

Supplementary Material for “LLARVA”

Here, we provide additional information about our model’s emergent properties, our constructed dataset, and implementation details. Specifically, Section A provides additional experiment results, Section B provides details about our constructed vision-action instruction dataset, and Section C provides additional implementation details.

A Additional Experiment Results

A.1 Additional Experiments

Method	open drawer	meat off grill	turn tap	put money	Task push buttons	sweep dustpan	slide block	close jar	screw bulb
<i>3-D methods</i>									
C2FARM-BC	20	20	68	12	72	0	16	24	8
PERACT	80	84	80	44	48	56	72	60	24
<i>2-D methods</i>									
Image-BC (CNN)	4	0	8	4	0	0	4	0	8
Image-BC (ViT)	0	0	16	0	0	0	0	0	16
LLARVA	60	80	56	44	56	84	100	28	8
Method	place wine	reach and drag	stack blocks	put in drawer	Task sort shape	insert peg	stack cups	put in cupboard	place cups
<i>3-D methods</i>									
C2FARM-BC	8	24	0	4	8	4	0	0	0
PERACT	12	68	36	68	20	0	0	16	0
<i>2-D methods</i>									
Image-BC (CNN)	0	0	0	8	0	0	0	0	0
Image-BC (ViT)	0	0	0	0	0	0	0	0	0
LLARVA	12	52	0	0	0	0	0	0	0

Table 4: **Success rate (%) on RLbench Multi-Task setting.** We fine-tuned (with visual trace prediction) the pre-trained model on 18 tasks and evaluate with 25 episodes per task. Each evaluation episode is scored either 0 for failure or 100 for success. We gray out methods with 3-D information.

Complete Simulation Results on RLbench. Following [16, 7], we evaluate LLARVA on 18 tasks in RLbench, with the comprehensive results presented in Table 4. LLARVA demonstrates significant improvements over 2-D methods and shows comparable performance to 3-D methods in most tasks. However, for “long horizon” tasks, which are more complex and involve multiple sub-steps or extended durations, LLARVA exhibits similar limitations to other methods. Specifically, for the “place cups” task, where success is defined by placing a specified number of cups on a rack, our experiments reveal that the model often successfully places the first cup but then becomes confused and wanders randomly. As discussed in the main paper, while the introduction of 2-D visual traces provides the model with a rudimentary sense of memory (as seen in the “push buttons” case), the length of this memory remains limited along the temporal axis. This issue can be partially addressed by incorporating additional conditions that include information from more previous steps, however, it will place greater demands on the maximum context length that vision-language models (VLMs) can handle. Fortunately, several recent works, such as [57], have demonstrated promising solutions and performance for VLMs with long context length.

Additional Explorations for Behavior-specific Tasks. In order to provide a more comprehensive evaluation of LLARVA, we explored additional tasks in RLbench with specific behavior patterns. The results for 5 additional tasks are presented in Table 5, using the same fine-tuning and evaluation settings as in the main paper. These 5 tasks are categorized into two types: “Bending

Task	Bending Task		Placement Task		
	toilet seat down	close laptop lid	put knife	put umbrella	move hanger
Success rate	96	68	40	4	88

Table 5: **Evaluation results on more tasks in RL Bench.** We explore additional tasks in RL Bench, which can be further categorized into “Bending Task” (*i.e.* the robot arm is supposed to grab the target object then bend and move the target down) and “Placement Task” (*i.e.* the robot arm is supposed to grab the target object, hold and move it to a specified area).

Task” and “Placement Task”. In “Bending Tasks”, the robot arm is required to grab the target object and move it down to a certain height. LLARVA demonstrates excellent performance on these tasks. In “Placement Task” the robot arm must grab the target object and move it to a pre-specified area. LLARVA performs well overall, except in cases requiring delicate operations during either the “grab” or “place” stages. For example, in the “put knife” task, most failures occur because the gripper misses the thin and delicate handle of the knife. Conversely, in the “put umbrella” task, most failures occur during the “place” stage, as the umbrella stand has a very small hole requiring precise positioning of the gripper during inserting. These issues are primarily due to the lack of detailed information from the visual observation, given that LLARVA uses only a single view image with a 128x128 resolution. Our future work will try to enable our VLM to process multiple view images or to adapt it with a more informative vision encoder that can better capture task-related features.

