

# Supplementary Material

## 1 Introduction

We provide additional material that supports our paper.

- We invite the reviewers to watch the demo video about the proposed pretext task (goal-image prediction), Omni-Object Pick-and-Place (OOPP) dataset, and real and simulated robot manipulation.
- In Sec. 2, we provide more details about our method, including our backbone and different output heads utilised in different downstream tasks.
- In Sec. 3, we describe more details of each downstream task, including the real-robot experiment.
- In Sec. 4, we provide additional examples of the proposed pretext task, goal-image prediction, and our OOPP dataset.

## 2 Method details

**Backbone architecture.** We adopt ViT-Base [1] with a patch size of 16 as our backbone. The original ViT-Base architecture consists of 12 attention blocks with an embedding dimension of 768, and its the decoder typically comprises 8 attention blocks with an embedding dimension of 512. For a fair comparison with other methods using ViT-Base as a backbone, we include 6 attention blocks and 6 bi-directional attention blocks in our encoder. In the pre-training stage, our goal-image prediction head is a one-layer fully connected network.

**Affordance head.** Our affordance head for generating  $SE(2)$  robot actions (the 2D location and 1D rotation) for tabletop manipulation tasks (*Ravens*, *OOPP*, and our real robot experiments) consists of 4 convolutional layers with skip connection to convert the output from the transformer backbone to the affordance map. To generate a rotation angle, we expand a single affordance map into 36 instances, where each instance represents a 10-degree step. We apply the softmax function to the expanded affordance map, which identifies the position and corresponding rotation angle. Since other pre-training methods have not been previously evaluated on this benchmark, we reimplemented them and used a unified action head across all pre-training baselines. For other Pick-and-Place methods, we retain their original action heads which are structurally similar but typically deeper due to their use of ResNet backbones. In addition, since our pre-training methods are designed to predict the goal image, we found that concatenating the goal image into the convolutional layers facilitates improved affordance generation, as the two tasks are closely correlated.

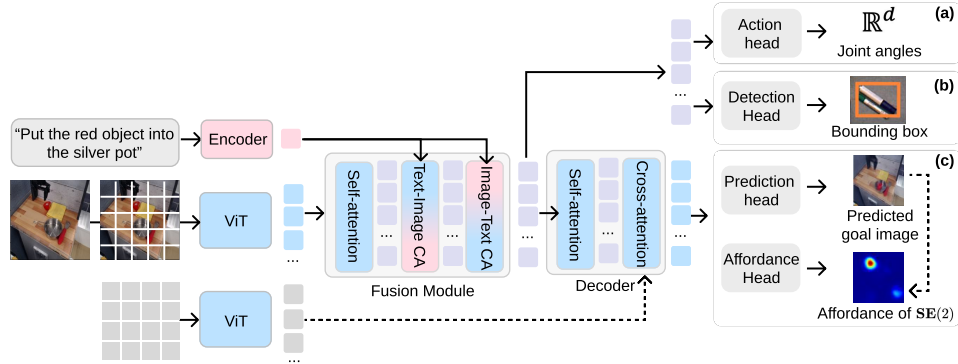


Figure 1: **Downstream heads for each task.** We predict joint angle velocities for the *Franka Kitchen* benchmark (a), bounding boxes for the *Referring Expression Grounding* task (b), and  $SE(2)$  affordances for *Ravens* and *OOPP* tasks (c). We observe that it is beneficial to keep the trained goal prediction heads for affordance prediction.

30 **Action head and detection head.** Following the baseline established by Voltron [2] and MPI [3], we  
 31 adopt the same shallow MLP policy network to predict joint velocities of  $\mathbb{R}^9$  (7 degree of freedom  
 32 and two for grasp status) for robot actions in the Franka Kitchen benchmark, and to regress bounding  
 33 boxes in the Referring Expression Grounding benchmark, as shown in Fig. 4. We use features after  
 34 the fusion module for two main reasons: First, for fair comparison: previous methods, including  
 35 R3M [4], MVP [5] and Voltron [2], only evaluate frozen representations dropping the decoder part.  
 36 Since we directly report their results on these benchmarks, we adopt a consistent setting. Second,  
 37 based on task requirements, both benchmarks rely on understanding the current state or the next  
 38 state, while features after the decoder in our model represent the goal image and correspond to the  
 39 final state. Therefore, using features before the decoder is more suitable for these two benchmarks.

