InternData-A1: Pioneering High-Fidelity Synthetic Data for Pre-training Generalist Policy

Supplementary Material

8. Detailed Data Statistics

We report the complete dataset statistics in Tab. 6. In total, the dataset contains **4** embodiments, **70** tasks, **637,498** trajectories, **401,430,981** frames, and **7,433.91** hours of interaction data. As outlined above, the dataset is organized into four categories: **Articulation**, **Long-horizon**, **Base**, and **Pick and Place**. These categories comprise **74,415**, **138,782**, **229,168**, and **195,133** trajectories, accounting for **11.67%**, **21.77%**, **35.95%**, and **30.61%** of the dataset, respectively. For each task, we report the exact number of trajectories contributed by each embodiment. See Tab. 6 for detailed per-task and per-embodiment statistics.

9. Detailed Data Synthesis

We present an example task config below. Following the exact task config, we elaborate on each part in our data synthesis in detail.

9.1. Environment Construction

As shown in the configuration, we set the room environment using `defaults/arenas@arena` and select a dining-room layout in the outer script. We load the Agilex Split Aloha robot—one of our four embodiments—and specify its motion planner via the `robot_file`. We then retrieve two task-relevant assets, the plate and plate shelf, from our asset library, where each asset is automatically annotated with gravity parameters, collision properties, and grasp poses. After obtaining the two objects, we assign their initial translations and orientations. Objects belonging to the same category share a unified canonical pose definition.

9.2. Skill Composition

As shown in the configuration—particularly in the `skills/split_aloha` section—we construct tasks by composing skills either sequentially or in parallel. Users can simply copy and paste different skill blocks to assemble a task. For example, a complete task may be formed by chaining together `pick`, `goto_pose`, `pick`, `gripper_action` (close or open), `home`, and `place`. The framework supports both sequential execution and parallel execution (e.g., one gripper opens while the other closes), enabling users to specify diverse task requirements. Users may also define task-level constraints; for instance, in a placement operation, we enforce `align_pick_obj_axis` and `align_place_obj_axis` to be parallel to ensure accurate insertion. Similarly, `x_ratio_range` and `y_ratio_range` can be used to specify the target insertion layer. All script-level policies have undergone substantial refinement. For manual tuning, users may configure grasp-pose filtering rules (`filter_x_dir`, `filter_y_dir`, `filter_z_dir`) and adjust parameters such as `post_grasp_offset_min`, `post_grasp_offset_max`, `pre_place_z_offset`, and `place_z_offset` to ensure stable grasping and placement while avoiding unsafe motions.

9.3. Domain Randomization

For visual domain randomization, we provide options in `env_map`, allowing light intensity and rotation to be perturbed within predefined ranges. We also support camera extrinsic randomization, where camera poses are perturbed by up to 5° in rotation and 5 cm in translation. Room scenes can be randomized by sampling from the specified room types. For objects, replacements can be sampled from assets within the same category. At the trajectory level, we define a spatial region in which target objects and robots are initialized with randomized poses for each episode. Additionally, the `robots` configuration allows specifying the mean and standard deviation of the home configuration, enabling diverse initial joint states. Within the skill definitions, we further introduce loose filtering ranges for grasp and placement poses. All poses that satisfy these constraints are retained, and a final pose is selected randomly. Together, these mechanisms significantly enhance trajectory diversity within each task.

9.4. Generation & Storage

After the previous steps, our data engine begins planning the trajectory based on the waypoints produced by the skill module. We design an optimization strategy that improves generation efficiency by 2.5–3×, particularly for long-horizon tasks involving multiple skills. Because rendering is significantly more time-consuming than physics-only planning, we adopt a decoupled procedure. We first run pure physics simulation without rendering; if any intermediate step fails, we immediately terminate and restart trajectory planning. Only after a trajectory is successfully planned do we replay the entire sequence with rendering enabled, ensuring that all object states are identical to those in the successful physics-only episode. This approach eliminates unnecessary rendering for failed attempts and greatly

reduces overhead in tasks with high failure rates, maximizing overall data-generation efficiency.