A.2 Emergent Properties

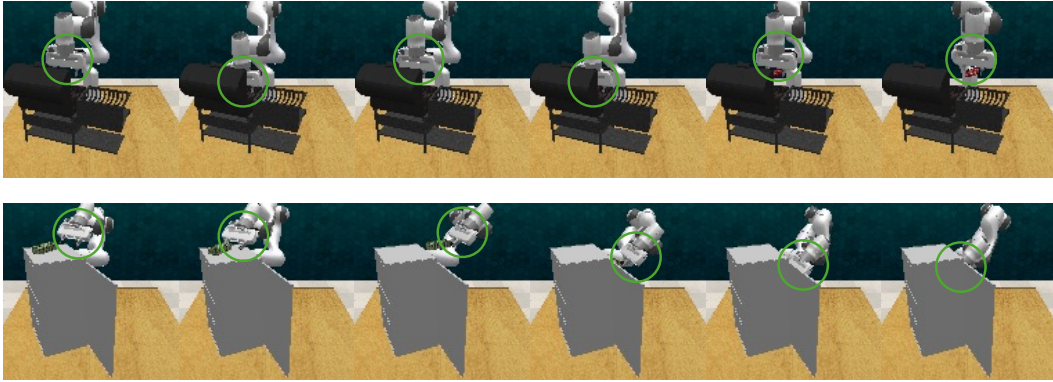


Figure 4: **Emergent Properties.** **Top:** The task in this episode is “take the steak off the grill”. As we see in the third image, LLARVA fails to pick up the steak in the first attempt. However, it tries again and succeeds the second time, showing capability of attempting a task multiple times. **Bottom:** The single view which is used as the model input shows the top and back of a safe. The task is for the robot to move the money stack from the top of the safe to one of the shelves inside the safe. LLARVA can still move the money to the correct shelf in the safe despite the camera view not showing the shelves.

Multiple Attempts after Failure. The top row of Figure 4 shows an example of the model failing to pick up an object, and then retrying as soon as the end-effector comes back into view without the object in its grasp. Specifically, the third image represents the moment where the end-effector becomes visible again, and LLARVA then attempts to complete the task again. This behavior emerges from the fact that the instruction prompt fed into LLARVA at this moment is similar to the prompt at the start of the first attempt, with the main difference being the previous actions/positions included in the prompt. We highlight this as an interesting emergent property since the training data does not include any examples with such behavior.

535 **Handling Obstructed Views.** In both pre-training and fine-tuning stages of LLARVA, we use only
 536 a single camera view to provide visual inputs. Using only 2-D inputs creates a challenge since
 537 robotics tasks require very accurate action predictions in three dimensions.

538 The model should be able to see the exact location of the target and also have a sense of depth,
 539 which other works typically achieve by using either multiple camera views or 3-D representations.
 540 We note that our model uses a single camera view due to input limitations of current open-source
 541 LMMs. These limitations can certainly be overcome and are left for future work.

542 Using a single camera view presents further challenges when objects in the scene occlude each
 543 other. However, we find that LLARVA can often complete tasks even in these occluded situations.
 544 For example, as shown in the second row of Figure 4, the task is “*put the money away in the safe*
 545 *on the top shelf*”. The camera view only shows the top and back side of the safe, which is enough
 546 information to pick up the money from the top of the safe. However, the top shelf of the safe is not
 547 visible, and LLARVA can still predict the correct actions to place the stack of money there. This
 548 example shows LLARVA can, in some cases, work despite visual obstructions, which we believe
 549 is in part attributable to the introduction of the visual traces. We hypothesize that understanding
 550 and predicting the visual trace provides the model with information about a successful end-effector
 551 trajectory in the presence of such occlusions.