### 40 3 Benchmark details

#### 41 3.1 Ravens

42 **Overview.** Ravens is a simulated benchmark to evaluate tabletop pick-and-place robot manipulation  
 43 tasks. We use PyBullet OpenAI Gym [6] based on the configuration described in CLIPort [7].  
 44 We chose 8 language-conditioned tasks for the experiment, as shown in 2, including *Packing seen*  
 45 *or unseen Google objects sequence*, *Packing seen or unseen Google objects group*, *Put block in*  
 46 *the bowl*, *Stack blocking pyramid*, *Towers of Hanoi*, *Packing boxes pairs*, *Assembling kits*, and  
 47 *Separating piles*. Details of each task, including the train and test split of objects, and the language  
 48 instruction template, can be referred to [7]. Note that we did not split the tasks according to seen or  
 49 unseen colours, as all the methods have already “seen” all the colours in their pre-train models or the  
 50 unsupervised pertaining phase. Therefore, we combine both “seen” and “unseen” splits of colours  
 51 into a single task and the scores just reflect all model’s perception ability on all the colours.

52 **Evaluation details.** We evaluated the capability of the proposed methods on multi-task experiments  
 53 in the benchmark. Specifically, we trained the model using 1,000 demonstrations drawn from all  
 54 task categories and assessed performance on another 100 test demonstrations per task. Since prior  
 55 pre-training methods [3, 2] have not been evaluated on this benchmark, we reimplemented their  
 56 models using the original codebases and pre-trained them on the same pre-training data as our ap-  
 57 proach. When adapting to the downstream tasks, these pre-training methods were also equipped  
 58 with the affordance head described in Section 2. Subsequently, all baselines, including the Pick-

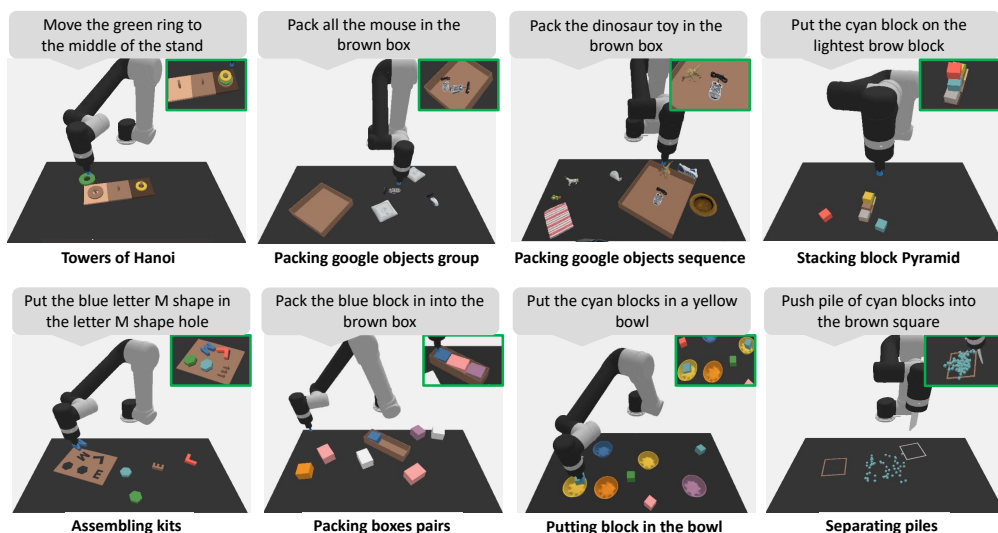


Figure 2: **Examples of eight robot manipulation tasks in the simulator.** The language instructions are on the top of each image and the final states are shown in the green box.



Figure 3: **Tasks in the Franka Kitchen benchmark.**

and-Place baselines, were fully fine-tuned on the same set of downstream demonstrations, following the evaluation protocol outlined in [7], to ensure a fair comparison.

### 3.2 Franka Kitchen

**Overview.** Franka Kitchen [8] is a well-established benchmark for evaluating the efficacy of visual representations in facilitating the learning of visuomotor control policies from limited demonstrations. This benchmark comprises five distinct visuomotor control tasks, as shown in the Fig. 3, each captured from two camera viewpoints.

