Open-Vocabulary Object Manipulation using Pre-Trained Vision-Language Models

Author Names Omitted for Anonymous Review. Paper-ID 110

***Abstract*—For robots to follow instructions from people, they must be able to connect the rich semantic information in human vocabulary, e.g. a "can you get me the pink stuffed whale?," to their sensory observations and actions. This brings up a notably difficult challenge for robots: while robot learning approaches allow robots to learn many different behaviors from first-hand experience, it is impractical for robots to have experiences that span all of this semantic information. We would like a robot's policy to be able to perceive and pick up the pink stuffed whale, even if it has never seen any data interacting with a stuffed whale before. Fortunately, static data on the internet has vast semantic information, and this information is captured in pre-trained vision-language models. In this paper, we study whether we can interface robot policies with these pre-trained models, with the aim of allowing robots to complete instructions involving object categories that the robot has never seen first-hand. We develop a simple approach, which we call Manipulation of Open-Vocabulary Objects (MOO), which leverages a pre-trained vision-language model to extract the relevant visual information from the language command, and conditions the robot policy on both the instruction and the corresponding visual information. In a variety of experiments on a real mobile manipulator, we find that MOO allows the robot to generalize zero-shot to a wide range of novel object categories and environments.**

I. Introduction

For a robot to be able to follow instructions from people, it must cope with the vast variety of human vocabulary, much of which may refer to objects that the robot has never interacted with first-hand. For example, consider the scenario where a robot has never seen or interacted with a plush animal from its own camera, and it is asked, "can you get me the pink stuffed whale?" How can the robot complete the task? While the robot has never interacted with this object category before, the internet and other data sources cover a much wider set of objects and object attributes than the robot has encountered in its own first-hand experience. In this paper, we study whether robots can tap into the rich semantic knowledge captured in such static datasets, and combine it with the robot's own experience, to be able to complete manipulation tasks involving novel object categories.

Computer vision and natural language models can capture the rich semantic information contained in static datasets. Indeed, composing modules for perception, planning, and control in robotics pipelines is a long-standing approach to robotics [30, 44, 16], allowing robots to perform tasks with a wide set of objects [5]. However, these pipelines are notably brittle, since the success of latter motor control modules relies on precise object localization. Furthermore, these approaches remain restricted to the set of object categories represented in the object detector's training set. On the other hand, several prior works have trained neural network policies with pre-trained image representations [18, 50, 31, 29] and pre-trained language instruction embeddings [28, 12, 1, 40]. While this form of vanilla pre-training can improve efficiency and generalization, it does not provide a mechanism for robots to ground and manipulate novel semantic concepts, e.g. unseen object categories referenced in the language instruction. This is because the language representation of this new object category will be out-of-distribution to the policy and remains disconnected with the robot's perceptual representation. This leads to a crossroads — some approaches can conceivably generalize to many object categories but rely on fragile pipelines; others are less brittle but cannot generalize to new semantic object categories.

To allow robots to generalize to semantic concepts, we will specifically choose to leverage pre-trained vision-language models (VLMs), rather than models pre-trained on one modality alone. Such models capture the rich information contained in diverse static datasets, while grounding the semantic linguistic concepts into a perceptual representation that can be directly connected to the robot's observations. Our goal is to combine this rich semantic information with diverse physical experiences on the robot to obtain a policy that generalizes to a broad set of language instructions, which involve semantically and physically novel objects. Our system receives a language instruction from a human and uses a VLM to identify the (x, y) image coordinates of all objects in the instruction. Since VLM models tend to be very large and computationally expensive to run in the control loop of robotics systems, we run the VLM only once at the start of an episode. The (x, y) coordinates of the objects on the first frame of the episode are fed into our manipulation policy allowing it to ground the natural language to objects and know which objects to act upon without seeing any demonstrations with those objects. The VLM is frozen throughout all of our training, and the policy is trained with the real VLM detector in the loop to prevent the brittleness that can plague prior pipelined approaches.

The main contribution of this paper is a flexible approach for open-vocabulary object manipulation that interfaces policy learning with pre-trained vision-language models. The pre-trained models we consider are trained on massive static image and language data that far exceeds the robot's own experience and provides the semantic knowledge needed to localize objects referenced in language instructions. The robot's policy is trained on top of these localizations to perform primitive skills using demonstration data covering a more modest yet still diverse set of 106 training objects. The composition of

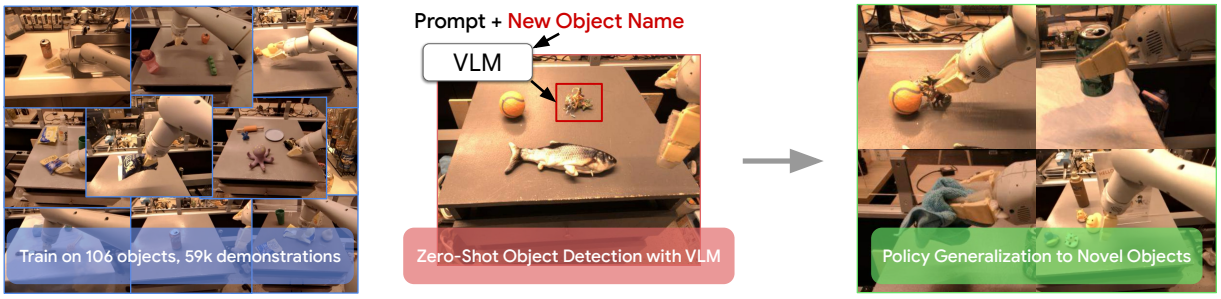

Fig. 1: Overview of our method: manipulation of open-vocabulary objects (MOO). We train a language-conditioned policy conditioned on object localizations from a frozen vision-language model (VLM). The policy is trained on demonstrations spanning a diverse set of 106 objects (left), allowing it to generalize to novel physical objects while the open-vocabulary VLM localization (center) allows the policy to generalize to novel semantic categories (right).