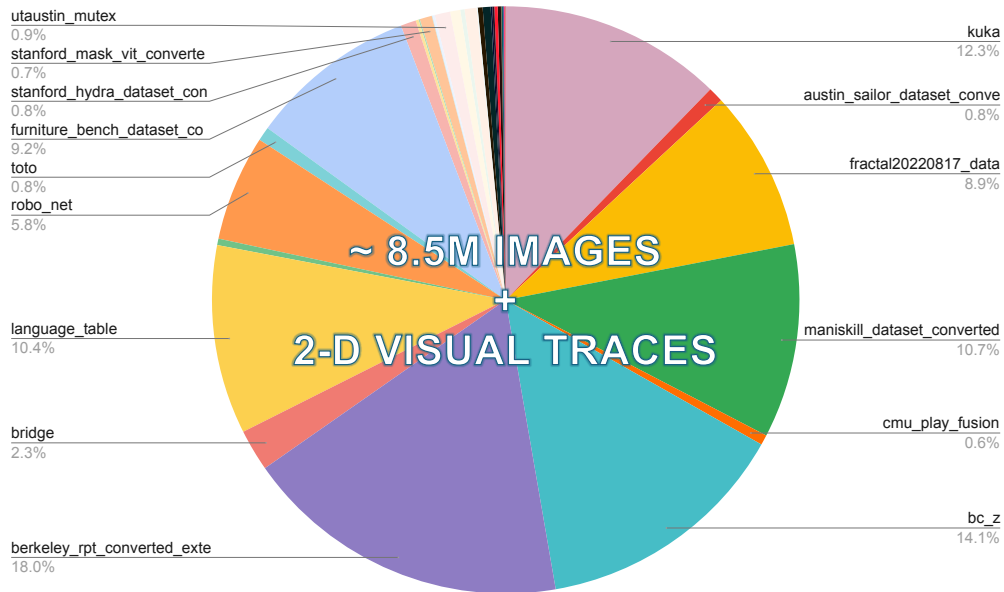


Figure 5: **Data distributions.** We do vision-action instruction pre-training for LLARVA on a dataset built up on Open X-Embodiment [10], including 8.5M image-2-D visual trace pairs.

552 B Additional Dataset details

553 Here, we provide more information about our constructed dataset.

554 **Data Distribution.** As mentioned in the paper, we construct the vision-action tuning dataset from
 555 a subset of Open X-Embodiment (OXE) [10]. We excluded OXE subsets with poor image quality,
 556 smaller image resolution, ambiguous action spaces, or those with widely different robot morpholo-
 557 gies, such as Autonomous Mobile Robots (AMRs, which involve locomotion), resulting in 8.5M
 558 image-text pairs, whose distribution is shown in Figure 5 and Table 6. Overall, we ensured that the
 559 resulting dataset contains subsets of [10] that use end-effector control and joint control, in addition
 560 to including both absolute and delta control modes.

OXE Subset	Number of Image + 2-D visual trace pairs
kuka	1044466
austin_sailor_dataset_converted_externally_to_rlds	70758
fractal20220817_data	753647
maniskill_dataset_converted_externally_to_rlds	909568
cmu_play_fusion	47115
bc_z	1198963
berkeley_rpt_converted_externally_to_rlds_new	1533451
bridge	195745
language_table	885876
stanford_kuka_multimodal_dataset_converted_externally_to_rlds	30128
robo_net	496454
toto	65527
furniture_bench_dataset_converted_externally_to_rlds	786692
stanford_hydra_dataset_converted_externally_to_rlds	72160
ucsd_pick_and_place_dataset_converted_externally_to_rlds	13545
kaist_nonprehensile_converted_externally_to_rlds	6512
stanford_mask_vit_converted_externally_to_rlds	57012
utokyo_pr2_opening_fridge_converted_externally_to_rlds	2276
berkeley_fanuc_manipulation	11854
utaustin_mutex	72461
taco_play	47780
berkeley_autolab_ur5	19621
austin_sirius_dataset_converted_externally_to_rlds	56101
columbia_cairlab_pusht_real	5486
stanford_robocook_converted_externally_to_rlds	22894
roboturk	37120
berkeley_cable_routing	7797
nyu_franka_play_dataset_converted_externally_to_rlds	9118
jaco_play	15515
viola	15146
tokyo_u_lsmo_converted_externally_to_rlds	2398
austin_buds_dataset_converted_externally_to_rlds	6771
dlr_sara_pour_converted_externally_to_rlds	2695
utokyo_xarm_pick_and_place_converted_externally_to_rlds	1381
utokyo_pr2_tabletop_manipulation_converted_externally_to_rlds	6545
dlr_edan_shared_control_converted_externally_to_rlds	746
dlr_sara_grid_clamp_converted_externally_to_rlds	1543

Table 6: More statistics about the vision-action instruction tuning dataset.

