-2-X [5]. Where these developments enable robots to execute open-vocabulary actions, our work addresses where in the environment such open-vocabulary actions could be executed by leveraging open-vocabulary affordance detection.

III. Approach

A. Problem Definition

Given a particular goal and environment, the robot should be capable of interacting with objects in the environment, so that it reaches the goal. Therefore, we consider the problem of detecting the objects in the environment that afford the robot to take relevant actions towards its goal. As we operate in the open world, we require action and goal descriptions to be open-vocabulary.

B. Dialogue Overview

By chaining LLMs and VLMs, the specification of objects, actions and goals can all be done in free-form text. Figure 2 provides a detailed overview of the dialogue implementation. The LLM is prompted to acquire a set of objects that allow for the action or goal, while taking the constraints into account that are posed by the robot embodiment, mission requirements (e.g. safety) and properties of the objects (e.g. material). The VLM is queried to find the objects that have the right properties in the images captured by the robot's camera. The found objects are checked by the LLM regarding which properties they should have in order to be successful.

C. Dialogue Configuration

The LLM is configured with two variables to ground its reasoning in the real world. The first variable is information about robot embodiment and the second one is which physical object properties the LLM should consider to assess if an object will be relevant to the completion of the goal.

Robot: To take the robot embodiment into account, we provide a textual description of the robot. This allows the LLM to reason about the robot's capabilities and limitations when suggesting objects to interact with. This description is based on the specifications of the robot platform and includes properties such as robot type (e.g. wheeled, legged), dimensions (e.g. height, width), type of manipulator and weight:

$$\begin{aligned}\texttt{Robot} = \{&type \rightarrow quadruped,\\ &weight \rightarrow 50kg, \cdots,\\ &height \rightarrow 50cm\}\end{aligned} \tag{1}$$

Objective: The goal and requirements are specified:

$$\begin{aligned}\texttt{Requirements} = \{&goal \rightarrow climb\ on,\\ &conditions \rightarrow \{safe,\ reliable\}\}\end{aligned} \tag{2}$$

Note that the conditions can be defined in a soft manner, because the LLM can deal with such descriptions.

Objects: The physical properties of an object determine for a large part which interactions are allowed, e.g.: a robot might be able to stand on a metal box, but not on a paper box. We specifically ask the LLM to consider what object properties are relevant for the given action and goal, such that the VLMs can subsequently find only those object instances that have these properties. We provide a list of the properties and values that the VLMs on the robot could potentially detect, such as material types and colors.

$$\begin{aligned}\texttt{Properties} = \{&colors \rightarrow \{blue, \cdots, green\}, \cdots\\ &material \rightarrow \{plastic, \cdots, metal\}\}\end{aligned} \tag{3}$$