10. Framework Optimization

In traditional synthetic data generation pipelines, trajectory planning and visual rendering are integrated into a single stage. While this architecture is suitable for rapid development and iteration, it exhibits substantial efficiency bottlenecks when scaled to large-scale data generation. The root causes can be summarized as follows:

1. **Declining planning success rate with increasing task complexity.** As task complexity grows, the success rate of trajectory planning decreases significantly. Failed trajectories do not require subsequent visual rendering, yet the single-stage architecture incurs redundant rendering overhead, resulting in unnecessary computational waste.
2. **Mismatch in computational characteristics.** Trajectory planning is fundamentally CPU-bound and executed serially, whereas visual rendering relies on GPU-based parallel computation. Executing these heterogeneous workloads in a serial manner leads to poor overall hardware utilization.

To mitigate these bottlenecks, we introduce a multi-level system optimization at the framework level. Our design includes:

1. **Stage decoupling with a pipelined architecture:** Trajectory planning and visual rendering are decoupled into two independent stages, with a pipelined execution mechanism established between the Planner and Renderer.
2. **Dynamic resource scheduling:** To address heterogeneous time-cost ratios across different tasks, we incorporate parallel batch processing strategies within both the Planner and Renderer, together with a dynamic scheduling algorithm to maximize resource utilization.
3. **Rendering efficiency optimization:** We introduce a stacked rendering (Stack Render) technique to further increase rendering throughput.
4. **Cluster stability mechanisms:** To handle stability issues and load imbalance in large-scale cluster deployments, we design a *Balancer* module for load distribution and a *Supervisor* module for monitoring and control, jointly improving cluster utilization and system robustness.

With these optimizations, our pipeline achieves a **2–3**$\times$ end-to-end performance improvement over the baseline. It further supports long-duration stable operation and efficient large-scale synthetic data generation, substantially improving productivity in synthetic data production.

11. Policy Training Details

During real-world training, we pretrain a new π_0 model, initialized with Paligemma weights and a scratched action expert, on InternData-A1 for 680k iterations using 64 GPUs (closely matches the 700k iteration steps of the official π_0 checkpoint trained on the π-dataset). For 10 sim-to-real experiments and 9 real-world tasks, we start from the 680k π_0(InternData-A1) checkpoint and perform post-training for 30k iterations on 8 GPUs for regular tasks and sim-to-real tasks. For dexterous tasks, we trained for 100k iterations. Key training hyperparameters are summarized in Tab. 5.

Table 5. **Training hyperparameters.**

Hyperparameters	Pre-training	Fine-tuning
Batch Size(Total)	512	128
Learning Rate	5e-5	2.5e-5
Learning Rate Schedule	Constant	Cosine Decay
Training Steps	680k	30k(Regular)/100k(Dexterous)

12. Real-World Experiments

In this section, we describe the real-world and sim-to-real tasks in detail. For both experiments, we post-train a JAX-version π_0 (InternData-A1) model for 30k iterations and use the 30k checkpoint for evaluation. For each task, we define 15 evaluation settings, and to reduce stochasticity, we run two trials per setting. In total, each task is evaluated with 30 rollouts, and we report the average success rate.

12.1. Real Task Description

Place Markpen. The Genie-1 robot is required to pick a black marker with its right arm and place it into a pen holder. This task evaluates the model's fundamental pick-and-place capabilities. A trial is considered successful only if the marker is placed precisely and fully inside the pen holder.

Pass Bottle. The Genie-1 robot is required to pick up a black tea bottle, lift it upright, and hand it to a nearby person with the right arm. The robot may release its gripper only when the human presents their hand. This task evaluates the model's fundamental abilities in picking, lifting, and human–robot interaction. A trial is considered successful only if the bottle is successfully transferred to the human and the robot releases its gripper accordingly.

Heat Sandwich. The ARX Lift-2 robot must open the oven with its left arm, pick up the plate containing the sandwich, place it into the oven using its right arm, and then close the oven with its left arm. This task assesses the model's ability to operate articulated objects. A trial is considered successful only if the plate is correctly inserted into the

oven and the oven door is fully closed.

Sort Rubbish. The ARX Lift-2 robot must use its right arm to place all recyclable waste into the right bin and all non-recyclable waste into the left bin. This task evaluates the model's ability to handle diverse object layouts and perform repetitive pick-and-place operations. A trial is considered successful only if all waste items are fully and correctly sorted.