2D Visual Trace





Subset	Visual Observations	Robot	Control Mode
fractal20220817_data		Franka	End effector control
berkeley_rpt_converted_external_ly_to_rlds		Franka	Joint control
toto		Franka	End effector control
berkeley_autola_b_ur5		UR5	End effector control

Figure 6: A few samples from our constructed vision-action tuning dataset. We visualize some samples of the instruction tuning dataset used in the pre-training stage of LLARVA, with the corresponding robot type and control mode.

The Generation of 2-D Visual Traces. The 2-D visual traces can be seen as a trace of the end-effector location in the image plane across time. To generate these traces, we trained an object detector to locate the end-effector from input 2-D images. We use the Detectron2 [12] implementation of Faster R-CNN [58] to obtain bounding boxes enclosing the end-effector, and then use the center point of the bounding boxes as the end-effector keypoint. The detector was trained using 200 manually annotated images from each OXE subset. Some examples of the resulting detector training set are shown in Figure 6, where the 2-D visual traces are shown in yellow. Note that the traces are a sequence of 2-D coordinates, and Figure 6 is a visualization of these sequences. During training the sequences are predicted in language token space and compared to ground truth.

C Additional Implementation Details

C.1 RLBench Experiments

LLARVA is evaluated on 12 tasks from RLBench. All RLBench tasks include two or more variations of a language instruction describing the goal. For example, there might be three variations of the instruction for the same task: “open the top drawer”, “grip the top handle and pull the top drawer open” and “slide the top drawer open”. For simplicity, we use the first instruction variant for training. Below, we describe the RLBench tasks we use for simulator evaluation, along with any modifications we made to the tasks. The intention behind the modifications is to increase the variations of the tasks, such as adding distractor objects with different colors. This exercises the model’s language grounding abilities. All tasks are unmodified unless otherwise noted.

Training Setup. We start with a LLARVA model that has undergone vision-action instruction pre-training on OXE as described in Section 2.3, and perform step 2 (Section 2.3) instruction fine-tuning for four epochs on task-specific downstream data (e.g., picking, stacking, destacking) using eight A100 GPUs. Step 2 instruction tuning is done using 800 demonstrations for each RLBench task. The domain gap between step 1 and step 2 is large as we change from almost entirely real data to simulation while at the same time changing robots and tasks. We note that while other works train on a smaller amount of data, they use roughly the same order of magnitude of data as LLARVA, and exploit the power of 3-D representations. For example, PerAct [7] uses 100 examples per task but exploits voxel-based 3-D representations, which are rare and difficult to obtain. Our approach has the advantage of being able to leverage 2-D representations, which may require additional data but with roughly the same order of magnitude as methods that utilize 3-D.

591 **Open Drawer.** The task is to open one of three drawers. The success metric is a full extension of
592 the prismatic joint of the target drawer.

593 **Meat off Grill.** The task is to take either a piece of chicken or steak off the grill and put it on the
594 side. The success metric is the placement of the specified meat on the side, away from the grill.

595 **Turn Tap.** The task is to turn either the left or right handle of the tap. Left and right are defined
596 according to the orientation of the faucet. The success metric is the joint of the specified handle
597 being at least 90° away from the starting position.

598 **Put Money.** The task is to pick up the stack of money and place it on the specified shelf of a safe.
599 The safe has three shelves: top, middle, and bottom. The success metric is the placement of the
600 stack of money on the specified shelf in the safe.

601 **Push Buttons.** The task is to push the colored buttons in the specified sequence. There are always
602 three buttons present in the scene, whose colors are sampled from 20 options, and the number of
603 buttons to press is between one and three. The success metric is all specified buttons being pressed
604 in the right order.