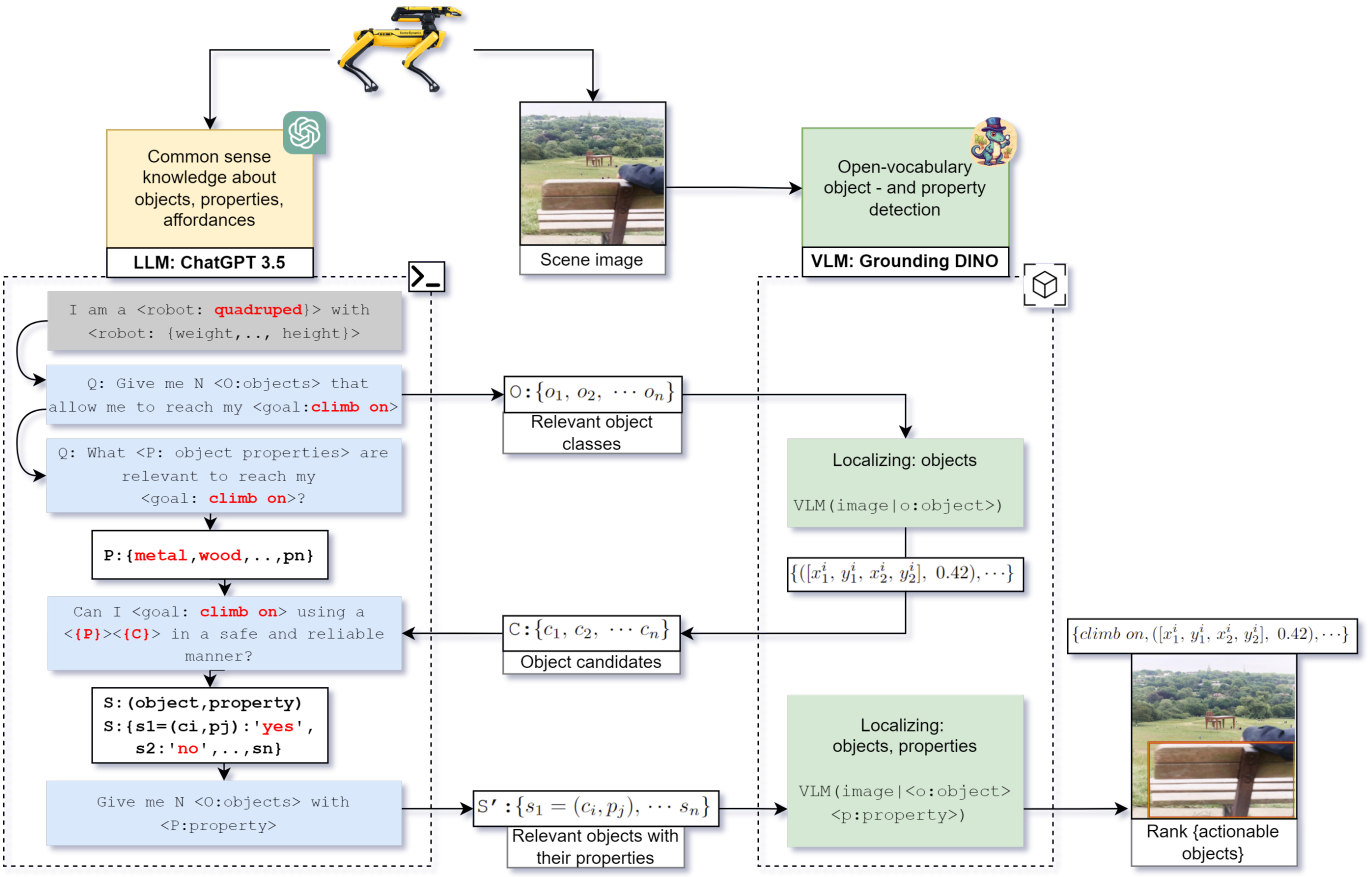

Fig. 2: The dialogue between an LLM (left) and VLM (right) reasons about what object in the given scene would give the quadruped robot the ability to climb to a better viewpoint.

TABLE I: **Intended effect**: For the same Action, but different intended Effects, our method suggests different objects.

Action *Effect*	Contain to get: liquids from A to B	groceries from A to B	Stand on to: increase robot's height	float on water
Bowl	✓	✓	-	-
Box, Bucket	✓	✓	✓	-
Blender, Can, Carton, Cup	✓	-	-	-
Jar, Kettle, Mug, Tray, Vase	✓	-	-	-
Bag, Belt	-	✓	-	-
Bench	-	✓	✓	-
Bottle, Ladder, Stool, Book	-	-	✓	-
Basket	-	✓	✓	✓

TABLE II: **Embodiment**; For a different embodiment or action, our method suggests other object properties.

	Small robot stands on	Large robot stands on	Large robot places a small object on
Basket	{plastic, metal}	{plastic, metal}	{plastic, metal}
Bench	{plastic, metal}	{plastic, metal}	{plastic, metal, wood, glass}
Box	{plastic}	{plastic}	{plastic, metal, wood, paper}
Book	{plastic, paper}	{}	{}
Ladder	{plastic, metal}	{plastic, metal}	{}
Stool	{plastic, metal, wood}	{plastic, metal}	{plastic, metal, wood, paper}

TABLE III: **Adaptation**: Adapting a VLM to the properties of objects is effective, increasing mAP.

VLM	Wood Basket	Stool	Ladder	Bench	Paper Box	Plastic Stool	Basket	Metal Ladder	Basket	Stool	Bench	Avg.
As-is	0.12	0.44	0.32	0.56	0.07	0.10	0.13	0.21	0.28	0.43	0.28	0.27
Physicality	0.53	0.66	0.37	0.66	0.46	0.13	0.25	0.34	0.32	0.43	0.48	**0.42**

TABLE IV: **Generalization**: The adapted VLM improves the prediction (measured by mAP) of properties of unseen objects (mAP).