**Evaluation details.** We use the action head described in Sec. 2 for predicting joint velocities. As prior works [2, 3, 4, 5, 9], which leverage supervised or unsupervised pre-training for robot manipulation, commonly adopt this benchmark, we directly report the results stated in their original papers and compare our method against these approaches. Following the evaluation protocols widely adopted in these works, we trained the action head with the backbone frozen using 25 demonstrations, and report average success rates across five tasks, two viewpoints, and three random seeds.

### 3.3 Referring Expression Grounding

**Overview.** The goal of this task is to predict a bounding box of an object in a cluttered scene based on the nature language expression. This task offers the evaluation of language-conditioned scene understanding and object recognising ability, which serves as an important prerequisite for language-based robot manipulation. The benchmark is based on OCID-Ref Dataset [10], which provides representatives scenes in robotised settings. The benchmark also provides splits based on the clutter level.

**Evaluation details.** We use a shallow MLP as detection head as describe in Sec. 2 to regress the bounding box directly. We use the evaluation codebase provided by [2]. Similar to Franka Kitchen, we report the results stated in the paper for each baseline [2, 3, 4, 5, 11]. The evaluation metrics are the average precision at 0.25IoU under each clutter level.



Figure 4: **Examples of the task of Referring Expression Grounding.** This task offers varied language instructions involving a wide range of objects and scenes.

Table 1: **Real robot tasks settings.** We present the language template, variable factors and the success condition for each real robot task

Task name	Language template	Variable factors	Success condition
Stacking blocks	Stack the {color} block on the {color} blocks	Block color and position (pick/place)	Correct block stacked on target block
Folding cloth	Fold the cloth from {direction} to the {direction}	Cloth color, initial orientation	Grasp and fold directions match the template
Packing objects	Pick the {object} into the {object}	Object type, distractor objects	Target object placed into correct container
Opening drawer	Pull out the drawer	Drawer position	Drawer fully opened
Press button	Press the {color} button	Button color, button location	Correct button touched
Aligning rope	Align the rope from {direction} to the {direction}	Rope position, direction variation	Rope aligned from start to target direction
Packing blocks	Put the {color} block into the {color} bowl	Block color, bowl color	Correct block placed into matching bowl
Pushing piles	Push the pile of {objects} to the {color} area	Object color, target area color	Pile reaches target area within five attempts



Figure 5: **Real robot experiment.** The figure shows our physical robot setup along with both seen and unseen objects. Seen and unseen colored blocks are also included.

### 3.4 Real robot experiments

**Overview.** We validate the applicability of our method in real-world scenarios. We validate our model on 10 manipulation tasks: *Stacking blocks*, *Folding cloth*, *Packing objects*, *Opening drawer*, *Pressing button*, *Aligning rope*, *Packing blocks*, and *Pushing piles*. Each task contains 5 different scenarios that differ in either objects or locations. We manually collected 200 training demos that contain robot image pairs, language descriptions, and annotations for real-world fine-tuning. Five colored blocks and five unseen objects were excluded from the training demonstrations, and no demonstrations from the *Opening drawer* or *Pushing piles* tasks were included, although similar tasks are present in the images used in the self-supervised pre-training phase. This design aims to evaluate whether the model can effectively generalised from the self-supervised pre-training, rather than relying on downstream demonstrations.

**Evaluation details.** Fig. 5 shows our real-robot environment and objects. We trained both our model and CLIPort [7] on our manually collected training demos. We utilise a 6-degree-of-freedom (6-DoF) UR5 robotic arm, Robotiq 2F-85 two-finger gripper, and an Intel RealSense RGB-D camera for our real-world experiments. We capture the top-down RGB observation which covers the workspace of  $60 \text{ cm} \times 30 \text{ cm}$ , and the image from the camera is resized to  $320 \times 160$  pixels.

**Task details.** Here we show the language template, variable factors and the success condition for each real robot task in Tab. 1. Images for each task could be referred in the main paper.