605 **Sweep Dustpan.** The task is to sweep the dirt particles into the specified dustpan. There are two
606 dustpans, one short and one tall, and both are always present in the scene. The success metric is
607 all five dirt particles being inside the specified dustpan. We modified this task by adding a variation
608 with a different-sized dustpan.

609 **Slide Block.** In this task there is a block and four colored squares in the scene (green, blue, pink,
610 and yellow). The task is to slide the block onto either the green or pink squares. The success metric
611 used is some part of the block being on the specified target square. The original task only had one
612 target square, and we modified it by adding three additional colored squares — one target and two
613 distractors.

614 **Close Jar.** The task is to screw in the lid on the jar with the specified color. There are always two
615 colored jars in the scene, one target jar and one distractor jar. The success metric used is the lid
616 being on top of the specified jar and the robot gripper not grasping any object. We modified this task
617 so that the target jar color is drawn from a list of two possible colors (blue or teal). The color for the
618 distractor jar was still chosen out of 20 options.

619 **Screw Bulb.** There are two bulb holders of different colors, and the task is to pick up a light bulb
620 from the stand specified by color and screw it into the bulb stand. The color of the target holder is
621 sampled from two colors, while the color of the distractor holder is sampled from the original 20
622 color options. The success metric used is the bulb from the specified holder being inside the bulb
623 stand. We modified this task to use two colors for the target holder (yellow and purple) rather than
624 20 as in the original task specification.

625 **Place Wine.** The task is to pick up the wine bottle and place it at the specified location in a wooden
626 rack. The rack has three locations: left, middle, and right. The success metric is the placement of
627 the bottle on the specified location in the rack.

628 **Reach and Drag.** The environment has a cube, a stick, and four possible colored target squares.
629 The task is to pick up the stick and use it to drag the cube to the target square of a specified color.
630 The other three squares are considered distractors. The success metric used is some part of the block
631 being inside the target's area. We modified this task to sample the target color from a list of three
632 colors (maroon, magenta, teal). The colors for distractor squares are still sampled from 20 options.

633 **Stack Blocks .** The scene starts with 8 blocks and a green platform. Four of the blocks are of a target
634 color, and the other four have a distractor color. The task is to stack N blocks of the target color on
635 the green platform. The success metric is N blocks being inside the area of the green platform.

636 **Put Item in Drawer.** There is a block kept on top of a chest of closed drawers. The task is to
637 place the block into the specified drawer among three possible options: top, middle, or bottom. The
638 success metric is the placement of the block inside the specified drawer.

639 **Sort Shape.** The scene has four distractor shapes and one correct shape. The task is to pick up the
640 shape specified in the language instruction and place it in the correct hole in the sorter. The success
641 metric is the correct shape being inside the corresponding hole.

642 **Insert Onto Square Peg.** The scene has a platform with three differently colored pegs, and one
643 square shaped object with a hole in the middle. The three colors are sampled from 20 color instances.
644 The task is to pick up the square and put it on the peg specified in the language instruction, with the
645 success metric being the placement of the square fully on the peg.

646 **Stack Cups.** The scene has three cups with colors sampled from 20 options. The task is to stack all
647 cups inside the cup specified in the language instruction. The success metric for this task is all other
648 cups being inside the specified cup.

649 **Put Groceries in Cupboard.** The scene always has nine grocery items and one cupboard. The task
650 is to place the item specified in the language instruction inside the cupboard. The success metric
651 used is the placement of the item inside the cupboard.

652 **Place Cups.** The scene always has one cup holder with three spokes and three cups with handles.
653 The task is to place N of the cups on the cup holder ($N \in \{1, 2, 3\}$). The success metric used is the
654 alignment of each cup’s handle with a spoke on the cup task.

655 **Toilet Seat Down.** The scene consists of a toilet which initially has its seat up. The task is to put the
656 toilet seat down. The success metric used is the joint of the toilet seat being at an angle consistent
657 with the seat being fully down.

658 **Close Laptop Lid.** The scene consists of a laptop which initially has its lid open. The task is to
659 close the laptop. The success metric used is the joint of the laptop lid being at an angle such that the
660 screen is fully down.

661 **Put Knife on Chopping Board.** The scene consists of a knife inside a knife holder, and a chopping
662 board. The task is to pick up the knife from the holder, and place it on the chopping board. The
663 success metric used is the knife being on the surface of the chopping board, and the robot gripper
664 not grasping anything.

