# Vision Foundation Models for an embodiment and environment agnostic scene representation for robotic manipulation

Kevin Riou[1], Kevin Subrin[1] and Patrick Le Callet[1,2]

***Abstract*— Traditional Imitation Learning (IL) approaches often rely on teleoperation to collect training data, which ensures consistency between training and deployment action and observation spaces. However, teleoperation slows data acquisition, distorts expert behavior and data can be affected by the lack of teleoperation skills. To overcome these limitations, IL training on human demonstrations requires visual representations that are agnostic to both embodiment and environment. Recent advancements in Vision Foundation Models, such as Grounded-Segment-Anything (Grounded-SAM), offer a solution by extracting meaningful scene information while filtering out irrelevant details without manual annotation. In this work, we collected 50 human video demonstrations of a manipulation task from the RLBench benchmark. We evaluated Grounded-SAM's ability to automatically annotate objects of interest and proposed a 3D visual representation using depth maps. This representation was used to train a diffusion policy, which successfully generalized to simulated robot deployment in RLBench, despite being trained exclusively on real-world human demonstrations. Our results demonstrate that efficient training can be achieved with just 50 demonstrations and half-an-hour training time.**

## I. Introduction

Robotic manipulation learning is essential for equipping robots with complex skills without extensive programming and for transferring human expertise in tasks with hard-to-formalize decision-making rules. Two common training paradigms are Reinforcement Learning (RL) and Imitation Learning (IL). RL needs many interactions with the environment, which isn't always practical in real-world settings, and designing rewards can be more labor-intensive than programming the task directly. IL, particularly Behavior Cloning, offers a simpler alternative by training a policy from expert demonstration data without requiring interaction with the environment.

Most behavior cloning approaches collect their datasets by recording teleoperated demonstrations [14]. This is a practical scenario for the policy training, since the observation and action spaces are the same for the expert and the learner. However, this is not ideal for real-life scenarios for several reasons. Firstly, Mandlekar et al. [13] showed that the lack of skill of the expert in teleoperation can negatively impact the performance of the learner. Secondly, a policy trained on a dataset specific to one robot might not generalize well to other robots. The teleoperation process is also intrusive and time-consuming, especially for those unfamiliar with the technology, limiting real-world adoption. Several studies have attempted to address the human-to-robot IL problem, primarily relying on affordances [1], [11]. However, these approaches are not scalable when regions of interest lie outside the objects in the scene and for tasks that require multiple successive interactions with the environment.

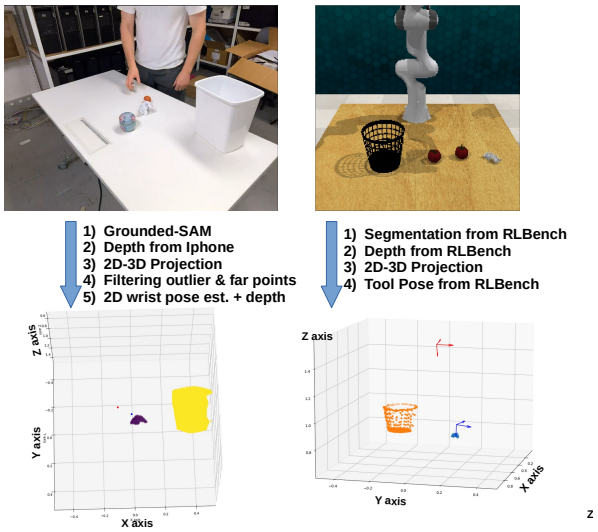


Fig. 1: Our visual representation uses open-vocabulary object detection and segmentation (Grounded-SAM) to represent a scene, focusing only on objects of interest and the hand/tool position, regardless of the operator or the environment.

Two main limitations are hindering the development of human-to-robot IL. First, the lack of public benchmarks that can provide both human demonstrations of manipulation tasks along with a publicly available simulation featuring the same tasks, which would allow the community to compare their approaches on the same tasks. Second, the **lack of visual representations that are agnostic to the embodiment and to background environment, which would allow to train a policy on human demonstrations from a given background environment, and deploy it on a robot in new environments**.

In this work, we selected one task from the publicly available simulated benchmark RLBench [6], and we collected a dataset of human demonstrations for this task. We collected 50 demonstrations of the task "put rubbish in bin", recorded using an iPhone 14 pro.

