Zero-shot Text Behavior Retrieval

Hung-Chieh Fang*
Department of Computer Science
b09902106@csie.ntu.edu.tw

Kuo-Han Hung*
Department of Computer Science
b09902120@csie.ntu.edu.tw

Nai-Xuan Ye*
Department of Computer Science
b09902008@csie.ntu.edu.tw

Shao-Syuan Huang*
Department of Computer Science
b09902131@csie.ntu.edu.tw

Abstract

Imitation learning has enabled robots to learn novel skills, yet its scalability is limited by high supervision demands. To address this, we propose Zero-Shot Text Behavior Retrieval, which leverages task descriptions and Vision Language Models (VLMs) to retrieve task-relevant data from offline datasets in a zero-shot manner, bypassing the need for expert data. Our approach uses text-guided object detection and CLIP embedding-based retrieval to detect task success. Tested on three simulated and one real-world task, it outperforms expert-dependent methods, demonstrating strong generalizability and efficiently training policies without additional metadata or demonstrations.

1 Introduction

In the realm of robotics, one of the ultimate goals is to empower robots to do versatile tasks across different environments. In recent years, imitation learning has emerged as a promising approach, achieving remarkable performance on diverse manipulation tasks. However, imitation learning requires a great collection of expert demonstrations, which can be difficult to obtain in real-world scenarios.

To address the limitation, prior works aim to enhance the sample efficiency for new tasks by leveraging unlabeled offline data and a small amount of expert data. Such offline data include many task-agnostic and sub-optimal data (Mees et al. [2022], Zhu et al. [2020]). One line of work focuses on learning pre-trained visual representations for downstream control (Nair et al. [2022], Xiao et al. [2022]). Despite the effectiveness of these approaches in utilizing unlabeled offline data, they do not make use of the offline data during fine-tuning, potentially hindering the downstream performance. Another line of approach aims to retrieve task-relevant data from the offline dataset (Du et al. [2023a], Nasiriany et al. [2022]) and use it for learning downstream control. One limitation of prior works is the necessity of expert data, which constrains generalizability to the real-world setting. Realizing the limitation, we propose to leverage only task descriptions to retrieve task-relevant data from the offline dataset as illustrated in Figure 4.

We propose Zero-Shot Text Behavior Retrieval, a method that retrieves task-relevant data with only task descriptions and visual language models, the data is then used to train a policy with behavior cloning. Our method is motivated by the recent success of visual language models, which can strongly connect the relationship between visual and text inputs. Our pipeline begins by segmenting crucial information of an image using text-guided object detection models. The cropped images are then used to compute the similarity to pre-defined positive and negative prompts, which describe the states of a successful and a failed task. If the positive prompt has a higher similarity score with the

*equal contribution

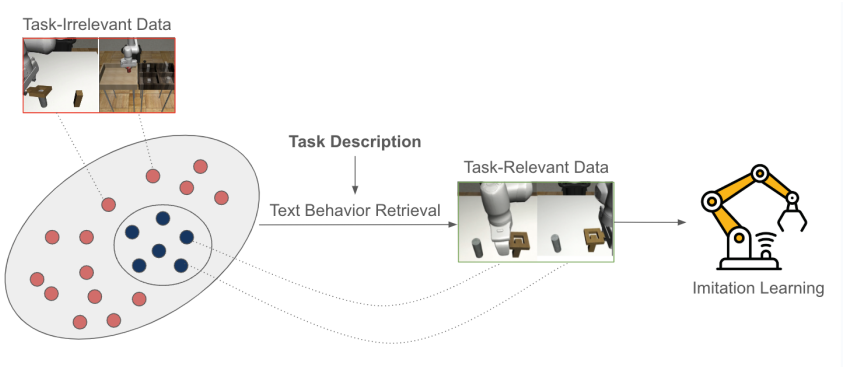

Figure 1: Illustration of our Text Behavior Retrieval framework

image, the instance is then added to the training data. Subsequently, the data is employed to train a policy through behavior cloning. We evaluate on 3 simulated and 1 real-world task and show that our method can outperform existing methods that rely on additional expert data. Our contribution can be listed as follows:

1. We are the first work on text-conditioned behavior retrieval to the best of our knowledge.
2. Given any unlabeled dataset, our framework can be used to train the policy without any other target environment's metadata or demonstration.