665 **Put Umbrella in Umbrella Stand.** The scene consists of an umbrella and an umbrella holder. The
666 task is to pick up the umbrella and put it into the stand. The success metric used is the umbrella
667 being inside the stand, and the robot gripper not grasping anything.

668 **Move Hanger.** The scene consists of a clothes hanger and two racks. The task is to move the hanger
669 from its current rack to the other rack. The success metric used is the hanger being placed on the
670 other rack.

671 C.2 Real Robots Experiments

672 **Hardware Setup.** We use a Franka Emika Panda robot with a Franka gripper for real robot data
673 collection and evaluations. A Logitech BRIO 4K camera positioned to the right of the Franka robot
674 provides single-view RGB (without depth data) vision input to our model, as shown in Figure 7.
675 Camera autofocus is disabled, and the data is captured at 640x480 resolution. The model inference
676 is done on a 48GB NVIDIA A6000.

677 **Data Collection.** We use the data collection code and process from <https://github.com/Max-Fu/franka-scripted>
678 to collect data for picking, stacking, and destacking tasks. The script generates
679 data for an arbitrary number of episodes. For each episode, the process generates x-y positions on
680 the table plane using a uniform random distribution for each axis. The script directs the robot to
681 place the cube at each location and then collects the camera and joint information as the robot is
682 directed to pick, stack, or destack the cubes. Vision is not used during this process as the cube
683 locations are all generated and therefore known.

684 **Training and Execution.** For the Franka Emika Panda robot experiments, we start with our
685 LLARVA model that has undergone vision-action instruction pre-training on OXE as described

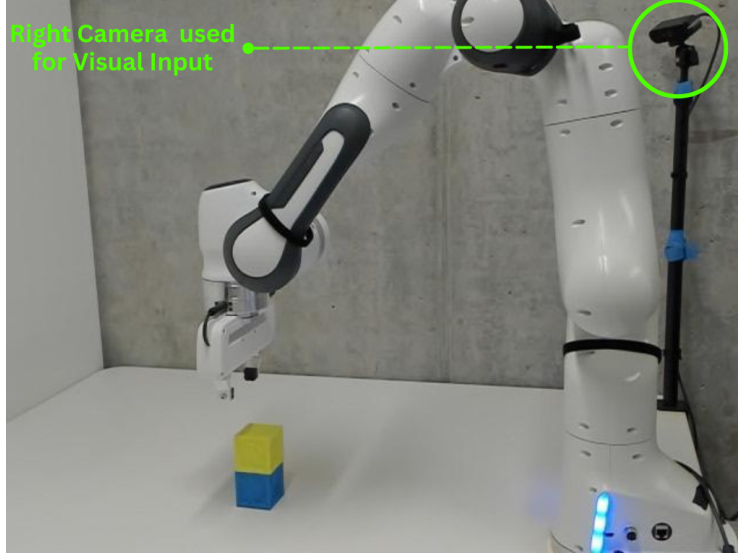


Figure 7: The real robot setup with Franka Emika Panda used for evaluating LLARVA.

in Section 2.3, and perform step 2 (Section 2.3) instruction fine-tuning for four epochs on 1920 episodes of task-specific downstream data (e.g., picking, stacking, destacking) using 8 A100 GPUs. This is similar to other baselines, such as RPT [17], that uses an equal amount of in-domain episodes (1920) for pre-training, with an additional 120-240 episodes used for fine-tuning depending on the task. Additionally, [17] uses three camera views for each episode, while LLARVA uses only one. Nevertheless, it can be observed that LLARVA demonstrates superior performance on all three tasks tested despite using comparable or even fewer episodes. Finally, each real robot evaluation consists of 16 repeated pick, stack, or destack operations at a random x-y location on the table plane for each repetition. We report the success rate of the 16 operations.

D Licenses and Privacy

The license, PII, and consent details of each dataset are in the respective papers. In addition, we wish to emphasize that the datasets we use do not contain any harmful or offensive content, as many other papers in the field also use them. Thus, we do not anticipate a specific negative impact, but, as with any machine learning method, we recommend exercising caution.