We further proposed a visual representation that **leverages recent advancements on open-vocabulary object detection [9] and segmentation [8], [17] to automatically extract objects of interest** in the scene **without human annotations**

[1]Nantes Université, Ecole Centrale Nantes, CNRS, LS2N, UMR 6004, Nantes, France. {kevin.riou, kevin.subrin, patrick.lecallet}@univ-nantes.fr
[2]Institut Universitaire de France (IUF)

from the RGB images, by only specifying the names of the objects of interest, and evaluate its performances on our data with different prompting strategies. By further leveraging the depth maps provided by the iPhones, we can recover a sparse point-cloud of the scene containing only the objects of interest, plus the position of the tool/hand. Since our point-cloud only contains **sparse but relevant information to the task at hand**, we can train a diffusion policy that achieves 20% success rate from those **50 demonstrations only**, and in a **less than 30 minutes of training**, while using the VIP state-of-the-art visual representation for robotic manipulation [12], fail to learn the task.

Overall, this work showcases the **potential of Vision Foundation Models to extract meaningfull information from the scene, enabling 0-shot transfert to new environments or new embodiments** and paves the way for the development of new benchmarks, visual representations, and learning paradigms around these problems.

## II. Dataset and annotation strategy

### A. Content

50 demonstrations of the task "put rubbish in bin" were collected using a moving iPhone 14 pro (carried by an external operator), providing RGB images and depth maps of the scene from various viewpoints. In the RLBench task, the robot is required to pick up a piece of rubbish from the table and place it in a trash bin. The rubbish is a small crumpled piece of paper, and is always accompanied by two distractors, which are other objects that the robot should not interact with. In the demonstrations that we collected, we included various distractors, but also several distinct trash-bins and pieces of crumpled paper as trashes. The set of objects present in the scene in the human demonstrations is shown in Fig. 2.

The intuition behind using a moving camera is to provide data that allow to train viewpoint agnostic deep-learning policies. If all the data are recorded in a fixed viewpoint, the trained policy will be biased towards this viewpoint, and will not generalize well to other viewpoints. On the other hand, capturing data from a moving camera allows to collect a dataset in which each image is taken from a different viewpoint, and therefore allows to train a policy that is agnostic to the viewpoint of the camera.

### B. Action annotations

An IL dataset is composed of pairs of observations and corresponding actions. The first step in the annotation process is to extract actions from the human demonstrations. On the robot side, the actions should correspond to the position and orientation that the robot gripper should reach, regarding the state of the scene. The gripper opening state after reaching the target pose is also part of the action. Therefore, the actions from the human demonstrations should similarly correspond to the position and orientation that the human hand reaches next in the scene, regarding the state of the scene.

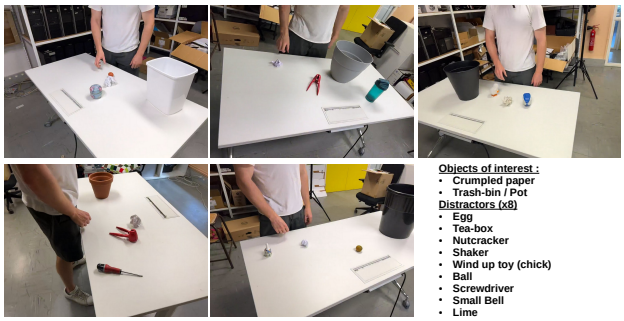


Fig. 2: Visualization of the objects present in the scene in the human demonstrations. The images arise from the moving camera, and therefore provide a sample of the viewing angles that this camera provides.

We annotated the hand pose in the collected human demonstrations using the Keypoint-Fusion [10] model, that was trained to extract the 3D position of 21 keypoints of the human hand from RGB images and depth maps. We defined the tool position as the average of the two furthest keypoints of all fingers to mitigate errors from hand pose estimation. In this pick-and-place task, the tool's orientation was kept orthogonal to the table during deployment to avoid uncertainty in orientation estimation and to focus the study on how visual representation affects policy generalization when trained on human demonstrations and applied in robot simulation.

Additionally, 3 primitives were manually annotated in all the human demonstrations: "reach and grasp", "reach and release", and "reach a point to avoid a collision". The opening state of the hand was annotated to "closed" after a grasp primitive, and "open" after a release primitive, and opened at the beginning of the demonstrations. These opening states were used to define the gripper state in the actions. The primitives that we defined in the human demonstrations are shown in the bottom part of Fig. 3.