the pre-trained vision-language model and the control policy leads to an overarching language-conditioned policy that can complete commands that refer to novel semantic categories. When evaluated with unseen object categories on a real manipulator, our experiments indicate that our approach is significantly more successful than recent competing state of the art learned manipulation methods. Throughout this paper, we refer to our approach as Manipulation of Open-vocabulary Objects (MOO).

II. Related Work

a) Leveraging Pre-Trained Models in Robotic Learning: Using off-the-shelf vision, speech, or language models is a long-standing approach in robotics [43, 47, 12]. Modern pre-trained vision and language models have improved substantially over older models, and have played an increasing role in robotics research. A large body of prior work has trained policies on top of frozen or fine-tuned visual representations [18, 17, 50, 52, 48, 4, 38, 31, 29, 35, 23], while other works have leveraged pre-trained language models [11, 22, 28, 12, 1, 2, 13, 40]. Unlike these prior works, we aim to leverage vision-language models that ground language in visual observations. Our use of vision-language models enables generalization to novel semantic object categories, which cannot be achieved by using vision models or language models individually.

b) Generalization in Robotic Learning: A number of recent works have studied how robots can complete novel language instructions [41, 11, 22, 28, 12, 1, 25, 21, 2], typically focusing on instructions with novel combinations of words, i.e. compositional generalization, or instructions with novel ways to describe previously-seen objects and behaviors. Our work focuses on how robots can complete instructions with entirely new words that refer to objects that were not seen in the robot's demonstration dataset. Other research has studied how robot behaviors like grasping and pick-and-place can be applied to unseen objects [33, 24, 19, 7, 14, 6, 49, 3, 46], focusing on generalization to visual or physical attributes. Our experimental settings require visual and physical object generalization but also require semantic object generalization. That is, unlike these prior works, the robot must be able to ground a description of a previously-unseen object category. Works such as CLIPort [39] have shown generalization to unseen semantic categories and attributes. We aim to achieve such generalization without being restricted to table-top pick-and-place and without requiring a calibrated camera.

c) Open-World Object Detection in Computer Vision: Historically, object-detection methods have been restricted to a fixed category map covering a limited set of objects [8, 36, 10, 20]. These methods work well for the object categories on which they are trained, but have no knowledge of objects outside their limited vocabulary. Recently, a new wave of object detectors have emerged that aim to go beyond simple closed-vocabulary tasks by replacing the fixed one-hot category prediction with a shared image-language embedding space that can be used to answer open-vocabulary object queries [27, 9, 15, 51]. Typically these methods rely on internet-scale data in the form of pairs of image and associated descriptive text to learn the grounding of natural language to objects. Various methods have been employed to then extract object localization information in the form of bounding boxes and segmentation masks. In our work, we use the OWL-ViT detector due to it's strong performance in the wild and publicly available implementation [27].

III. Preliminaries

a) Problem Set-Up: Our goal is to allow robots to complete instructions for tasks that involve novel object categories that are not previously observed by the robot's camera. However, the object categories are represented by the vast amount of data on the internet, and hence, should be captured by vision-language models trained on static data.

Formally, we assume that the robot, with image observations $o \in \mathcal{O}$ and actions $a \in \mathcal{A}$, is provided with a set of expert demonstrations $\mathcal{D}_{\text{robot}}$ collected via teleoperation. Each demonstration corresponds to a sequence of observation-action pairs $\{(o_j, a_j)\}_{j=1}^{T}$ collected over a time horizon T, and is annotated with a structured language instruction ℓ for the task being performed in the demonstration. To help facilitate object generalization, we assume that these language instructions

are structured as a combination of a template and a list of object descriptions within that template. For example, for the instruction ℓ ="*move yellow banana near cup,*", the template is *"move X near Y,"* and the object descriptions are X ="*yellow banana*" and Y ="*cup.*"

All of the objects seen in the demonstrations are drawn from a set $\mathcal{S}_{\text{robot}}$, and our key objective is to complete new structured language instructions with a seen template but novel objects that are not in $\mathcal{S}_{\text{robot}}$, which also have novel object descriptions. In aiming to complete this goal, our approach will leverage imitation learning and vision-language models, which we briefly review in the rest of this section.