References

- [1] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.
- [2] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [3] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *NeurIPS*, 2023.
- [4] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning, 2023.
- [5] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. M. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. C. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. García, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Díaz, O. Firat, M. Catasta, J. Wei, K. S. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2022. URL <https://api.semanticscholar.org/CorpusID:247951931>.
- [6] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. H. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. R. Florence. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, 2023. URL <https://api.semanticscholar.org/CorpusID:257364842>.
- [7] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.
- [8] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.
- [9] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, K. Choromanski, T. Ding, D. Driess, K. A. Dubey, C. Finn, P. R. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. C. Julian, D. Kalashnikov, Y. Kuang, I. Leal, S. Levine, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. S. Ryoo, G. Salazar, P. R. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. H. Vuong, A. Wahid, S. Welker, P. Wohlhart, T. Xiao, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *ArXiv*, abs/2307.15818, 2023. URL <https://api.semanticscholar.org/CorpusID:260293142>.
- [10] O. X.-E. Collaboration, A. O’Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang,

- H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Thompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Bajjal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. <https://arxiv.org/abs/2310.08864>, 2023.
- [11] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [12] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [14] 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.
- [15] 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. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.
- [16] S. James, K. Wada, T. Laidlow, and A. 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.
- [17] I. Radosavovic, B. Shi, L. Fu, K. Goldberg, T. Darrell, and J. Malik. Robot learning with sensorimotor pre-training. In *Conference on Robot Learning*, pages 683–693. PMLR, 2023.
- [18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and

- 378 D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Had-
 379 sell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*,
 380 volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL [https://proceedings.](https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf)
 381 [neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf](https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf).
- 382 [20] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch,
 383 K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Saman-
 384 gooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh,
 385 M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a vi-
 386 sual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022. URL <https://api.semanticscholar.org/CorpusID:248476411>.
 387
- 388 [21] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière,
 389 N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama:
 390 Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. URL <https://api.semanticscholar.org/CorpusID:257219404>.
 391
- 392 [22] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL [https://api.](https://api.semanticscholar.org/CorpusID:257532815)
 393 [semanticscholar.org/CorpusID:257532815](https://api.semanticscholar.org/CorpusID:257532815).
- 394 [23] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified
 395 vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.
- 396 [24] J. Li, D. Li, S. Savarese, and S. Hoi. BLIP-2: bootstrapping language-image pre-training with
 397 frozen image encoders and large language models. In *ICML*, 2023.
- 398 [25] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information*
 399 *processing systems*, 36, 2024.
- 400 [26] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language
 401 understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.
- 402 [27] H. Zhang, X. Li, and L. Bing. Video-llama: An instruction-tuned audio-visual language model
 403 for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. URL [https://arxiv.](https://arxiv.org/abs/2306.02858)
 404 [org/abs/2306.02858](https://arxiv.org/abs/2306.02858).
- 405 [28] Q. Ye, H. Xu, J. Ye, M. Yan, A. Hu, H. Liu, Q. Qian, J. Zhang, F. Huang, and J. Zhou.
 406 mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration.
 407 *ArXiv*, abs/2311.04257, 2023. URL [https://api.semanticscholar.org/CorpusID:](https://api.semanticscholar.org/CorpusID:265050943)
 408 [265050943](https://api.semanticscholar.org/CorpusID:265050943).
- 409 [29] P. Zhang, X. Wang, Y. Cao, C. Xu, L. Ouyang, Z. Zhao, S. Ding, S. Zhang, H. Duan,
 410 H. Yan, X. Zhang, W. Li, J. Li, K. Chen, C. He, X. Zhang, Y. Qiao, D. Lin, and J. Wang.
 411 Internlm-xcomposer: A vision-language large model for advanced text-image comprehension
 412 and composition. *ArXiv*, abs/2309.15112, 2023. URL [https://api.semanticscholar.](https://api.semanticscholar.org/CorpusID:262824937)
 413 [org/CorpusID:262824937](https://api.semanticscholar.org/CorpusID:262824937).
- 414 [30] B. Li, Y. Zhang, L. Chen, J. Wang, F. Pu, J. Yang, C. Li, and Z. Liu. Mimic-it: Multi-modal
 415 in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023.
- 416 [31] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani,
 417 S. Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning*
 418 *Research*, 25(70):1–53, 2024.
- 419 [32] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le.
 420 Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.
- 421 [33] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong. Unleash-
 422 ing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint*
 423 *arXiv:2312.13139*, 2023.