VLM	Plastic Lamp	Crate	Hammer	Glass Lamp	Laptop	Wood Lamp	Crate	Metal Lamp	Crate	Laptop	Laptop	Avg.
As-is	0.45	0.05	0.15	0.12	0.29	0.11	0.02	0.42	0.11	0.17	0.43	0.21
Physicality	0.73	0.14	0.18	0.13	0.38	0.12	0.11	0.51	0.19	0.51	0.65	**0.33**

D. Reasoning

During runtime, the dialogue has as input a set of images that were collected of the environment, a (sub)goal specification and a desired action. Its output is a set of predictions of the objects that afford the desired action and contribute to reaching the goal.

The LLM is used in chain mode, such that earlier prompts and responses are stored as context. Our dialogue starts by informing the LLM of the context from the previous subsection:

$$\begin{aligned} &\texttt{I am a } \texttt{<}robot : type\}\texttt{> with} \\ &\texttt{<}robot : \{weight, \cdots, height\}\texttt{>} \end{aligned} \tag{4}$$

The LLM is prompted with the question which N objects can reach the goal. Its response is a text that includes a list of objects. The text is parsed to extract the names of the objects that are potentially suitable. The VLM is tasked to find these object names by prompting it with these names as labels. The threshold is set low (0.3) to avoid false negatives. For all object candidates provided by the VLM, the LLM is prompted whether the specific object can solve the task.

The LLM is queried for the object properties that are relevant to solve the task. For instance, object color is not relevant to climb on the object, but its material is. For the set of properties that is considered relevant, the specific instances are retrieved. E.g., object materials can be metal, wood, plastic, etc. The LLM is prompted which combinations of the object class with the relevant properties can solve the task. For instance, a prompt 'can the robot stand on a metal box in a safe and reliable manner?' More formally:

$$\begin{aligned} &\texttt{Can I <}goal\texttt{> using a <}property\texttt{> <}object\texttt{>} \\ &\qquad\qquad \texttt{in a <}conditions\texttt{> manner?} \end{aligned} \tag{5}$$

The response is parsed by extracting affirmative or negative words to understand if the object-property combination is suitable for the task at hand. The VLM is prompted for the set of suitable object-property combinations.

$$\begin{aligned} &\texttt{VLM(image} \mid \texttt{<}property\texttt{> <}object\texttt{>)} \rightarrow \\ &\qquad \{([x_1^i,\, y_1^i,\, x_2^i,\, y_2^i],\; c), \cdots\} \end{aligned} \tag{6}$$

Here are x_1^i, y_1^i the x, y of the upper-left position in the image, x_2^i, y_2^i the x, y of the lower-right position in the image, and c the confidence value. These predictions are the output of our dialogue.

IV. Experiments

We analyze the capabilities of our method, by answering the following questions:

1) For a given action, will it search for different objects when the intended effect is different?
2) For another embodiment of the robot, will it search for different objects with other properties?
3) Can a VLM designed for object detection be finetuned to estimate object properties? Does that generalize to unseen objects?
4) How well does our method find the right objects in the wild?

A. Setup

The LLM is ChatGPT 3.5 [14]. The VLM is Grounding DINO [12], because it can detect objects (i.e. localization). PG-BLIP [8] is also interesting, but it can only classify images (i.e. no localization) and it is a very large model which is disadvantage for deployment on a robot. For evaluation, we consider the PACO image dataset [15], because it has annotations of the objects, parts and their properties including materials. We also include an experiment with the SPOT robot in our Open-World Robotics lab.

B. Effect-specific Objects

For a given action, but for a different intended effect, our method suggests different objects. Table I shows the results after evaluating Equations 4 and 5 for the respective Actions and Effects in the table header (in a combined prompt of Action + Effect in Equation 5). When the desired effect is to get liquids from one location to another location, the intended action is contain, and suitable objects are a Bowl, Can or Vase. However, if the action is the same, but the intended effect is to bring groceries to another location, a Can is not suitable, while a Bag or Basket are suitable. Likewise, for a robot that is tasked to stand on something, the intended effect matters. There is a difference when the intended effect is to increase the robot's height (e.g. Bucket, Bench, Basket), or that the robot should float on water (only Basket). Our method can handle various intended effects.