b) Imitation Learning and RT-1: MOO will build upon a language-conditioned imitation learning setup. The goal of language-conditioned imitation learning is to learn a policy $\pi(a \mid \ell, o)$, where a is a robot action that should be applied given the current observation o and task instruction ℓ. To learn a language-conditioned policy π, we build on top of RT-1 [2], a recent robotics transformer-based model that achieves high levels of performance across a wide variety of manipulation tasks. RT-1 uses behavioral cloning [34], which optimizes π by minimizing the negative log-likelihood of an action a given the image observations seen so far in the trajectory and the language instruction, using a demonstration dataset containing N demonstrations:

$$J(\pi) := \sum_{n=1}^{N} \sum_{t=1}^{T^{(n)}} \log \pi(a_t^{(n)} \mid \ell^{(n)}, \{o_j^{(n)}\}_{j=1}^{t}). \quad (1)$$

c) Vision-Language Models: In recent years, there has been a growing interest in developing models that can detect objects in images based on natural language queries. These models, known as vision-language models (VLMs), are enabling detectors to identify a wide range of objects based on natural language queries. Typically the text queries are tokenized and embedded in a high-dimensional space by a pretrained language encoder, and the image is processed by a separate network to extract image features into the same embedding space as the text features. The language and image representations are then combined to make predictions of the bounding boxes and segmentation masks. Given a natural language query, q, and an image observation on which to run detection, o, these models aim to produce a set of embeddings for the image $f_i(o)$ with shape $(\text{height}, \text{width}, \text{feature dim})$ and an embedding of the language query $f_l(q)$ with shape feature dim such that logits $= f_i(o) \cdot f_l(q)$ gives a logit score map and is maximized at regions in o which correspond to the queries in q. Each image embedding location within $f_i(o)$ is also associated with a predicted bounding box or mask indicating the spatial extent of that object corresponding to $f_i(o)$. In this work, we use the Owl-ViT detector [26], which we discuss further in Sec. IV-C.

IV. Manipulation of Open-Vocab Objects (MOO)

In this section, we now describe our method, MOO, that allows the robot to complete manipulation tasks involving novel object categories that have not been previously seen by the robot. The key goal of MOO is to develop a policy that can leverage the visually-grounded semantic information captured by pre-trained vision-language models for generalization to object types not in the policy training set. More specifically, we aim to use the VLM to localize objects described in a given instruction, while allowing the policy to complete the task using both the instruction and the object localization information from the VLM. MOO accomplishes this in two stages. First, the current observation and the words in the instruction corresponding to object(s) are passed to the VLM to localize the objects. Then, the object localization information and the instruction sans object information are passed to the policy, along with the observation, to predict actions.

The key design choice of MOO lies in how to represent object information encoded in VLMs and how to feed that information to the instruction-conditioned policy. In the remainder of this section, we first describe the design of these crucial aspects of the method. Given these choices, we then provide an overview of the model architecture and the training procedure as well as describe practical implementation details that allows us to deploy MOO on real robots.

A. Representing Object Information

To fully utilize the object knowledge encoded in the VLMs, we need to pick a representation that can be easily transferred to a text-conditioned control policy. We start with the task instruction ℓ, which may refer to previously unseen objects. We first split the task instruction ℓ into a verb v, which describes the skill that the we want the robot to perform, and a single noun or phrase X that describe the object of interest. For instructions involving multiple objects, we extract a sequence of such object descriptions as $X, Y, ...$ Note, that this is not a particularly restrictive assumption and it can be done in multiple ways, including querying a large language model to split the instruction into the two separate parts. Inspired by RT-1 [2], in this work, we focus on five different types of skills: "*pick* X," "*move* X *near* Y," "*knock* X *over*," "*place* X *upright*," and "*place* X *into* Y," where X and Y are object descriptions, such as "red cup" or "pink stuffed elephant," describing objects in the scene. We exclude all of RT-1 tasks involving drawers due to difficulties detecting drawers with the current state-of-the-art VLMs. We extract the object descriptions from the task commands using a simple regular expression.

Equipped with an object description X, we query a VLM to produce a bounding box of the object of interest with the prompt q = "An image of an X", and use the resulting detection (if any) as conditioning of our policy. To reduce the reliance of the exact segmentation of the object dimensions, we select a single pixel that is at the center of the predicted bounding box as the object representation. In the case of one object description, we use a one-hot single-channel object mask with the value set to 1.0 at the pixel of the object's predicted location. In the case of two object descriptions, we set the pixel value of the first to be 1.0 and the second to be 0.5. This design has two main advantages: first, it is a

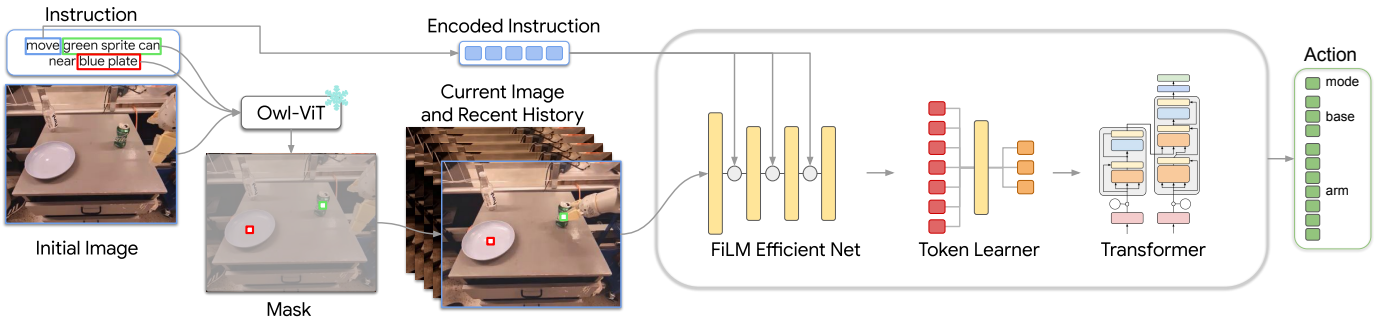

Fig. 2: MOO architecture overview: We build on the architecture of RT-1. We extract object location information (represented as a small mask on the center of the bounding box for each detection) on the first frame of an episode. When multiple objects are involved in an episode (such as "move can near plate," we use a different indicator value for each object mask. The segmentation mask is concatenated channel-wise to the input image for each subsequent frame of the episode. This requires modifying the EfficientNet to take in 4-channel input. Additionally, we remove the language embedding for everything except the task so, for example, "pick apple" only receives a language embedding for "pick" on the FiLM EfficientNet layers. The object specific information is only provided through the object instance mask.