Sweep Trash. The ARX Lift-2 robot must grasp the dustpan with its right arm and the broom with its left arm. It then uses the broom to sweep all crumpled paper balls into the dustpan. Afterwards, the robot empties the dustpan into the left rubbish bin. Finally, it releases both grippers and returns to the home position. A trial is considered successful only if every step is finished successfully.

Sort Parts. The ARX Lift-2 robot must sort four types of small industrial components into four designated containers. These components include small nuts, assembly parts, and small screws. Each arm is responsible for sorting two categories. A trial is considered successful only if all components are placed into their correct containers.

Unscrew Cap The ARX AC One robot must grasp the tea bottle with its left arm and move it to the designated middle zone. It then uses its right arm to approach the bottle cap and unscrew it. A trial is considered successful only if the cap is fully removed.

Fold Cloths. The ARX AC One robot must fold the cloth into its designated final shape with both hands. A trial is considered successful only if the cloth is folded correctly.

Zip Bag. The ARX AC One robot must use its left arm to open the bag, place all designated objects inside, and then zip it closed. A trial is considered successful only if the bag is fully and correctly zipped.

12.2. Sim-to-real Task Description

Flip Package. A package is placed on the conveyor and moves toward the robot. The ARX Lift-2 robot must grasp the package with its right arm, flip it over, and place it back onto the conveyor. It must then grasp the package with its left arm and scan the QR code using the robot-mounted camera. A trial is considered successful only if all steps are completed correctly.

Instructional Pick. Eight types of objects are placed on the table. A trial is considered successful only if the robot correctly picks the target object specified by the command.

Sort Rubbish. This is the same task as described before.

Wipe Stain. The ARX Lift-2 robot uses its left arm to pick up the towel and wipe stains located in one or two clusters. A trial is considered successful only if all stains are completely removed.

Sandwich. The ARX Lift-2 robot uses its right arm to grasp a piece of bread and place it on the plate. It then uses its left arm to grasp a piece of beef and place it on the bread, followed by using the right arm again to place another piece of bread on top of the beef. A trial is considered successful only if the sandwich is assembled correctly and neatly.

Box. The ARX Lift-2 robot sequentially closes the box lids with its right and left arms. A trial is considered successful upon complete closure.

Microwave. The ARX Lift-2 robot uses the right arm to close the microwave lid. A trial is considered successful upon complete closure.

Pack. The ARX Lift-2 robot manipulates objects and places them into a box using its right and left arms. A trial is considered successful only after all objects have been placed inside.

Sweep. This is the same task as described before.

Handover. The ARX Lift-2 robot uses its left arm to pick up a long-shaped object and hands it over to the right arm, which then places it into the box. A trial is considered successful only upon the object's transfer into the box.

Table 6. **Task Statistics Across Robots.**

Task Name	Franka	ARX Lift-2	Agilex Split Aloha	Genie-1	Sum
Articulation Tasks (11.67%)					**74,415**
Close The Electric Cooker	1776				
Close The Laptop	578				
Close The Pot	2595				
Close The Trashcan	2996				
Close The Microwave	2496	6831	4367		
Open The Laptop	416				
Open The Pot	3250				
Open The Trashcan	3507				
Open The Microwave	2139	5148	4817		
Pull The Storage Furniture		4368	4776		
Push The Storage Furniture		4548	4775		
Rotate The Hearth		6046	4653		
Open Microwave From Scratch			1501		
Heat Food In Microwave		108			
Close The Package		2724			
Long-horizon Tasks (21.77%)					**138,782**
Clean Dirt With Brown Cloth		3000	3000		
Clean Dirt With Sponge		3000	3000		
Clean Dirt With White Cloth		3000	3000		
Collect Three Glues		2000	2000		
Gather Three Teaboxes		110	2000		
Handover Objects		7863			
Pack In Objects		4052	2345		
Pack Out Objects			2070		
Sort The Rubbish		4579	8860		
Stack Multiple Objects		4867	4664		
Sweep The Trash	2344	528	1626		
Put Trash In Trashcan	1480				
Collaborate Assemble Beef Sandwich		4854			
Stack A Beef Sandwich		4271	670		
Store Objects In Drawer		2822			
Collaborate Assemble Ham Sandwich		3168			
Continues Pick And Place	20036	15000	18573		
Base Tasks (35.95%)					**229,168**
Track The Target	2959	2954	3000		
Organize Three Brushes	5064	2000	2000		
Organize Alarm Clocks	5111	2000	2000		
Organize Colorful Cups	5097	2000	2000		
Organize Three Glues	5120				
Collect Shoes	5114	2000	2000		
Organize Three Teaboxes	5119				
Sort Table Waste	5117	214	2000		
Store Eggs				4244	
Take Shelf Items To Cart				6040	
Pick Beef Sandwich On Conveyor		6658	6647		