C. Constraints of the Embodiment

For a robot with a different embodiment, our method yields objects with different properties. Table II shows the objects and their properties as suggested by our method, for respectively a small robot (height of 25 centimeters and 5 kilograms) and a large robot (height of 50 centimeters and 50 kilograms). This is to evaluate the effect of Equation 4 on Equation 5. The

method indicates that the two robots can stand on different objects with different properties. The small robot can stand on a Book, whereas the large robot cannot. The small robot can stand on a Wooden Stool, whereas the large robot can only stand on Metal and Plastic Stools. The intended action also matters: a small object can be placed on a Glass Bench and Paper Box, while both robots cannot stand on these objects.

D. Improving Detection of Object Properties

Given the importance of objects and their properties: how well can a VLM detect objects with specific properties? Table III shows the results for our VLM on the first row. The performance is not good: mAP=0.27. Especially the Paper Box (mAP=0.07), Wood Basket (mAP=0.12) and Plastic Stool (mAP=0.10) are hard to detect.

Finetuning is applied with the goal to improve the performance of estimating object properties. The subset of PACO as shown in Table III is leveraged for this purpose. We follow Grounding DINO's standard recipe for finetuning, i.e. 15 epochs with the provided learning rate and schedule. With finetuning, the results can be improved significantly, from mAP=0.27 to mAP=0.42 on average. Paper Box is improved from mAP=0.07 to mAP=0.46, whereas Plastic Stool is not improved much: mAP=0.10 to mAP=0.13. Plastic Stool is a rare object-material combination, which makes it hard to learn. Wood Basket is improved from mAP=0.12 to mAP=0.53.

E. Generalization to Unseen Object-Properties

The question is whether the newly learned object properties generalize to unseen objects. There is a performance gain for the unseen objects from Table IV. On average, the VLM performance of mAP=0.21 is increased to mAP=0.33. Some objects do not improve much, e.g. Plastic and Metal Crate; Wood and Glass Lamp. These are hard objects because they are respectively partially visible (crates often are in between other objects) and small. Paper Box, Wood Stool and Metal Ladder are improved by large margins, without having seen these object-material combinations during training.

F. Analysis of Results

We inspect the objects and properties for which the improvement is most significant. Figure 3 shows the largest gains, sorted from most to less gain. In all cases, the VLM as-is predicts the objects wrongly, mAP=0. The finetuned VLM with physical properties predicts the objects and their properties well, even in very challenging circumstances, e.g. the Metal Ladder on the back of the firetruck photographed under a shear viewing angle (Figure 3, right). The Wood Bench and the Paper Box (left) are small, whereas the Plastic Box on the motorcycle is in the midst of clutter.

Most object predictions have the correct object label, but a wrong material label. This is to be expected: it is easier to assess the object class than its properties. Also, the pretraining of most VLMs is focused on object classes, not on object properties. We inspect which object properties have improved most, see Figure 4. The Wood Bench is corrected to a Metal Bench (left) and vice versa (second column). Glass Basket is corrected to Metal Basket (fourth column) and Metal Ladder to Wood Ladder (right).

Figure 5 shows several examples of remaining errors. The Plastic Box (left) is probably correct; it appears to be a mislabeling in the groundtruth. The Wood Bench is mistaken for a Metal Bench (second image) and vice versa (third image). The prediction Wood Bench (third image) is actually a Metal Bench, but it is hard to distinguish, even for humans. The Metal Stool (fourth column) is a Wood Stool but it has metal legs.

G. Searching for the Right Objects

Our method is a dialogue between an LLM and a VLM to find the right objects with the desired properties to fulfil an action for an intended effect. We evaluate how well our method can find the desired combination of the object and the property. This is to validate the joint efficacy of Equations 4, 5 and 6.

If the desired object is Paper Box, we collect distractor images from PACO that contain Boxes with other properties than Paper, e.g. a Plastic Box. The objective is to find the Paper Box in the target image, in the midst of N images that contain other Boxes. We increase N progressively to assess how well our method can find the desired object when the task becomes increasingly difficult. This is to mimic that the robot's environment increases. We repeat this trial for all images and combinations from Table III and average the results. The results are shown in Figure 7, with the N distractor images on the x-axis and the rank at which the desired object is found on the y-axis (log scale). The relation between the amount of distractors and the rank is approximately linear, as expected. The blue line ('object VLM') is the performance when only the object class is taken into account. Evidently, this is not an adequate strategy: the objects with the right properties are found at ranks > 10, e.g. with 8 distractor images it is found around rank 70 on average. Our method takes the desired properties into account. Even with the VLM as-is (orange line), which is not optimal for predicting object properties, the ranking is significantly improved. With 8 distractor images, the rank improves from 70 to <9. It can be concluded that it is beneficial to take the object properties into account. When our method is combined with the finetuned VLM (green line), the average rank further decreases from 9 to 4.5. This means that the efficiency of finding the object with the right properties is further improved by a factor of 2 on average.