general visual representation that can work with objects of any size as long as they are visible in the image, and second, it is compatible with a large selection of vision methods such as bounding boxes or segmentation masks as these can be easily transformed into a single, object-centric pixel location. We carefully ablate other object representation choices in the experiments.

Given the visual representation of the object and the remaining description of the skill that the robot is tasked to perform, we combine this task-relevant information with the current robot image to obtain the full observation that is fed to the MOO policy, which we describe in the next section. Importantly, this approach can handle object descriptions that are not previously seen in the robot's demonstration data, as long as it is sufficiently represented in the static large-scale training data of the VLM. For any of such unseen objects, we simply need to include a description of the object in the task command, e.g., "pick *green soda can*." Once the object description is translated into a pixel location by the VLM, the robot's policy trained on demonstration data only needs to be capable of correctly interpreting the mask location and how to physically manipulate the novel object's shape, rather than needing to also ground the semantic object description.

B. Architecture and Training of MOO

We present the model architecture and information flow of MOO in Fig. 2. We start by extracting the object descriptions from the language instruction, which are then, together with the initial image, fed into the VLM to output object locations in the image. As described previously, the object information is represented as a one-hot object mask where the only pixels with non-zero values are in the center of objects of interest.

Once we obtain the mask, we append it channel-wise to the current image together with the recent image history, which is passed into the RT-1 policy architecture [2]. We use a language model to encode the language instruction that excludes the object descriptions extracted earlier in an embedding space of an LLM. The images are processed by an EfficientNet [42] conditioned on the text embedding via FiLM [32]. This is followed by a Token Learner [37] to compute a small set of tokens, and finally a Transformer [45] to attend over these tokens and produce discretized action tokens. We refer the reader to the RT-1 paper for details regarding the later part of the architecture. The action space corresponds to the 7-DoF delta end-effector pose of the arm (including x, y, z, roll, pitch, yaw and gripper opening).

The entire policy network is trained end-to-end using the imitation learning objective introduced in Equation 1 in Sec. III. Importantly, the VLM used to detect the objects of interest is frozen during training, so that it does not specialize or overfit to the objects in the robot demonstration data. The policy is trained with this frozen VLM in the loop, so that the policy can learn to be robust to errors made by the VLM given other information in the image.

C. Practical Implementation

To detect objects in our robot images, we use the Owl-ViT open-vocabularly object detector [27]. In practice, we find that it is capable of detecting most clearly visible real world objects without any fine-tuning, given a descriptive natural language phrase. This is likely due to the billions of internet images and associated text on which the model was trained. The interface to the detector requires a natural language phrase describing what to detect (e.g., "An image of a small blue elephant toy.") along with an image to run the detection on. The output from the model is a score map indicating which locations are most likely to correspond to the natural language description and their associated bounding boxes. We select a universal score threshold to filter detections. To detect our objects, we rely on some prompt-engineering using descriptive phrases including the color, size, and shape of objects, though most of our prompts worked well on the first attempt. We share the prompts in the appendix.

To make the inference time practical to run on real robots, we extract the object information via VLM only in the first frame of the episode. By doing so, the majority of heavy computation is executed only once at the very beginning and we can perform real-time control for the entire episode. Since

the information is appended to the current observation, we rely on the policy to find the corresponding object in the current image if the object has moved since the first timestep.

D. Training Data

We start with the demonstration data used by RT-1 [2] covering 16 unique objects. However, despite the use of the VLM for semantic generalization, we expect that the policy will require more physical object diversity in order to generalize to novel objects. Therefore, we expand the dataset with additional diverse "pick" data across a set of 90 diverse objects, for a total of 106 object, as shown in Figure 4. We choose to only expand the set of objects for the picking skill, since it is the fastest skill to execute and therefore allows for the greatest amount of diverse data collection within a limited budget of demonstrator time.

Note that our additional set of 90 diverse objects only appear in episodes in which the robot is performing a "pick" task. All other tasks, such as "move near" or "place into", are learned from the original 16 objects in the RT-1 dataset. From our 90 diverse objects, we randomly selected 13 objects on which to perform a "seen" evaluation. We selected another 10 objects which were never seen during any training episodes on which to perform an "unseen" evaluation. We present the detailed object statistics in Fig. 3.

V. Experiments

In our experiments, we aim to answer the following questions:

1) To what extend does MOO generalize across objects for different manipulation skills including objects never seen at training time?
2) How can we leverage object-centric representations in MOO for novel use-cases?
3) How does the object generalization performance of MOO scale with (a) the number of training episodes, (b) the number of unique objects in the training episodes and (c) the size of the model?