2 Related Work

Vision Language Models (VLMs). Recently, vision language models have been intensively investigated. These models are designed to intricately learn the correlations between vision and language, enabling them to make accurate predictions even in zero-shot scenarios. Video-LLama (Zhang et al. [2023]) and BLIP (Li et al. [2022]) are the recent open-source state-of-the-art on visual question answering, demonstrating impressive capabilities in detailed image description. However, in our experiments, we observed limitations in their ability to recognize complex robotics control. OWL-ViT (Minderer et al. [2022]) and OWLv2 (Minderer et al. [2023]) are open-vocabulary object detection networks trained on a variety of (image, text) pairs. In our work, we utilize them to extract crucial information from the image. CLIP (Radford et al. [2021]) is a multi-modal vision and language model. It can be used for image-text similarity and for zero-shot image classification. In our work, we use it to compute the similarity between image and success/fail states.

VLMs for task generalization. In robotics research, there's a growing trend to emulate the success of Large Language Models (LLMs) by developing versatile, generalist control models for robots. These models aim for zero-shot capabilities in diverse environments. To achieve this, there's a convergence of robotics control and Vision-Language Models (VLMs). For instance, Jang et al. [2022] created an adaptive imitation learning system that learns from demonstrations and interventions, responding to various forms of task-related information. Brohan et al. [2023] explored how VLMs trained on extensive internet-scale data can enhance robotic control through direct integration, leading to improved generalization and emergent semantic reasoning. Although these methods effectively utilize unlabeled offline data, their potential is not fully exploited during fine-tuning, which could impact downstream performance.

Large-Scale vision-language pretraining for task specification. The detection of success signals is a crucial aspect of robotics research. A significant body of work has focused on training reward models by utilizing knowledge derived from large-scale pretrained success detectors, as indicated in Mahmoudieh et al. [2022]. Moreover, there has been an increasing trend to harness the extensive pretrained knowledge from vision-language models for success detection or task specification. For example, Du et al. [2023b] fine-tuned a large Vision-Language Model, Flamingo Alayrac et al. [2022], using substantial data from manipulation tasks, aiming to extract success signals from pretrained Visual Question Answering (VQA) models. Additionally, Cui et al. [2022] experimented with using

similarity measures between embeddings of user-specified goals and robot observations for goal selection and policy learning.

Unlike previous studies that primarily employed vision-language models for reward modeling or as success detectors, our work diverges by focusing on the use of these models for filtering robot data. Remarkably, we aim to achieve this without the need for additional pretraining, setting our approach apart from the conventional methodologies in the field.

Behavior Retrieval. Recently, some prior works have focused on retrieval from offline datasets in imitation learning. Nasiriany et al. [2022] proposed to learn skills from the offline dataset and subsequently learn a policy for the target task that invokes these learned skills. Du et al. [2023a] aims to retrieve task-relevant data from the offline dataset using a small amount of expert data. The retrieved state-action tuples are then used to learn low-level control policy. Different from prior works, we do not use expert demonstrations for retrieval. Instead, we use task descriptions and VLMs to retrieve relevant demonstrations from the offline dataset.

3 Method

3.1 Problem Formulation

The problem setting in Behavior Retrieval involves learning a target task from a limited amount of task-specific expert data (D_t) and a larger amount of sub-optimal, unlabeled data (D_{prior}). Both datasets share a state space S and an action space A. Instead of the standard approach, where models are pre-trained on D_{prior} and fine-tuned on D_t, Behavior Retrieval first learns a similarity metric for (s, a) pairs from D_{prior}. Subsequently, given D_t, this metric is used to retrieve relevant (s, a) pairs from D_{prior}, and finally, π_t is trained on the combined dataset of D_t and the retrieved data using imitation learning. From the work of Du et al. [2023a], the behavior retrieval method outperform the standard fine-tuning method.

From a high-level perspective, the retrieval process in previous work (Du et al. [2023a]) employs an auto-encoder to compress state information, such as images, into low-dimensional embeddings. The similarity of embeddings from D_t and D_{prior} is then computed to determine whether an instance in D_{prior} is relevant to the target task. During the training of imitation learning, expert data and relevant data, denoted as D_{ret}, are combined as training data, as illustrated in Figure 3.

In the next section, we introduce a method that doesn't require environment information and expert data. Using a text prompt, we leverage text vision models to retrieve relevant data. This approach doesn't necessitate fine-tuning, and the zero-shot method can be generalized to various tasks.