## 4 Additional details

**Goal-image prediction.** We provide more qualitative examples for the goal-image prediction and the results during the training of masked auto-encoders, as shown in Fig. 6 and Fig. 7. In Fig. 6 we show examples of the same input image but with different language instructions. The results show that our model can effectively predict different goal states given different language-based instructions and the initial observation. Namely, these results show that our model can interpret the input instructions, factorize different objects that need to be manipulated, and further understand the



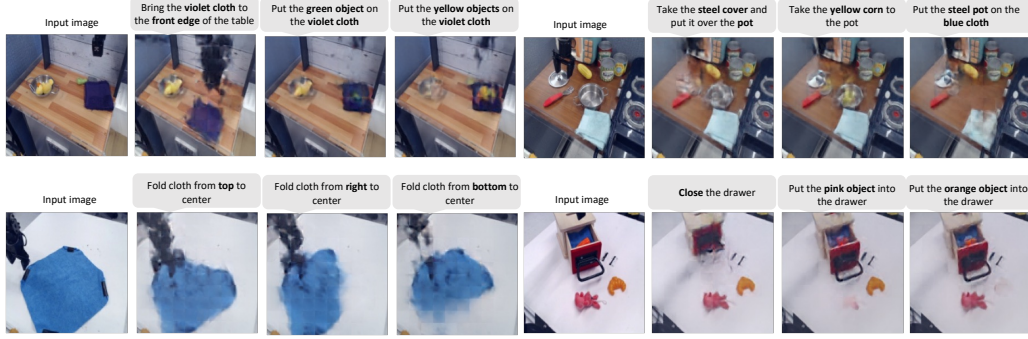


Figure 6: **Examples of goal prediction images.** Given the same input image with a different language, the model effectively understands the diverse language instructions and predicts goal images aligning with the semantics. Notably, all images belong to the test set and are novel to the model.

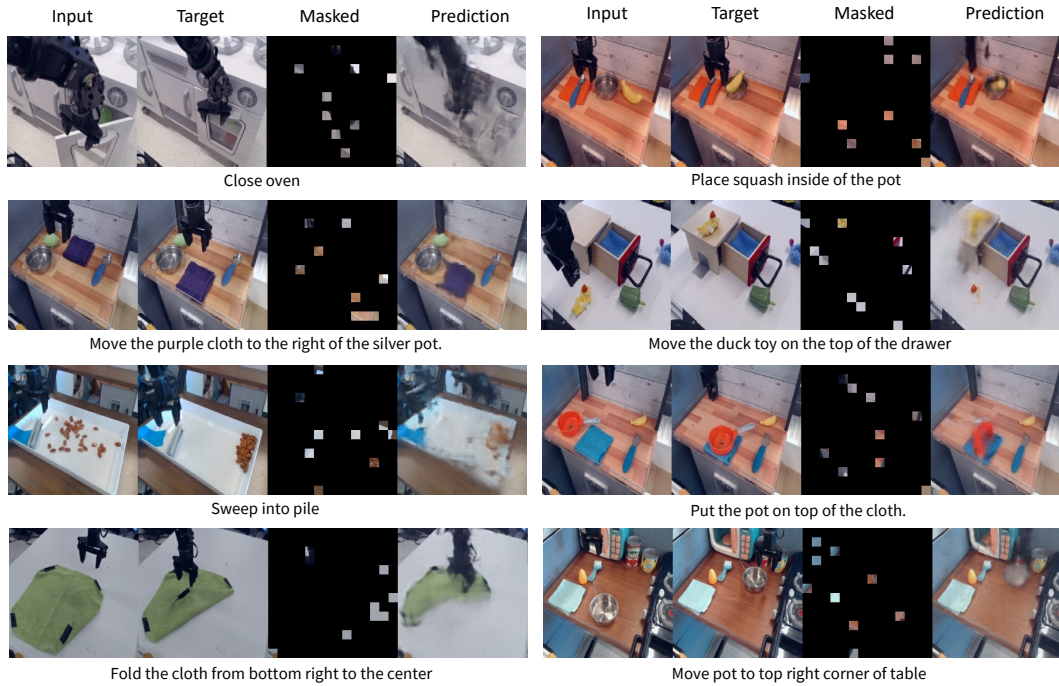


Figure 7: **Examples of masked image training.** We show the qualitative examples of our masked auto-encoders. The text descriptions are shown below each sample.

108 spatial location or direction in the scene. This indicates the learned visual-action representations  
 109 after the self-supervised learning with the pretext task effectively associate visual states with action.  
 110 In Fig. 7, we present example results in our pre-training phase. Our predicted goal images are  
 111 blurry as in other MAE-based methods [12, 13, 14]. These results demonstrate that our approach  
 112 successfully learns to predict the goal image.