Continued on next page

Task Name	Franka	ARX Lift-2	Agilex Split Aloha	Genie-1	Sum
Pick Ham Sandwich On Conveyor		4092	4220		
Fold Long Shirts		731			
Fold Short Shirts		492			
Fold Towels		500			
Fold Short Pants		750			
Flip Package On Conveyor		4806			
Pick Package On Conveyor		4900			
Hang Cups On Rack		5000	5000		
Insert Flower In Vase		5000	4986		
Insert Markpen In Penholder		5000	5000		
Pour Baijiu		4999	4999		
Pour Redwine		5000	5000		
Pour Water		5000	5000		
Pick The Priced Item	5105	2000	2000		
Select A Drink	5121	2000	2000		
Stack Two Boxes		2270	2429		
Sort Tray On Rack		3851	3444		
Store Toothbrushes		1396			
Arrange The Tableware		650			
Recovery Pick Objects		10969			
Watering Plants		5000	5000		
Scan The QRcode				4000	
Sort Metallic Objects		2500	2500		
Pick and Place Tasks (30.61%)					**195,133**
Single Arm Pick	24598	38865	39219	21695	
Parallel Pick And Place		15687	18497	10381	
Grasp Functional Part				4833	
Multiple Pick And Place	21358				
Overall Trajectories					**637,498**
Overall Frames					**401,430,981**
Overall Hours					**7433.91**

```yaml
defaults:
  - _self_
  - world
  - logger
  - ../arenas@arena: scene_arena
  - ../cameras@astra: astra
  - ../cameras@realsense_d455_v3: realsense_d455_v3

name: banana_base_task
asset_root: assets
task: BananaBaseTask
task_id: 0

offset: null
render: True

env_map:
  envmap_lib: envmap_lib
  apply_randomization: True
  intensity_range: [4000, 7000]
  rotation_range: [0, 180]

robots:
  -
    name: "split_aloha"
    target_class: SplitAloha
    path: "split_aloha_mid_360/robot_task13.usd"
    camera_mount: "split_aloha_mid_360_with_piper/split_aloha_mid_360_with_piper/fl/camera"
    euler: [0.0, 0.0, 90.0]
    robot_file:
      - curobo/src/curobo/content/configs/robot/piper100_left_arm.yml
      - curobo/src/curobo/content/configs/robot/piper100_right_arm.yml
    left_joint_home: [0.00484993, 0.34198609, -0.14007858, 0.01680429, 0.14391101, -0.00252178]
    right_joint_home: [0.00484993, 0.34198609, -0.14007858, 0.01680429, 0.14391101, -0.00252178]
    left_joint_home_std: [0.12513939, 0.24539099, 0.24468172, 0.23398885, 0.2710117, 0.21726329]
    right_joint_home_std: [0.12513939, 0.24539099, 0.24468172, 0.23398885, 0.2710117, 0.21726329]

objects:
  -
    name: arcode_plate_blue
    path: assets/plate/plate_blue/Aligned_obj.usd
    target_class: RigidObject
    dataset: arcode
    category: plate
    prim_path_child: Aligned
    translation: [0.0, 0.0, 0.0]
    euler: [90.0, 0.0, 0.0]
    scale: [1.0, 1.0, 1.0]
  -
    name: arcode_plate_shelf
    path: assets/plate_shelf/shelf_0/Aligned_obj.usd
    target_class: RigidObject
    dataset: arcode
    category: plate
    prim_path_child: Aligned
    translation: [0.0, 0.0, 0.0]
    euler: [90.0, 0.0, 0.0]
    scale: [1.0, 1.0, 1.0]

regions:
  -
    object: ${robots.0.name}
    target: table
    random_type: A_on_B_region_sampler
    random_config:
      pos_range: [
```

```
        [0.0, -0.86, -0.765],
        [0.0, -0.86, -0.765]
      ]
      yaw_rotation: [0.0, 0.0]
  -
    object: arcode_plate_blue
    target: table
    random_type: A_on_B_region_sampler
    random_config:
      pos_range: [
        [0.125, -0.20, 0.005],
        [0.25, -0.10, 0.005]
      ]