To find an object with the right property, is illustrated in Figure 6, for three examples (rows), respectively found at ranks 1 (optimal), 3 and 5. A Plastic Bench is found at rank 1 (top row). A Plastic Stool (middle row) is found at rank 3. At ranks 1 and 2, other Stools are found, which are not Plastic but Wood and Metal. A Wood Basket (bottom row) is found at rank 5. Again, the objects are all Baskets, but not the right property, e.g. a Plastic Basket.

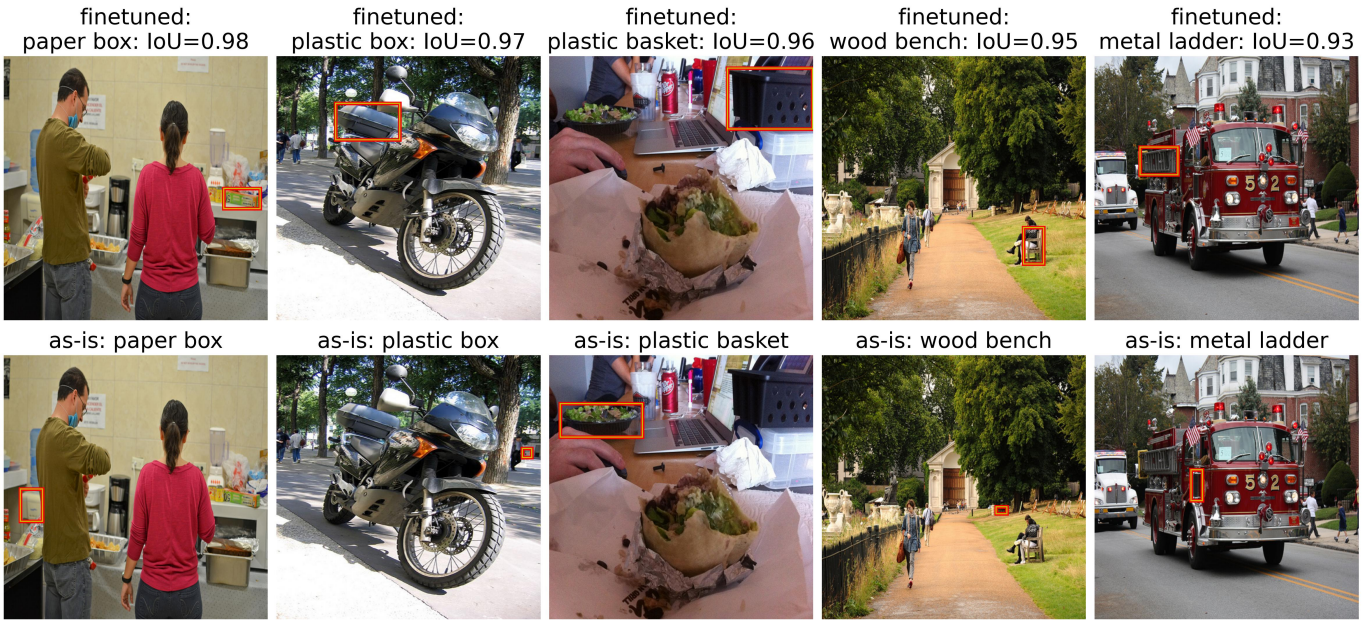

Fig. 3: **Improvements**: After adaptation, the VLM's prediction of objects and properties is improved (top row).