3.2 Text Behavior Retrieval

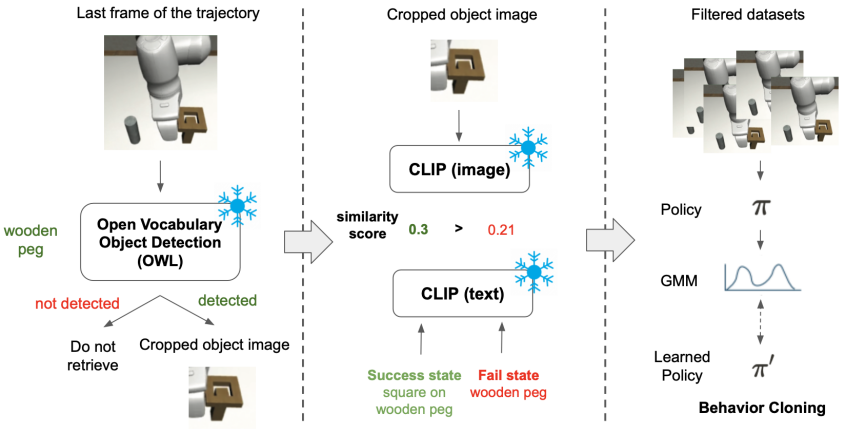

Figure 2: Zero-shot text behavior retrieval

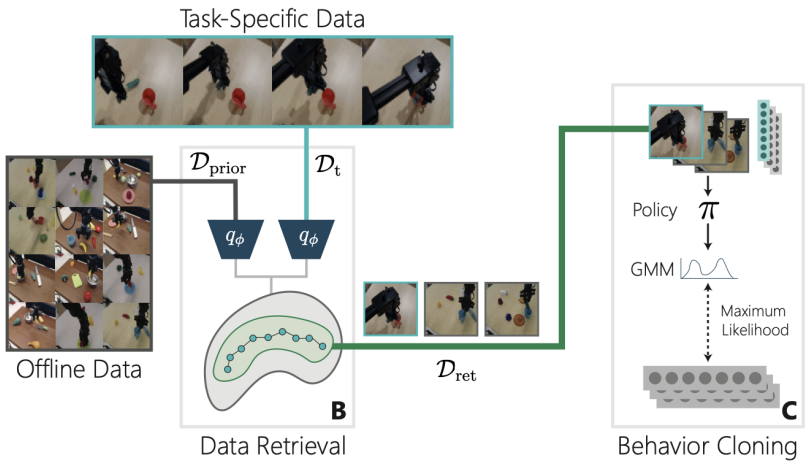

Figure 3: Behavior retrieval from Du et al. [2023a]

The pipeline of our approach is depicted in Figure 2. The method comprises three stages.

3.2.1 Object Segmentation

In the first stage, we employ text-guided object detection models, OWL-ViT (Minderer et al. [2022]) and OWLv2 (Minderer et al. [2023]), to segment the relevant part of the image crucial to the target task. For instance, in the NutAssembly task, the model captures the portion of the image determining the success of the demonstration (we prompt the model with "wooden peg). The image is then cropped, retaining only the essential part for further processing. If the model fails to detect any task-related object, the instance is considered failed and is not used for training the imitation learning model.

3.2.2 Retrieval Process

The second stage involves the main retrieval process. For each task, we compute the similarity of CLIP embeddings of the image with a pair of positive and negative text prompts. The positive prompt describes the successful scenario, and the negative prompt describes failure. For example, in the NutAssembly case, we have prompts "wooden peg" and "square on wooden peg." The CLIP embeddings of the cropped image and the two text prompts are compared. If the positive prompt has a higher similarity score with the image, the instance is considered a success, and vice versa. The CLIP similarity score helps retrieve relevant data for the subsequent step.

3.2.3 Imitation Learning

Following previous work, we train the agent to imitate D_{ret} with a behavior cloning loss. Specifically, we train

$$\min_{\psi} \mathbb{E}_{(s,a)\sim D_{ret}}[-\log \pi_\psi(a|s)] \tag{1}$$

We encode the observation modalities separately before combining them into a shared state embedding for the policy. Using an LSTM architecture for the policy allows for object permanence amidst occlusion and the observation of features like object velocity. The policy outputs parameters to a Gaussian mixture model (GMM), and the final actions are chosen by sampling from the GMM, enabling better modeling of multi-modal behavior often present in large multi-task datasets.

4 Experiments

4.1 Datasets

Following the method from Du et al. [2023a], we build three different environments, **CanPick**, **NutAssembly**, and **Office** and gather trajectories from them as D_{prior}.

CanPick. In this RoboSuite environment, the task is for a simulated robot to pick and place a coke can from one bin to another. We use the same trajectories as Du et al. [2023a], which contains a mix of 400 human-collected demos where half complete the task and half fail by randomly throwing the can out of the bin.