A. Experimental Setup

Seen and unseen objects. We collect training data on a table-top environment across a broad set of 106 different object types. A human operator manually pilots the robot to complete training tasks. We evaluate performance on the aforementioned 13 "seen" objects in our training data and report the performance as "seen." We hold out another 10 objects not present in any of our training episodes and report performance on these as "unseen." Note that previous works often focus on unseen combinations of previously seen commands (e.g. "pick an apple" even though the training data only contains "move an apple into a bowl"), we define a more strict definition of unseen objects, where all of our unseen objects were not seen during training at any point for any task, therefore making our unseen performance a zero-shot generalization task. Furthermore, we report results across a range of different environments that introduce novel textures, backgrounds, locations, and additional open-world objects not present in the training data.

Object generalization evaluation details. We evaluate our method on a set of tabletop tasks involving manipulating a diverse set of objects. We separately report performance both on objects seen during training and objects outside the training distribution, which is the main focus of this work. We use a mobile manipulator with 7 degree-of-freedom arm and a two-fingered gripper (see 5). Our experiments evaluate the percent of successfully completed manipulation commands which include five skills: "pick", "move near", "knock" and "place upright" across a set of evaluation episodes. For "pick" episodes, success is defined as 1.) grasping the specified object and 2.) lifting the object 6 inches from the table top. For "move near" episodes, success is defined as 1.) grasping the specified object and 2.) placing it within 6 inches of the specified target object. For "knock" episodes, success is defined as placing the specified object from an "upright" position onto its side. "Place upright" tasks are the inverse of "knock" and involve placing an object from its side into an upright position. "place into" tasks involve placing one object into another, such as an apple into a bowl. In order to demonstrate object specificity and robustness, for all evaluation episodes, we include between two to four distractor objects in the scene which the robot should not interact with. For each evaluation episode, we randomly scatter the evaluation object(s) and the distractor objects onto the table. There is no consistent placement of the target object relative to the distractors. We repeat this process 21 times and report the performance in the form of success percentage. We present the experimental setup in Fig. 5.

Robustness evaluation details. In most of our evaluations, the only difference from training episodes are the newly introduced "unseen" objects; however, we also investigate generalization to different furniture and backgrounds as visualized in Figure 7. The first set of these difficult evaluation scenes introduces 6 evaluations across 5 additional open-world objects that correspond to random household items that have not been seen at any point during training. The second set of difficult scenes introduces 14 evaluations across 2 patterned tablecloths that are visually similar to the object arrangements in the scene; these tablecloth textures are significantly more challenging than the foregrounds seen in the training demonstration dataset, which contained solely unaltered default gray counter-tops. Finally, the last set of difficult scenes utilize 14 evaluations across 3 new environments in kitchen and office spaces that were never present training demonstrations. These new scenes simultaneously change the counter-top materials, the backgrounds, the lighting conditions, and distractor items.

Baselines. We compare MOO to two prior methods: RT-1 [2] and a modified version of VIMA [13], which we refer to as "VIMA-like". VIMA-like preserves the cross-attention mechanism, but uses the mask image as the prompt token and the current image as state token. These modifications are necessary for this comparison because the original VIMA implementation is tied to the action space of a UR5 robot and is not suitable to use with our data and robot, i.e., our robot

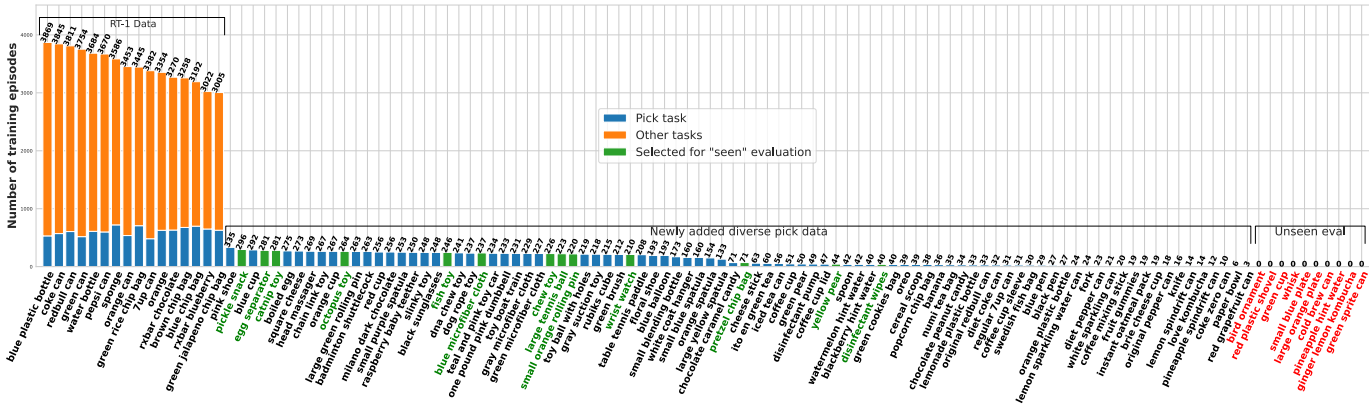

Fig. 3: Distribution of training objects for "pick" episodes and other skills. The data on the left was what was used by [2]. We augmented RT-1 data with a large number of diverse pick episodes in order to demonstrate strong generalization to unseen objects. Blue and green bars represent "pick" episodes and orange bars represent other tasks like "move near" or "knock." "Green" bars were the objects we randomly selected for "seen" evaluations. All randomly selected "unseen" objects are shown to the right.

arm moves in 6D and it has a gripper that can open and close continuously.