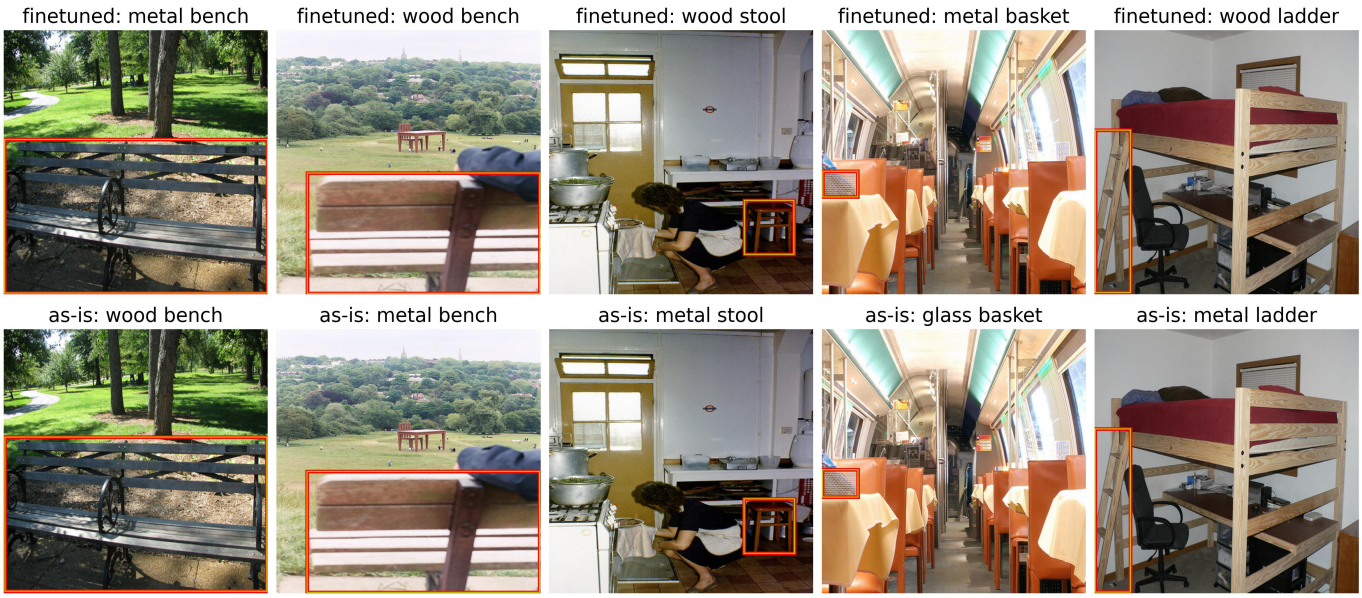

Fig. 4: **Object properties**: Correcting the wrong properties for several object classes.

Fig. 5: **Errors**: Remaining errors such as a Metal Bench was is confused with a Wooden Bench (third image) and a Metal Stool (right) which is a Wooden Stool but it has Metal legs.

Fig. 6: **Sorting**: A Plastic Bench is found directly i.e. rank 1 (left), whereas the Plastic Stool is found at rank 3 (right) after finding Stools with the wrong properties.

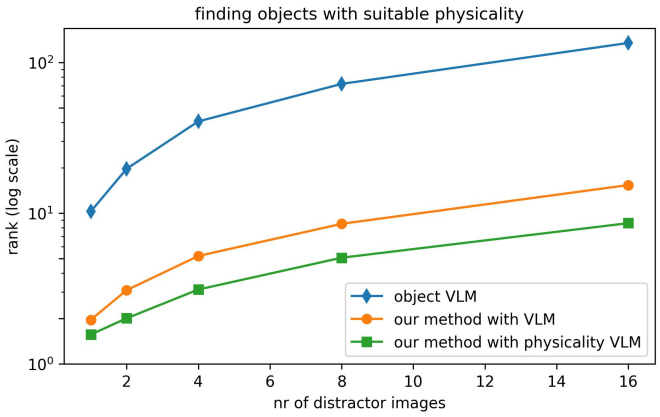

Fig. 7: **Effectiveness**: Our method with adapted VLM finds the right objects with suitable properties much faster than the object detection VLM and our method without adapted VLM.

H. Our Method on a Robot

Our final experiment is to equip a robot with our method. The robot is Spot by Boston Dynamics. We task it to search for objects to climb on, e.g. to increase its height and look over obstacles. It is remotely controlled past 16 positions that are a few meters apart. At each position, an image is recorded by its omni-directional cameras. Some examples are provided in Figure 8. Our method is applied to the collected images. In the 16 images, there are several distractor objects such as cables, desks, equipment, tape, etc.