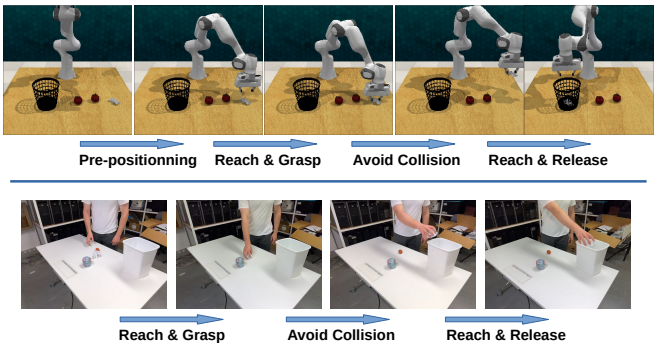


Fig. 3: Comparing robot primitives obtained from the RLBench demontration generator and human primitives annotated from the human demonstrations.

### C. Observation annotations

The second step in the annotation process is to extract a visual representation from the observations in the human

demonstrations. We annotated the objects of interest in the scene using the Grounded-SAM model [17]. It combines two models. The Grounded-Dino [9], an open-vocabulary object detection model, allows to detect objects of interest in the images by specifying their names with an input prompt. The Segment-Anything (SAM) model [8] further segments the objects detected by the Grounded-Dino model. We only segemented the first frame of the video, and subsequently tracked the segmented objects using the Cutie tracker [2]. Initially, we prompted Grounded-SAM with the names of the target objects, "crumpled paper" and "trashbin," aiming for accurate differentiation from distractors (see Fig. 2). We evaluated the model's detection performance using precision, which measures how many of the detected objects are actually of interest, and recall, which measures the number of the actual objects of interest that were correctly detected. We focused the evaluation of Grounded-Dino on detection performance, measured by precision and recall, since segmentation was nearly perfect when the correct objects were detected.

However, the initial prompt resulted in numerous false positives for "crumpled paper," with objects like a white wind-up toy being misclassified, leading to low precision (Naive Prompt line, Table I). Additionally, the trashbin was occasionally missed, causing poor recall.

To improve detection, we enhanced the prompt by including the names of all objects in the scene and filtering the results by name. This allowed Grounded-SAM to correctly categorize similar objects, such as the white wind-up toy, under different labels. We also added "pot" to the prompt to increase the likelihood of detecting the trashbin. This refinement improved detection accuracy, with 76% of the demonstrations having all target objects correctly detected without the need for human annotation (Table I). For the remaining 24%, we used the interactive segmentation feature of SAM [8].

TABLE I

| | **Trashbin** | | **Crumpled Paper** | | **Succ. Rate** |
|---|---|---|---|---|---|
| | Recall | Precision | Recall | Precision | |
| Naive Prompt | 0.84 | 0.95 | 0.9 | 0.79 | 52 |
| **Enhanced Prompt** | 0.92 | 1.0 | 0.9 | 0.92 | **76** |

The pixels corresponding to the objects of interest in the scene were then projected to the 3D space using the depth maps and known intrinsic parameters of the iPhone 14 Pro camera. This process generates a point cloud representation of the scene, containing only points from objects of interest. Each point is a 4D vector with its 3D position in the camera coordinate system and a scalar indicating the object's category ("trash" or "trashbin"). This representation filters out the background, the operator, and distractors (objects in the scene the robot shouldn't interact with). The tool's position (human hand) is also provided to the policy to help locate the operator in the scene.

Finally, when the camera moves, its coordinate system shifts, causing a mismatch between the future tool pose (action) and the current scene observation (4D point cloud). To address this, ground-truth actions are projected into the camera's coordinate system at the time of the observations using the iPhone's odometry data.

## III. Primitive based behavior cloning

*1) Behavior cloning in fixed horizon settings:* We formalize our dataset as a set of demonstration trajectories $D = \{\tau_i\}_{i=1}^{N}$, where each trajectory $\tau_i$ is defined as a sequence of observation-action pairs $\tau = \{(o_t, a_t)\}_{t=1}^{T_i}$. Training a policy $\pi$ using behavior cloning on a fixed action horizon of 1 time step, and equipped with a visual representation function $\phi$ is equivalent to solving the optimization problem defined by Equation 1.

$$\theta^* = \arg\min_{\theta} \sum_{i} l\left(\pi\left(\phi(o_i); \theta\right), a_i\right) \tag{1}$$