B. Generalization to Novel Objects

We start our experiments with investigating the question: *To what extend does MOO generalize across objects for different manipulation skills including objects never seen at training time?* To answer this question, we compare our method, MOO, to two other baselines: RT-1 [2] that uses an ImageNet-pretrained EfficientNet tokenizer but does not use an additional VLM to detect objects and VIMA-like [13] that also utilizes object-centric representations but not in the form of object pixels. These two baselines correspond to common alternatives where the computer vision data is used as a pre-training mechanism (as in the case of RT-1) or object-centric information is fed to the network in a different way (as in the case of VIMA-like).

The results of these experiments are in Table I and example images of MOO execution are in Fig 6. We present two versions of RT-1 – the original algorithm trained on the dataset described in [2] as well as RT-1 that is trained on the additional diverse pick data introduced in this work for a fair comparison. Comparing MOO to the baselines on the pick tasks, we can observe a substantial improvement over the seen object performance as well as the unseen objects, which in both cases reaches $\sim 50\%$ improvement. This can be explained by the fact that MOO is able to correctly utilize an underlying VLM to find novel objects that the robot has not interacted with and it can incorporate that information more effectively than the VIMA-like baseline. When comparing the performance on seen objects for the skills other than pick, we observe a slightly worse performance than for the pick tasks. This is understandable given the fact that the "Seen objects" for "Other skills" have only been seen during the "pick" episodes as shown in Fig. 3. This demonstrates the ability of MOO to transfer the learned object generalization across the skills so that the objects that have only been picked can now be also used for other purposes. In addition, we observe significant generalization of MOO to unseen objects (i.e. unseen in any previous tasks, including pick) that is at the same level as for unseen objects for the pick skill 50% better than the next best baseline.

	Pick		Other skills	
Method	Seen objects	Unseen objects	Seen objects	Unseen objects
RT-1 (our data) [2]	54	25	50	50
RT-1 (original data)	31[1]	38	17[1]	13
VIMA-like [13]	62	50	50	25
MOO (ours)	**92**	**75**	**83**	**75**

TABLE I: Overall success rate of MOO and competitive prior methods for seen and unseen objects across multiple skills including pick, move-near, place-upright, knock-over and place.

C. Robustness to Novel Use-Cases

To further test the robustness of MOO, we analyze novel evaluation settings with significantly increased difficulty and visual variation, which are shown in Figure 7. To reduce the number of real robot evaluations, we focus this comparison on the picking skill.

Across all of these challenging evaluation scenes, we find that MOO is significantly more robust compared to a VIMA-like baseline [13] and an RT-1 baseline [2]. This is particularly visible in the "Challenging Textures" eval, where our method is more than $7\times$ better than the baselines. This indicates that the use of VLMs in MOO not only improves generalization to new objects that the robot has not interacted with, but also significantly improves generalization to new backgrounds and environments.

D. MOO Ablations

To answer our last experimental question, we conduct a number of ablations to assess the impact of the size and

[1]RT-1 (original data) has not actually seen any any objects in the "seen" or "unseen" categories, since it was trained only on RT-1 data.