Our method is searching for objects that can fulfil 'climb on' (e.g. to look over some obstacle). Equations 4 is initialized with the SPOT specifications, Equation 5 is invoked with 'climb on' and 'safe', with Grounding DINO for Equation 6. For this task, our method has ranked the most suitable objects and their properties. At rank 1, it finds a Wood Crate (top left), although actually it is a composite material. At rank 2, it suggests a Wood Bench (top right); and at rank 3 (bottom left) it suggests a Plastic Crate. The results are not perfect, but the suggested objects are sensible and can serve the purpose. It shows the potential of our method in practice.

V. Discussion and Conclusion

We proposed a dialogue of out-of-the-box foundational models (LLM and VLM) to find objects in the open world that afford desired actions that contribute to a specific robot's goal. We ground the responses of the LLM and VLM by taking the robot embodiment and physical object properties into account.

The results show that the framework can successfully localize the relevant objects, while taking into account robot embodiment and goal context. By forcing the LLM and VLM to reason about and detect relevant object properties, the method finds objects that are more useful to the task than a naive approach without the dialogue. Detecting objects with relevant properties is further improved by specifically finetuning the VLM on e.g. materials. There is still a performance gap as the VLM still struggles with distinguishing between subtle object properties, especially for small objects or objects that are often (partially) obscured. The current finetuned VLM already improves the search for the right objects with suitable properties.

As future work, we consider a number of ways to improve the proposed dialogue. The VLM currently finds objects that have a certain property, e.g. a stool that is made from wood. However, this does not consider that objects can be composed of different parts that have different properties. The work could therefore be extended to allow the VLM to find objects that have a mixture of properties (e.g. a stool made from wood and metal), or to query the LLM if an object property is relevant to some *part* of the object. Moreover, the dialogue has as input an open-vocabulary action, in combination with the intended effect. The dialogue can be edited to have the action-object tuple as output instead. Then, by only giving a textual specification of the goal, the framework could output open-vocabulary actions on relevant objects. Lastly, the LLM now manipulates class labels to find objects that are useful in the queried setting. Inspired by [17], the method could become more nuanced if considering the attributes that make an object useful instead. Then, any object with a useful attribute can be found, hence reducing reliance on the LLM to generate the correct class labels.