In Equation 1, $\theta$ represents the learnable parameters of the deep learning policy and $l$ is the loss function that seeks to minimize the difference between the predicted action $\pi\left(\phi(o_i); \theta\right)$ and the ground truth action $a_i$. Here $a_i$ can be fully defined as the tuple

$$a_i = (x_{tcp}^{i+1}, y_{tcp}^{i+1}, z_{tcp}^{i+1}, Grip.State^{i+1})$$

, where $(x_{tcp}^{i+1}, y_{tcp}^{i+1}, z_{tcp}^{i+1})$ is the 3D position of the robot's tool center point (TCP) at frame i+1 (next frame). As mentionned earlier, the orientation of the gripper will be fixed, orthogonal to the table for the considered pick-and-place task. $Grip.State^{i+1}$ is the opening state of the robot's gripper (open/closed).

The policy can be trained to predict not just the next action, but the next "h" actions, enhancing its planning capabilities. Chi et al. [2] extended this by training a diffusion policy to predict the next "h" actions but only executing the first "a" actions, balancing long-term planning with reliable short-term execution. After a hyperparameter search, they found that "h=16" and "a=8" worked best for tasks using a transformer-based policy trained on teleoperated demonstrations.

*2) Behavior cloning in primitives based settings:* In the case of primitives based actions [5], [7], the trajectories can be reformulated as $\tau' = \{(o_i, p(i))\}_{t=1}^{T}$., where $p(i) = a_{min(k>i, \forall k \in \{k_1, k_M\})}$ represents the 3D position and opening state of the gripper at the end of the ongoing primitive that is being performed in the scene at time $i$.

$\{k_1, k_M\}$ is the set of timesteps that correspond to the end of the primitives, if we have M primitives in the demonstrations. Regarding the proposed data, M=3, with "reach and grasp", "reach and release", and "avoid collision" primitives.

Training a policy to predict actions in the primitive-based setting is equivalent to solving the optimization problem defined by Equation 2.

$$\theta^* = \arg\min_{\theta} \sum_{i} l\left(\pi\left(\phi(o_i); \theta\right), p(i)\right) \tag{2}$$

*3) The challenges of the "avoid collision" primitive:* The "avoid collision" primitive can confuse the policy because it doesn't involve direct interaction with objects of interest. Without a clear signal, like a gripper change, the policy may struggle to recognize the end of this primitive and fail to transition to the next one. This issue is worsened by the variability in human demonstrations, where multiple paths can be taken to avoid obstacles, increasing uncertainty.

We propose an alternative solution that focuses on the "reach and grasp" and "reach and release" primitives but introduces an intermediate point between the start and end positions. The action is represented by a 7-dimensional vector: the first 3D position avoids obstacles, the second 3D position targets the grasp or release position, and the gripper state changes only after reaching the second position. This approach ensures the policy passes through the avoid point while still achieving the final grasp or release point anyway. When their's nothing to avoid, the avoid point is set halfway between the start and the end of the primitive. This second solution is denoted as "two-step keypoint" in the result tables.

*4) Implementation details and evaluation metrics:* The demonstrations were split 80% for training and 20% for validation. We trained the transformer-based diffusion policy from Chi et al. [3], with the configuration they used for the "low dim push-t task", using either our 4D point cloud or Value-Implicit Pre-training (VIP) [12], which has shown superior performances for robotic manipulation compared to prior pre-trained visual representations [15], [4], [16]. The results of train and validation error on the prediction of the next tool position are reported for each training setting. Additionally, the position errors for the "avoid collision" primitives only, are also reported to quantify the uncertainty on the corresponding keypoints. These are denoted as "Train/Eval Av. Pos. Err." in the results tables. All policies are trained for 3000 epochs. Every 1000 epochs, the policy is evaluated on the validation set and deployed in simulation. In the simulation, 50 rollouts are performed using front and overhead cameras, and the success rate is calculated as the number of successful rollouts out of the total. Since rollouts are conducted after 1000, 2000, and 3000 epochs, we report the best success rate obtained, along with corresponding train/val position errors.

## IV. Human to robot performances

In Table II, we observe that fixed-horizon action prediction ("Frame") results in lower position accuracy compared to Keypoint-based actions ("Two-Step Keypoints"), leading to a 0% success rate. This may be due to high uncertainty in hand pose estimation, possibly exceeding the distance between successive hand positions.

Regarding the Two-Step Keypoints, while the VIP representation achieves similar tool position accuracy to our 4D Pt.Cl. representation, it fails to generalize to simulated robot deployments. In contrast, the 4D Pt.Cl. representation achieves a 20% success rate.