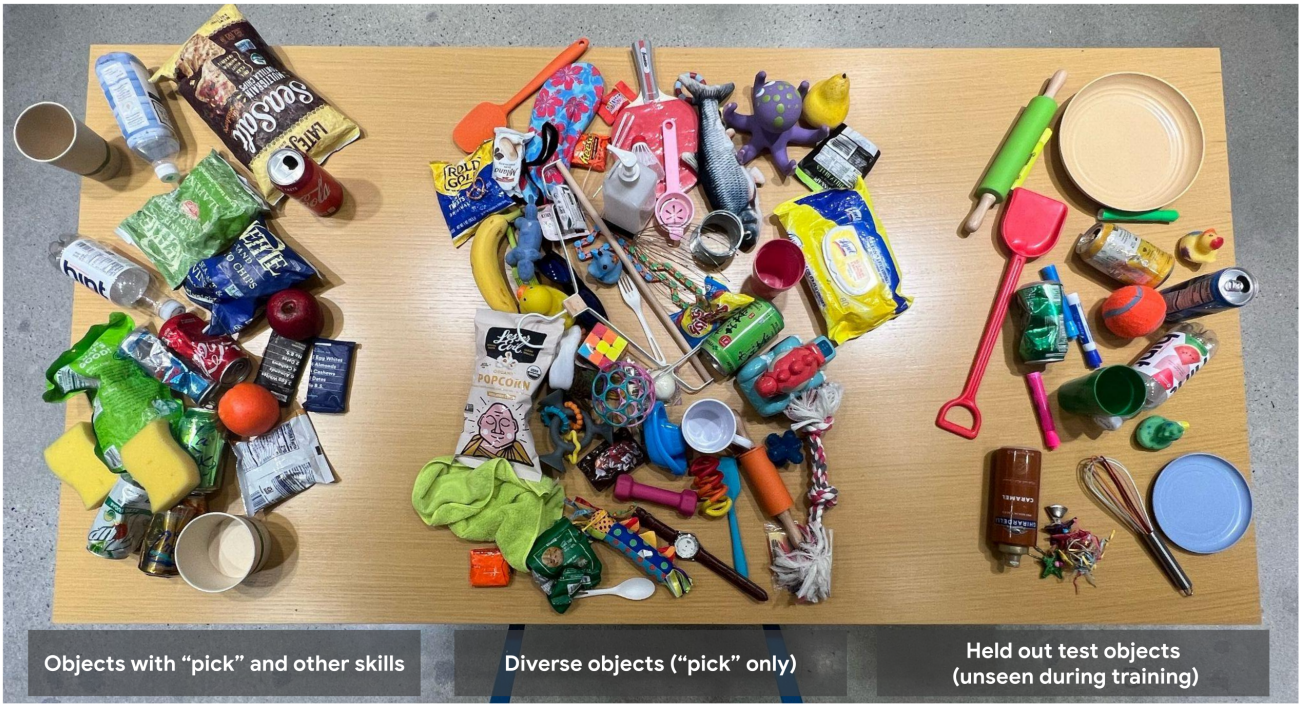

Fig. 4: We display objects used in this paper separated into three groups. The first group on the left is the RT-1 objects. These are most represented in the dataset (approximately 70% of all training data) and contains all skills (“pick”, “move near”, “knock”, “place upright”, “place into”). The diverse objects in the middle appear only in “pick” episodes and appear less frequently in the training data. All ”seen” evaluations are conducted on objects selected from the middle pile. The objects on the far right were held out from all training and are used only for evaluation purposes (performance on these is reported as “unseen” throughout the text).

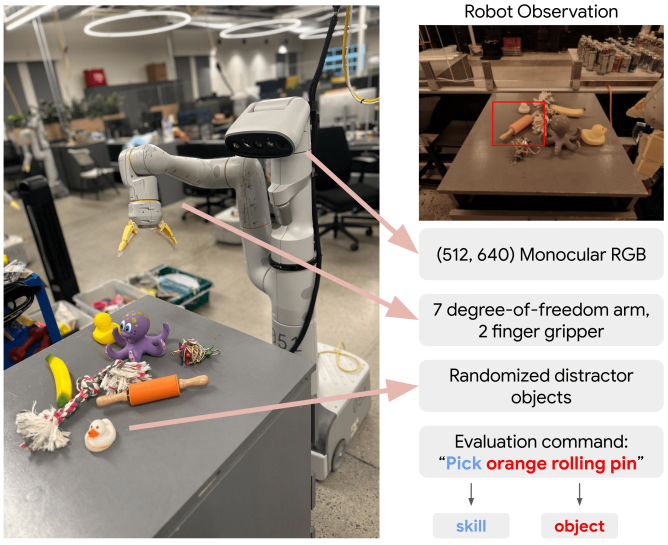

Fig. 5: Image of our robot hardware and evaluation setting.

Method	Open-World Objects	Challenging Textures	New Environments
RT-1 (our data) [2]	17	7	29
VIMA-like [13]	50	7	7
MOO (ours)	**67**	**50**	**43**

TABLE II: Robustness evaluations for novel use cases. MOO is able to handle new objects, textures, and environments with substantially greater success than prior methods.

Dataset Filtering		Pick	
Objects	Episodes per Object	Seen objects	Unseen objects
100%	100%	**92**	**75**
50%	100%	62	38
100%	50%	46	38
100%	10%	23.	0

TABLE III: Performance of our policy relative to the amount of data used for training.

diversity of our dataset and the scale (in terms of number of parameters) of our model. In Table III we vary both the number of unique objects in the training set (reducing it from 106 to 53 unique training objects) as well as the number of total training episodes (reducing it by half – from 59051 training episodes to 29525) while keeping all objects in the dataset. Note that cutting the number of objects from 106 to 53 also reduces the total number of training episodes by about half. We aim to vary these two axes to determine the impact of the overall size of the dataset vs its object diversity on the final results. Interestingly, we find similar performance in both cases, especially for the unseen objects. This indicates that it might be possible for our method to achieve a similar level of object generalization by scaling up the total amount of training episodes without significantly increasing the set of unique objects.

Fig. 6: Example images of our policy detecting and grasping objects not seen during training time. The object detections are colored in correspondence to the text above the image, and the images are ordered left to right across time.

In Fig. 8, we investigate the impact of scaling the size of our model. We train to completion two smaller versions of MOO where we scale down the total number of layers and the layer width by a constant factor. The version of MOO that we use in our main experiments has 111M parameters, which, for the purpose of this ablation, we then reduce by an order of magnitude down to 10.2M and then by 5X again down to 2.37M. Comparing different sizes of the model, we find significant drop offs in both "seen" and "unseen" object performance. We also note that we could not make MOO larger than 111M parameters without increasing the latency on robot to an unacceptable level, but we expect continued performance gains with bigger models if the speed and compute requirements can be relaxed.

VI. Conclusion

In this paper, we presented an approach for leveraging the rich semantic knowledge captured by vision-language models in robotic manipulation policies. Our evaluation showed that our approach substantially improves the generalization of robot policies, allowing robots to complete novel instructions involving previously-unseen object categories and enabling

Fig. 7: To test the robustness of MOO even further, we evaluate on (a) additional novel objects, (b) challenging texture backgrounds with two separate tablecloths that are quite similar visually to objects in the scene, and (c) novel environments that were never seen during training.