113 **Dataset details.** We build our Omni-Object Pick-and-Place (OOPP) dataset upon the previous  
 114 benchmarsk [7, 15] in the PyBullet Gym environment [6]. We manually selected 180 real-scanned  
 115 object classes from the OmniObject3D dataset [16], focusing on those suitable for robotic manipula-  
 116 tion, resulting in a total of 3,200 distinct instances. For each object, we reduced the mesh resolution  
 117 to 20K faces to enable efficient rendering in simulation. Additionally, we filtered out objects that  
 118 were either too large or too small i.e., with dimensions less than 4cm or greater than 40cm). Our  
 119 dataset includes full robot manipulation episodes, comprising annotated robot actions, language in-

structions, and simulation-generated rewards. We use 160 object classes for training. For evaluating intra-class generalisation, we hold out a subset of instances from 20 categories included in the training set. For inter-class generalisation, we reserve 20 object categories that are entirely unseen during training. Fig. 8 illustrates the first 50 examples across different object classes, while Fig. 9 highlights examples of intra-class variation. The held-out categories for inter-class generalisation are sampled from four high-level semantic groups, as detailed in Table 2.

Table 2: **Semantic group division of seen and unseen classes**We divided all objects into four semantic groups. We selected unseen classes from each group in proportion to the number of object classes it contains, ensuring an even distribution of unseen categories.

Semantic Group	Seen Classes	Unseen Classes
<b>Food</b>	red_jujube, corn, strawberry, anise, pizza, longan, loquat, chocolate, brussels_sprout, haw_thorn, green_bean_cake, cucumber, litchi, cake, dumpling, mooncake, rice_cake, puff, water_chestnut, mushroom, egg, broccoli, pastry, egg_tart, kiwifruit, fig, cheese, chili, tomato, lemon, oyster, steamed_bun, carrot, mangosteen, bread, ginger, waffle, bun, peach, apple, pear, potato, zongzi, pomegranate, sweet_potato, onion, banana, chicken_leg, sausage, coconut, broccolini, hami_melon, durian, asparagus, walnut, mango, loquat, bucket_noodle	orange, biscuit, shrimp, garlic, donut, sweet_potato, candy, cherry, pancake
<b>Daily-use</b>	thimble, beauty_blender, battery, candle, calculator, plug, watch, nipple, power_strip, bottle, medicine_bottle, tissue, belt, dish, flash_light, canned_beverage, fork, cup, glasses_case, bowl, tape_measure, speaker, laundry_detergent, teapot, glasses, wallet, insole, bumbag, fan, knife, umbrella, kettle, light, picnic_basket, hammer, shoe, hat, laptop, vase, ornaments, spanner, book,	soap, mouse, scissors, teapot, shampoo, toothpaste
<b>Entertainment</b>	toy_boat, toy_plant, toy_car, toy_bus, toy_plane, timer, whistle, table_tennis_bat, toy_motorcycle, drum, remote_control, toy_animals, garage_kit, china, chess, Chinese_chess, rubik_cube, dinosaur, doll	toy_bus, teddy_bear, toy_animals
<b>Others</b>	hairpin, lotus_root, house (model), plant, dumbbell, package, bamboo_shoots, brush, flute, ornaments, conch, magnet, box	flower_pot, red_wine_glass