- [34] R. Herzig, A. Mendelson, L. Karlinsky, A. Arbelle, R. Feris, T. Darrell, and A. Globerson. Incorporating structured representations into pretrained vision & language models using scene graphs. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.
- [35] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.
- [36] X. Li, X. Yin, C. Li, X. Hu, P. Zhang, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020.
- [37] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. pages 5099–5110. Association for Computational Linguistics, 2019.
- [38] F. Yu, J. Tang, W. Yin, Y. Sun, H. Tian, H. Wu, and H. Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3208–3216, May 2021. doi:10.1609/aaai.v35i4.16431. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16431>.
- [39] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.
- [40] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019. URL <https://api.semanticscholar.org/CorpusID:152282269>.
- [41] C. Shang, A. You, S. Subramanian, T. Darrell, and R. Herzig. Traveler: A multi-lmm agent framework for video question-answering. *ArXiv*, abs/2404.01476, 2024. URL <https://api.semanticscholar.org/CorpusID:268857079>.
- [42] R. Herzig, E. Levi, H. Xu, H. Gao, E. Brosh, X. Wang, A. Globerson, and T. Darrell. Spatio-temporal action graph networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [43] A. Bar, R. Herzig, X. Wang, A. Rohrbach, G. Chechik, T. Darrell, and A. Globerson. Compositional video synthesis with action graphs. In *ICML*, 2021.
- [44] R. Herzig, E. Ben-Avraham, K. Mangalam, A. Bar, G. Chechik, A. Rohrbach, T. Darrell, and A. Globerson. Object-region video transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [45] X. Wang and A. Gupta. Videos as space-time region graphs. In *ECCV*, 2018.
- [46] E. B. Avraham, R. Herzig, K. Mangalam, A. Bar, A. Rohrbach, L. Karlinsky, T. Darrell, and A. Globerson. Bringing image scene structure to video via frame-clip consistency of object tokens. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022.
- [47] J. Materzynska, T. Xiao, R. Herzig, H. Xu, X. Wang, and T. Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [48] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. In *ECCV*, pages 105–121, 2018.
- [49] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

- 468 [50] A. Jerbi, R. Herzig, J. Berant, G. Chechik, and A. Globerson. Learning object detection from
469 captions via textual scene attributes. *ArXiv*, abs/2009.14558, 2020.
- 470 [51] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image
471 retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and*
472 *pattern recognition*, pages 3668–3678, 2015.
- 473 [52] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene Graph Generation by Iterative Message
474 Passing. In *CVPR*, pages 3097–3106, 2017.
- 475 [53] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping images to scene
476 graphs with permutation-invariant structured prediction. In *Advances in Neural Information*
477 *Processing Systems (NIPS)*, 2018.
- 478 [54] R. Herzig, A. Bar, H. Xu, G. Chechik, T. Darrell, and A. Globerson. Learning canonical
479 representations for scene graph to image generation. In *European Conference on Computer*
480 *Vision*, 2020.
- 481 [55] M. Raboh, R. Herzig, G. Chechik, J. Berant, and A. Globerson. Differentiable scene graphs.
482 In *WACV*, 2020.
- 483 [56] J. Wen, Y. Zhu, M. Zhu, J. Li, Z. Xu, Z. Che, C. Shen, Y. Peng, D. Liu, F. Feng, and J. Tang.
484 Object-centric instruction augmentation for robotic manipulation. *ArXiv*, abs/2401.02814,
485 2024. URL <https://api.semanticscholar.org/CorpusID:266818502>.
- 486 [57] H. Liu, W. Yan, M. Zaharia, and P. Abbeel. World model on million-length video and language
487 with blockwise ringattention. *arXiv preprint arXiv:2402.08268*, 2024.
- 488 [58] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer*
489 *vision*, pages 1440–1448, 2015.