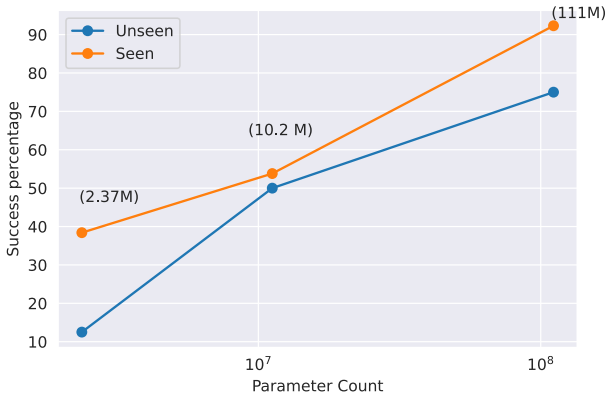

Fig. 8: Pick success vs. model size. We see continuous improvements on both seen and unseen objects as we increase the number of parameters of our model architecture while keeping the data set size fixed. In comparison to our main model, we scaled down layer widths and depth by the same constant multiplier. We expect more performance gains at larger model capacity, yet are currently unable to scale further due to real time inference constraints on our robot.

greater performance with visually-complex table textures and in novel environments.

Despite the promising results, MOO has multiple important limitations. First, the object mask representation used by MOO can uniquely identify an object in a scene in many but not all cases. For example, if an apple is placed on a plate, a mask centered on the apple may be referring to the plate or the apple. Exploring more expressive object representations is an exciting direction for future work. Second, we expect the physical object generalization of the policy to still be limited by the diversity of robot data. For example, we expect that the robot may struggle to grasp novel objects with drastically different shapes or sizes than those seen in the training demonstration data, even if the vision-language model can accurately localize the object. Finally, MOO cannot currently handle complex object descriptions involving spatial relations, such as "the small object to the left of the plate." Fortunately, we expect performance on tasks such as these to improve significantly as vision-language models continue to advance moving forward.

References

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

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

[3] Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*, 2021.

[4] Bryan Chen, Alexander Sax, Gene Lewis, Iro Armeni, Silvio Savarese, Amir Zamir, Jitendra Malik, and Lerrel Pinto. Robust policies via mid-level visual representations: An experimental study in manipulation and navigation. *arXiv preprint arXiv:2011.06698*, 2020.

[5] Aidan Curtis, Xiaolin Fang, Leslie Pack Kaelbling, Tomás Lozano-Pérez, and Caelan Reed Garrett. Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1940–1946. IEEE, 2022.

[6] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, 2019.

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

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. doi: 10.1109/CVPR.2014.81.

[9] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=lL3lnMbR4WU.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

[11] Felix Hill, Sona Mokra, Nathaniel Wong, and Tim

Harley. Human instruction-following with deep reinforcement learning via transfer-learning from text. *arXiv preprint arXiv:2005.09382*, 2020.

[12] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2021.

[13] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2022.

[14] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018.

[15] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr - modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1780–1790, October 2021.

[16] Peter Karkus, Xiao Ma, David Hsu, Leslie Pack Kaelbling, Wee Sun Lee, and Tomás Lozano-Pérez. Differentiable algorithm networks for composable robot learning. *Robotics: Science and Systems (RSS)*, 2019.

[17] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 769–776. IEEE, 2017.

[18] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[19] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5), 2018.

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

[21] Hao Liu, Lisa Lee, Kimin Lee, and Pieter Abbeel. Instruction-following agents with jointly pre-trained vision-language models. *arXiv preprint arXiv:2210.13431*, 2022.

[22] Corey Lynch and Pierre Sermanet. Grounding language in play. 2020.

[23] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[24] 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.

[25] Oier Mees, Lukas Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics and Automation Letters*, 7(4):11205–11212, 2022.

[26] 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 with vision transformers. *arXiv preprint arXiv:2205.06230*, 2022.

[27] Matthias Minderer, Alexey A. Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. *ArXiv*, abs/2205.06230, 2022.

[28] Suraj Nair, Eric Mitchell, Kevin Chen, Brian Ichter, Silvio Savarese, and Chelsea Finn. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pages 1303–1315. PMLR, 2021.

[29] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[30] NJ NILLSON. Shakey the robot. *SRI International, Technical Note*, 323, 1984.

[31] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. *arXiv preprint arXiv:2203.03580*, 2022.

[32] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[33] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *IEEE international conference on robotics and automation (ICRA)*, 2016.

[34] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[35] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. *arXiv preprint arXiv:2210.03109*, 2022.

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali

Farhadi. You only look once: Unified, real-time object detection, 2015. URL https://arxiv.org/abs/1506.02640.

[37] Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021.

[38] Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021.

[39] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, 2022.

[40] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *arXiv preprint arXiv:2209.05451*, 2022.

[41] Simon Stepputtis, J. Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and H. B. Amor. Language-conditioned imitation learning for robot manipulation tasks. *ArXiv*, abs/2010.12083, 2020.

[42] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[43] Seth Teller, Matthew R Walter, Matthew Antone, Andrew Correa, Randall Davis, Luke Fletcher, Emilio Frazzoli, Jim Glass, Jonathan P How, Albert S Huang, et al. A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments. In *2010 IEEE International Conference on Robotics and Automation*, pages 526–533. IEEE, 2010.

[44] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[46] Bo-Han Wu, Suraj Nair, Roberto Martín-Martín, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. *ArXiv*, abs/2103.04174, 2021.

[47] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[48] Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7293. IEEE, 2020.

[49] Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. In *CoRL*, 2020.

[50] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018.

[51] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16793–16803, June 2022.

[52] Brady Zhou, Philipp Krähenbühl, and Vladlen Koltun. Does computer vision matter for action? *Science Robotics*, 4(30):eaaw6661, 2019.