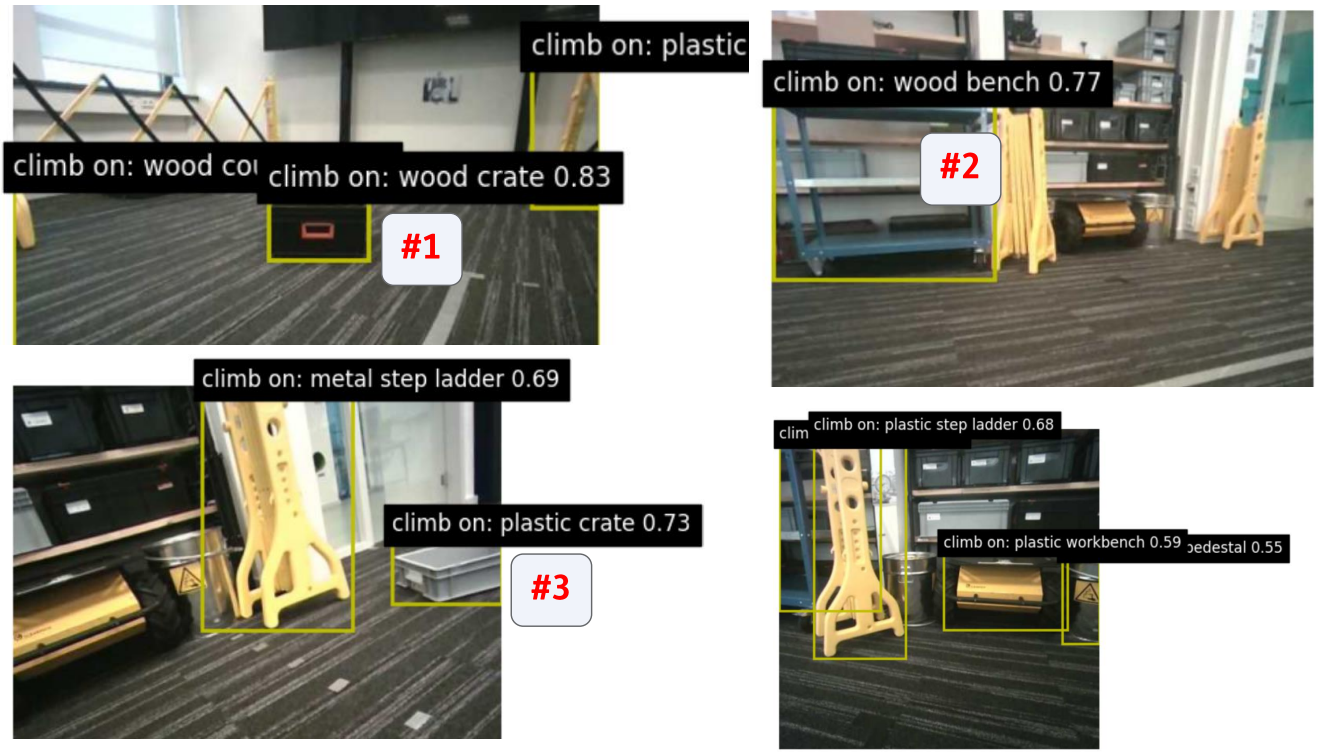

Fig. 8: **Robot**: With our method, the robot finds a Wooden Crate (top left) when the intended action is 'climb on'.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

[2] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics, 2023.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[4] Edward Y. Chang. Prompting large language models with the socratic method, 2023.

[5] Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman,

Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2024.

[6] Mingyu Ding, Zhenfang Chen, Tao Du, Ping Luo, Joshua B. Tenenbaum, and Chuang Gan. Dynamic visual reasoning by learning differentiable physics models from video and language, 2021.

[7] Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, Brian Ichter, Danny Driess, Jiajun Wu, Cewu Lu, and Mac Schwager. Foundation models in robotics: Applications, challenges, and the future, 2023.

[8] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. *ICRA*, 2024.

[9] James J. Gibson. The theory of affordances. In John Bransford Robert E Shaw, editor, *Perceiving, acting, and knowing: toward an ecological psychology*, pages pp.67–82. Hillsdale, N.J. : Lawrence Erlbaum Associates, 1977. URL https://hal.science/hal-00692033.

[10] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[11] Junnan Li, Dongxu Li, Silvio Savarese, Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. 2023.

[12] 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.

[13] Hongchen Luo, Wei Zhai, Jing Zhang, Yang Cao, and Dacheng Tao. Grounded affordance from exocentric view, 2023.

[14] OpenAI. ChatGPT. 2023. Accessed: 2024-05-21.

[15] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023.

[16] Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, Yue Wu, Wenhai Wang, Junsong Chen, Zhangyue Yin, Xiaozhe Ren, Jie Fu, Junxian He, Wu Yuan, Qi Liu, Xihui Liu, Yu Li, Hao

Dong, Yu Cheng, Ming Zhang, Pheng Ann Heng, Jifeng Dai, Ping Luo, Jingdong Wang, Ji-Rong Wen, Xipeng Qiu, Yike Guo, Hui Xiong, Qun Liu, and Zhenguo Li. A Survey of Reasoning with Foundation Models. pages 1–160, 2023. URL http://arxiv.org/abs/2312.11562.

[17] Jiajin Tang, Ge Zheng, Jingyi Yu, and Sibei Yang. Cotdet: Affordance knowledge prompting for task driven object detection. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3045–3055, 2023. URL https://api.semanticscholar.org/CorpusID:261531326.

[18] Xiaojuan Tang, Zilong Zheng, Jiaqi Li, Fanxu Meng, Song-Chun Zhu, Yitao Liang, and Muhan Zhang. Large language models are in-context semantic reasoners rather than symbolic reasoners, 2023.

[19] Jingkang Yang, Yuhao Dong, Shuai Liu, Bo Li, Ziyue Wang, Chencheng Jiang, Haoran Tan, Jiamu Kang, Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Octopus: Embodied vision-language programmer from environmental feedback, 2023.

[20] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language, 2022.

[21] Wei Zhai, Hongchen Luo, Jing Zhang, Yang Cao, and Dacheng Tao. One-shot object affordance detection in the wild, 2021.

[22] Zhipeng Zhang, Zhimin Wei, Guolei Sun, Peng Wang, and Luc Van Gool. Self-explainable affordance learning with embodied caption, 2024.