      yaw_rotation: [0, 0]
  -
    object: arcode_plate_shelf
    target: table
    random_type: A_on_B_region_sampler
    random_config:
      pos_range: [
        [-0.25, -0.20, 0.005],
        [-0.15, -0.10, 0.005]
      ]
      yaw_rotation: [0, 0]

cameras:
  -
    name: ${robots.0.name}_hand_left
    translation: [0.0, 0.08, 0.05]
    orientation: [0.0, 0.0, 0.965, 0.259]
    camera_axes: usd
    params: ${astra}
    parent: "${robots.0.name}/split_aloha_mid_360_with_piper/fl/link6"
    apply_randomization: False

  -
    name: ${robots.0.name}_hand_right
    translation: [0.0, 0.08, 0.04]
    orientation: [0.0, 0.0, 0.972, 0.233]
    camera_axes: usd
    params: ${astra}
    parent: "${robots.0.name}/split_aloha_mid_360_with_piper/fr/link6"
    apply_randomization: False

  -
    name: ${robots.0.name}_head
    translation: [0.0, -0.00818, 0.1]
    orientation: [0.658, 0.259, -0.282, -0.648]
    camera_axes: usd
    params: ${realsense_d455_v3}
    parent: "${robots.0.name}/split_aloha_mid_360_with_piper/top_camera_link"
    apply_randomization: False

data:
  save_root_path: "InternData-A1/sim/raw_data"
  task_dir: "Sort Tray On Rack"
  language_instruction: "Pick the plate, make the handover and place it on the water cooling holder"
  detailed_language_instruction: "Pick the plate with the right arm, make the handover to the left
      arm, and then place it on the water cooling holder."
  collect_info: ""
  version: "v3.0, head camerea 1280x720, wrist 640x480, y 45 degrees"
  update: True
  max_episode_length: 4000

skills:
  -
    split_aloha:
```

```
    -
      right:
        -
          name: pick
          objects: [arcode_plate_blue]
          filter_x_dir: ["upward", 90, 45]
          filter_y_dir: ["forward", 40]
          filter_z_dir: ["downward", 110, 140]
          t_eps: 0.01
          o_eps: 1
          close_wait_steps: 10
          post_grasp_offset_min: 0.1
          post_grasp_offset_max: 0.1
          direction_to_obj: right

        -
          name: goto__pose
          frame: robot
          gripper_action: close_gripper
          translation: [0.3, 0.13, 0.15]
          quaternion: [-0.15, -0.37, -0.84, -0.36]

    -
      left:
        -
          name: pick
          objects: [arcode_plate_blue]
          filter_y_dir: ["upward", 40]
          filter_z_dir: ["forward", 90, 45]
          close_wait_steps: 10
          t_eps: 0.01
          o_eps: 1
          post_grasp_offset_min: 0.0
          post_grasp_offset_max: 0.0
          direction_to_obj: left

    -
      left:
        - name: gripper__action
          action_type: close
      right:
        - name: gripper__action
          action_type: open

    -
      right:
        - name: home

    -
      left:
        -
          name: place
          place_direction: vertical
          objects: [arcode_plate_blue, arcode_plate_shelf]
          filter_y_dir: ["upward", 60, 0]
          filter_z_dir: ["forward", 90, 30]
          position_constraint: object
          x_ratio_range: [0.5, 0.5]
          y_ratio_range: [0.8, 0.8]
          align_pick_obj_axis: [0, 1, 0]
          align_place_obj_axis: [0, 0, 1]
          align_obj_tol: 10
          pre_place_z_offset: 0.15
          place_z_offset: 0.01
```