Table III demonstrates that using two-step primitives significantly improves deployment performance. In contrast, treating "avoid collision" as an independent primitive often led the policy to get stuck around the "avoid collision" keypoint. Additionally, the choice of human pose estimation method is crucial, improving the success rate by 4% compared to simply projecting 2D poses with depth maps. Adding a pre-positioning primitive before each grasp further enhances deployment success.

Finally, all models however fail to generalize to the overhead view, which contains strong self-occlusions with the robot.

| | | Train Pos. Err. | Train Av. Pos. Err. | Val Pos. Err. | Val Av. Pos. Err. | Suc.Rate Front | Suc.Rate Overhead |
|---|---|---|---|---|---|---|---|
| Two-Step Keypoints | VIP | 8.4 | 10.5 | 283.0 | 263.0 | 0 | 0 |
| | 4D Pt.Cl. | 9.0 | 10.0 | 128.0 | 167.0 | **20** | 0 |
| Frame | VIP | 27.2 | 22.4 | 154.3 | 154.3 | 0 | 0 |
| | 4D Pt.Cl., h=1, a=1 | 28.4 | - | 54.0 | - | 0 | 0 |
| | 4D Pt.Cl., h=16, a=8 | 261.0 | - | 328.1 | - | 0 | 0 |

TABLE II: Training on human demonstrations, deploying on simulation. Hand pose obtained with Keypoint-Fusion model. Using 2-step primitives and prepositionning.

| Kpts type | Pre-Grasp Positions | Pose est. method | Train Pos. Err. | Train Av. Pos. Err. | Val Pos. Err. | Val Av. Pos. Err. | Suc. Rate Front | Suc. Rate Overhead |
|---|---|---|---|---|---|---|---|---|
| 1-step | No | 2D + depth | 11.8 | 13.6 | 188.1 | 221.4 | 0 | 0 |
| **2-step** | No | 2D + depth | 13.8 | 18.0 | 128.3 | 171.0 | 14 | 0 |
| **2-step** | **Yes** | 2D + depth | 22.2 | 30.7 | 126.0 | 165.4 | 16 | 0 |
| **2-step** | **Yes** | **RGB-D model** | 9.0 | 10.0 | 128.0 | 167.0 | **20** | 0 |

TABLE III: 4D Pt.Cl. model, moving camera

## V. Conclusion

The Grounded-SAM model successfully detected all objects of interest in 76% of the 50 demonstrations in our dataset. While this performance is insufficient for direct deployment, it could be adequate for fine-tuning a lighter segmentation model in a few-shot manner [20], [18], [19] using images from those successful examples. The 4D Pt.Cl. representation used as input for a diffusion policy achieved a 20% success rate when trained on real-world human videos and deployed on a simulated robot, despite significant embodiment and environment shifts. It would be valuable to explore the impact of incorporating these successful examples into the training data to provide the model with target domain samples, all without requiring a teleoperation phase, and assess the effects on deployment performance. Additionally, data from alternative viewpoints—including a fixed camera and an egocentric perspective—were collected alongside the moving camera for these 50 demonstrations. Future work could investigate the benefits of these different viewpoints. Lastly, a crucial area for future research is evaluating the model's ability to predict gripper orientations, starting with simulated data.

## References

[1] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.

[2] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3151–3161, 2024.

[3] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[5] Stephen James and Andrew J Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022.

[6] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[7] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[9] 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.

[10] Xingyu Liu, Pengfei Ren, Yuanyuan Gao, Jingyu Wang, Haifeng Sun, Qi Qi, Zirui Zhuang, and Jianxin Liao. Keypoint fusion for rgb-d based 3d hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3756–3764, 2024.

[11] Manuel Lopes, Francisco S Melo, and Luis Montesano. Affordance-based imitation learning in robots. In *2007 IEEE/RSJ international conference on intelligent robots and systems*, pages 1015–1021. IEEE, 2007.

[12] 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.

[13] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[14] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[15] 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.

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

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

[18] Kevin Riou, Jingwen Zhu, Suiyi Ling, Mathis Piquet, Vincent Truffault, and Patrick Le Callet. Few-shot object detection in real life: case study on auto-harvest. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2020.

[19] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 9197–9206, 2019.

[20] Xin Wang, Thomas Huang, Joseph Gonzalez, Trevor Darrell, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning*, pages 9919–9928. PMLR, 2020.