NutAssembly. The modified RoboSuite task is for a simulated robot to pick and insert a square into the right side peg. We collect a mix of 400 machine-generated demos where half complete the task and, in the other half, the robot puts the square onto the wrong (left) peg.

Office The third task is for a simulated robot in an office environment to pick an eraser and place it into a specified tray. We collect 1200 machine-generated demos where half complete the task and, in the other half, the robot fails to grasp the eraser and drop it into the tray. As for text inputs to object detection and CLIP models, we have tested many kinds of prompt and use Table descriptions as our final inputs.

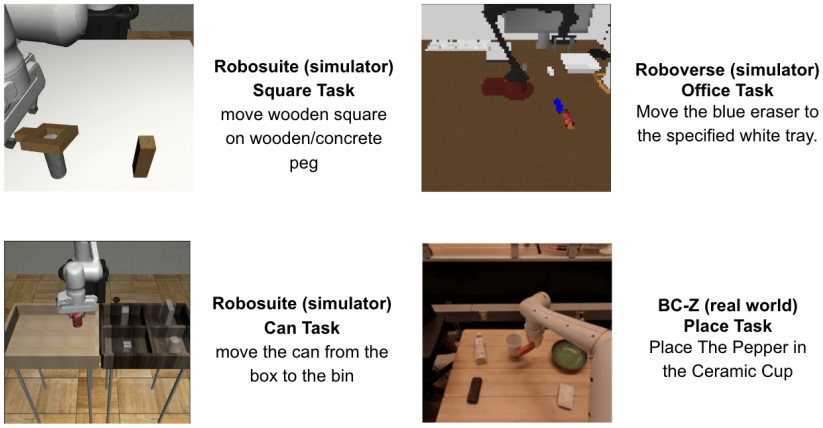

Figure 4: Illustration of our testing environments

4.2 Experimental Settings

Text-to-image object detection. We test SOTA models Owl-V1 and Owl-V2 to find the most important area for each task. Given that Owl-V2 was trained on larger and more realistic datasets, it may not be able to detect simulated objects which tend to lack surface details and have low resolution. We set the confidence threshold to 0.1 for Owl-V1 and 0 for Owl-V2 to enhance their generalization.

Table 1: Prompts for models

		CanPick	NutAssembly	Office
CLIP	Success	dark drawer with a red can	wooden square nut on wooden rod	a blue object in the white tray
	Failed	dark drawer	wooden rod	a pure white tray
Object Detection		dark drawer	wooden peg	white tray

Imitation Learning. Based on the difficulty of tasks, we train the agents with 9000, 800, 5500 epochs for CanPick, NutAssembly, and Office respectively. After that, we evaluate the agents with Success Rate on 100 trials. For better comparison, we build two baselines **All** and **Success Only**, where **All** takes all of the trajectories, including failed ones, as the input and **Success Only** takes only successful trajectories. These two settings can be regarded as the bottom and top line. Also, we have reproduced the settings from Du et al. [2023a] to compare them with our methods.

4.3 Results and Analysis

Table 2 summarizes our results* on CanPick, NutAssembly, and Office. Our method can significantly outperform baselines on both retrieval and behavior cloning. In the NutAssembly task, our method

*In early CLIP experiments, we fed the whole image instead of classifying it as a failure when no object was detected. Therefore, the reproduced result of NutAssembly may slightly differ from the one we reported.

Table 2: Main Results

	NutAssembly		CanPick		Office	
	Retrieval Accuracy	**Success Rate**	**Retrieval Accuracy**	**Success Rate**	**Retrieval Accuracy**	**Success Rate**
All	50.00%	55.00%	50.00%	47.00%	58.00%	28.00%
Behavior retrieval	-	59.00%	-	52.00%	-	50.00%
CLIP+Owlv1 (ours)	**86.00%**	69.00%	100.00%	**94.00%**	95.66%	**92.00%**
CLIP+Owlv2 (ours)	84.50%	**72.00%**	100.00%	**94.00%**	**96.91%**	83.00%
Success Only	100.00%	79.00%	100.00%	94.00%	100.00%	100.00%

shows an improvement of 13% in IL success rate. For the CanPick task, our method can reach optimal results. In the Office task, our method demonstrates a remarkable improvement of 42% in IL success rate.

It's worth noting that the retrieval results of Behavior Retrieval are not reported, since its retrieval is conducted on (state, action) units, while our methods retrieve the entire trajectory.

4.4 Experiments On Noisy Offline Dataset

In practical scenarios, offline datasets often consist of multiple tasks. To verify our method's ability on noisy offline datasets, we manually combine NutAssembly, CanPick, and Office as a mixed offline dataset. Our experiments on CanPick demonstrate the noisy level of the offline dataset does not significantly impact our performance of retrieval and imitation learning.