Listing 1. **A Task Config Example on *Sort Tray On Rack*.**

References

[1] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. In ArXiv, 2024. 6

[2] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. In ArXiv, 2025. 2, 3

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In RSS, 2024. 2, 3

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In ArXiv, 2022. 3

[5] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In IROS, 2025. 2, 3, 7

[6] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. In RSS, 2025. 3

[7] Chilam Cheang, Sijin Chen, Zhongren Cui, Yingdong Hu, Liqun Huang, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Xiao Ma, et al. Gr-3 technical report. In ArXiv, 2025. 3

[8] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. In ArXiv, 2024. 3

[9] Anka He Chen, Ziheng Liu, Yin Yang, and Cem Yuksel. Vertex block descent. In TOG, 2024. 5

[10] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In ArXiv, 2025. 2, 3, 4, 6

[11] Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. In ArXiv, 2025. 2, 3, 8

[12] Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, et al. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. In ArXiv, 2025. 8

[13] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In CVPR, 2023. 4

[14] Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Wenhao Zhang, et al. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. In ArXiv, 2024. 2, 3, 8

[15] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. In TRO, 2023. 4, 5

[16] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In CVPR, 2023. 4

[17] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. In ArXiv, 2023. 2

[18] Pu Hua, Minghuan Liu, Annabella Macaluso, Yunfeng Lin, Weinan Zhang, Huazhe Xu, and Lirui Wang. Gensim2: Scaling robot data generation with multi-modal and reasoning llms. In ArXiv, 2024. 4

[19] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In CORL, 2025. 2, 3

[20] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. In RAL, 2020. 2, 4

[21] Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and g0 dual-system vla model. In ArXiv, 2025. 3

[22] Zhao Jin, Zhengping Che, Zhen Zhao, Kun Wu, Yuheng Zhang, Yinuo Zhao, Zehui Liu, Qiang Zhang, Xiaozhu Ju, Jing Tian, et al. Artvip: Articulated digital assets of visual realism, modular interaction, and physical fidelity for robot learning. In ArXiv, 2025. 4

[23] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In RSS, 2024. 3

[24] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In CORL, 2024. 3

[25] Hao Li, Shuai Yang, Yilun Chen, Yang Tian, Xiaoda Yang, Xinyi Chen, Hanqing Wang, Tai Wang, Feng Zhao, Dahua Lin, et al. Cronusvla: Transferring latent motion across time for multi-frame prediction in manipulation. In AAAI, 2025. 2

[26] Xinyu Lian, Zichao Yu, Ruiming Liang, Yitong Wang, Li Ray Luo, Kaixu Chen, Yuanzhen Zhou, Qihong Tang, Xudong Xu, Zhaoyang Lyu, et al. Infinite mobility: Scalable

high-fidelity synthesis of articulated objects via procedural generation. In ArXiv, 2025. 4

[27] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In NIPS, 2023. 2

[28] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In ArXiv, 2023. 2

[29] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. In RAL, 2022. 2, 3, 4

[30] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In CVPR, 2024. 2, 3, 4, 7

[31] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In ICRA, 2024. 2, 3, 7

[32] Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, et al. Eo-1: Interleaved vision-text-action pretraining for general robot control. In ArXiv, 2025. 3

[33] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, Nathan Ratliff, and Dieter Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023. 4, 5

[34] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In ICLR, 2025. 3

[35] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In CORL, 2023. 3

[36] Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, et al. Grutopia: Dream general robots in a city at scale. In ArXiv, 2024. 4

[37] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In RSS, 2024. 3

[38] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In CVPR, 2023. 4

[39] Shuai Yang, Hao Li, Yilun Chen, Bin Wang, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Instructvla: Vision-language-action instruction tuning from understanding to manipulation. In ArXiv, 2025. 3

[40] Yuyin Yang, Zetao Cai, Yang Tian, Jia Zeng, and Jiangmiao Pang. Gripper keypose and object pointflow as interfaces for bimanual robotic manipulation. In RSS, 2025. 3

[41] Andy Zhai, Brae Liu, Bruno Fang, Chalse Cai, Ellie Ma, Ethan Yin, Hao Wang, Hugo Zhou, James Wang, Lights Shi, et al. Igniting vlms toward the embodied space. In ArXiv, 2025. 3

[42] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In ArXiv, 2025. 3

[43] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In CORL, 2023. 3