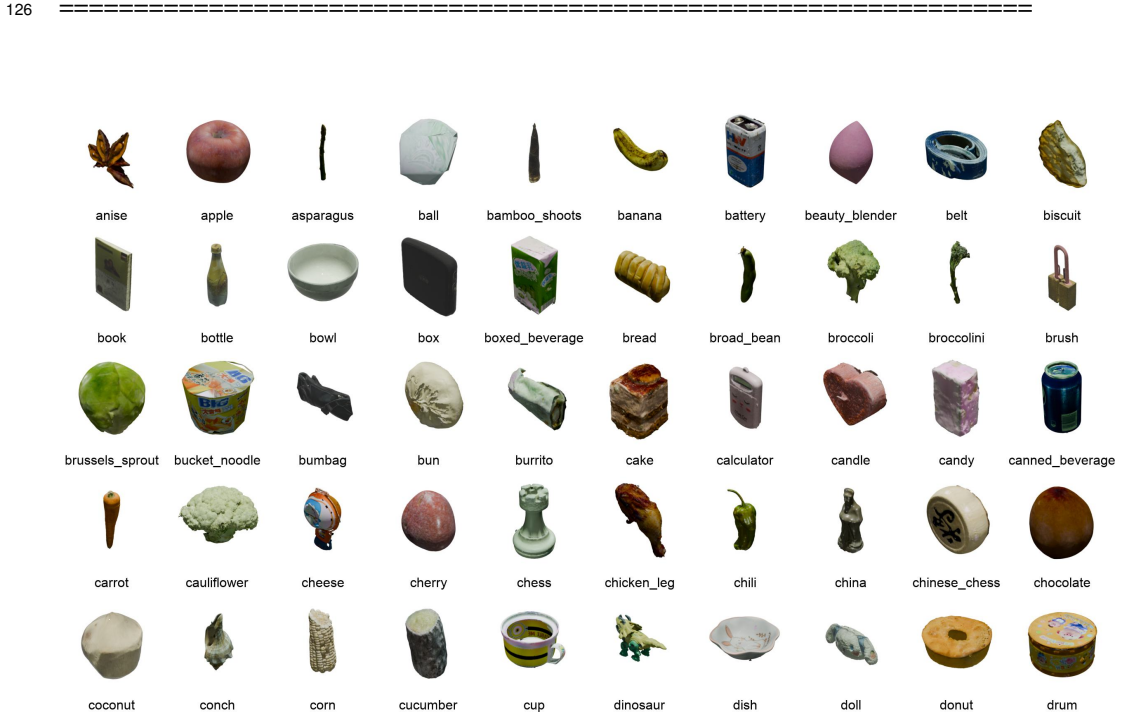


Figure 8: Examples of objects from different classes.



Figure 9: Examples of intra-class variance.

## References

- [1] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [2] S. Karamcheti, S. Nair, A. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. In *Robotics: Science and Syst.*, 2023.
- [3] J. Zeng, Q. Bu, B. Wang, W. Xia, L. Chen, H. Dong, H. Song, D. Wang, D. Hu, P. Luo, et al. Learning manipulation by predicting interaction. *Robotics: Science and Syst.*, 2024.
- [4] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. In *Conf. Robot Learning*, pages 892–909, 2023.
- [5] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Conf. Robot Learning*, pages 416–426, 2023.
- [6] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. <http://pybullet.org>, 2016–2021.
- [7] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conf. Robot Learning*, pages 894–906, 2022.
- [8] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conf. Robot Learning*, 2020.
- [9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proc. Int. Conf. Machine Learning*, pages 8748–8763, 2021.
- [10] K.-J. Wang, Y.-H. Liu, H.-T. Su, J.-W. Wang, Y.-S. Wang, W. Hsu, and W.-C. Chen. Ocid-ref: A 3d robotic dataset with embodied language for clutter scene grounding. In *Association for Computational Linguistics (ACL)*, 2021.
- [11] S. Chen, R. Garcia, I. Laptev, and C. Schmid. SUGAR: Pre-training 3d visual representations for robotics. In *Conf. Comput. Vis. Pattern Recognit.*, 2024.

- 152 [12] S. He, T. Guo, T. Dai, R. Qiao, C. Wu, X. Shu, and B. Ren. VLMAE: Vision-language masked  
153 autoencoder. *arXiv preprint arXiv:2208.09374*, 2022.
- 154 [13] P. Weinzaepfel, V. Leroy, T. Lucas, R. Brégier, Y. Cabon, V. Arora, L. Antsfeld, B. Chidlovskii,  
155 G. Csurka, and J. Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-  
156 view completion. 2022.
- 157 [14] A. Gupta, J. Wu, J. Deng, and F.-F. Li. Siamese masked autoencoders. *Advances in Neural*  
158 *Information Processing Systems*, 36:40676–40693, 2023.
- 159 [15] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu,  
160 and L. Fan. VIMA: General robot manipulation with multimodal prompts. In *Proc. Int. Conf.*  
161 *Machine Learning*, 2023.
- 162 [16] T. Wu, J. Zhang, X. Fu, Y. Wang, L. P. Jiawei Ren, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin,  
163 and Z. Liu. OmniObject3D: Large-vocabulary 3D object dataset for realistic perception, re-  
164 construction and generation. In *Conf. Comput. Vis. Pattern Recognit.*, 2023.