Table 3: Results of CanPick on the integration of NutAssembly, CanPick, and Office

	NutAssembly + CanPick + Office	
	Retrieval Accuracy	**Success Rate**
All	20.00%	49.00%
CLIP+Owlv1 (ours)	**98.20%**	**93.00%**

4.5 Can Our Method Apply To Real World?

To assess our method's applicability to real-world scenarios, we conducted retrieval experiments on a real-world dataset, specifically a subset of the BC-Z dataset (Jang et al. [2022]) that includes 865 episodes and encompasses 21 tasks. In our experiment, we focused on retrieving the task of 'placing the pepper in the ceramic cup' from this dataset. The results, as detailed in Table 5, demonstrate that our method maintains high retrieval accuracy even in complex and noisy settings.

Table 4: Results for BCZ Dataset Retrieval

	Retrieval Accuracy	**F1 Score**
All	6%	10%
CLIP+Owlv1	**95%**	**64%**

4.6 Ablation Study

In this section, we analyze two experiments, directly calculate similarity score without object detection and using VQA models to retrieve success trajectories. The results from Table 5 show that our methods outperform other methods. We hypothesize that VideoLLaMA was trained on standard VQA datasets, and therefore forcing it to predict a robot arm complete a task or not in the simulated environment is way out of domain. Similarly, since CLIP was trained on images that usually contain single object, we need to crop image first to remove other irrelevant objects for CLIP to find the relationship of text description and the image. These results prove each step from our methods is crucial and reasonable.

Table 5: Ablation study on NutAssembly Dataset Retrieval

	Retrieval Accuracy
VideoLLaMA	42%
CLIP+whole image	50%
Ours	**84.5%**

5 Conclusion

In this report, we have introduced "Zero-shot Text Behavior Retrieval," a pioneering approach that synergizes text-based guidance with vision-language models to redefine behavior retrieval in robotics. Distinct from conventional methodologies that predominantly rely on expert demonstrations, our method can adapt to any environment merely through text prompts. This integration enables the efficient retrieval of pertinent behaviors within a zero-shot learning framework, markedly enhancing the adaptability and scalability of robotic learning systems.

Our experiments, spanning simulated and real-world environments, have unequivocally demonstrated the superior performance of our approach. "Zero-shot Text Behavior Retrieval" consistently outperformed all established baselines across various scenarios, underscoring its robustness and efficacy. This remarkable performance illustrates the method's capacity to generalize across diverse tasks and environments, a notable leap over traditional imitation learning techniques.

The impact of our work on the field of robotics research is profound. By diminishing the dependency on expert-generated data and expanding the range of learning sources, our method sets the stage for more accessible, efficient, and flexible robotic systems. It exemplifies the potential of melding language comprehension with visual perception in robotics, heralding a new era of autonomous systems capable of learning and adapting in dynamic and varied settings. This research not only advances the domain of imitation learning but also contributes significantly to the broader discourse on the confluence of AI and robotics, with far-reaching implications across various industrial, commercial, and academic spheres.

References

J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. L. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. a. Bińkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf`.

A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation? In R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, and M. Kochenderfer, editors, *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pages 893–905. PMLR, 23–24 Jun 2022. URL `https://proceedings.mlr.press/v168/cui22a.html`.

M. Du, S. Nair, D. Sadigh, and C. Finn. Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets. 2023a.

Y. Du, K. Konyushkova, M. Denil, A. S. Raju, J. Landon, F. Hill, N. de Freitas, and S. Cabi. Vision-language models as success detectors. *ArXiv*, abs/2303.07280, 2023b. URL https://api.semanticscholar.org/CorpusID:257496810.

E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning, 2022.

J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

P. Mahmoudieh, D. Pathak, and T. Darrell. Zero-shot reward specification via grounded natural language. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14743–14752. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/mahmoudieh22a.html.

O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022.

M. Minderer, A. Gritsenko, A. Stone, D. W. Maxim Neumann, A. Dosovitskiy, A. Mahendran, M. D. Anurag Arnab, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby. Simple open-vocabulary object detection with vision transformers. *ECCV*, 2022.

M. Minderer, A. Gritsenko, and N. Houlsby. Scaling open-vocabulary object detection. *NeurIPS*, 2023.

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.

S. Nasiriany, T. Gao, A. Mandlekar, and Y. Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv:2203.06173*, 2022.

H. Zhang, X. Li, and L. Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. URL https://arxiv.org/abs/2306.02858.

Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.