NORTHWESTERN
UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2026-28

June, 2026

3D Asset Annotation For Scalable Robot Learning

Shuyang Yu

Abstract

Simulation enables scalable robot data collection, but raw 3D assets provide only geometry, lacking the semantic, interactive, and physical knowledge needed to specify where and how robots should act. In this work, we present ScalableAnno, a general automatic annotation framework that converts passive 3D assets into manipulation-ready assets with structured, diverse, and executable manipulation labels. ScalableAnno is built around two complementary pipelines. First, a unified visual-language annotation pipeline using vision-language reasoning to infer object semantics, interaction constraints, and 3D-grounded cues, providing human-prior guidance for identifying meaningful interaction regions. Second, a fully automatic and massively parallel physics annotation pipeline grounds these priors in each asset's geometry and physical constraints through candidate generation, geometry optimization and trajectory generation. This pipeline produces diverse and executable action annotations, including grasp poses, dexterous contacts, articulation waypoints, insertion directions, hanging affordances, and navigation targets. Based on the generated annotations, we further build an asynchronous parallel simulation data-collection system across diverse objects, tasks, and robot embodiments. Experiments demonstrate that ScalableAnno achieves superior annotation efficiency, data-collection efficiency, and task success rates over existing annotation and data-generation pipelines, while also supporting downstream tasks such as affordance detection, robotic VQA, and visual instruction finetuning.

Keywords

Automatic 3D Asset Annotation, Manipulation-Ready Assets, Visual-Language-Action Annotation, Actionable Robot Labels, Robot Manipulation, Simulation-based Robot Learning

NORTHWESTERN UNIVERSITY

3D ASSET ANNOTATION FOR SCALABLE ROBOT LEARNING

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

MASTER OF SCIENCE

Computer Science

By

Shuyang Yu

EVANSTON, ILLINOIS

June 2026

ABSTRACT

Simulation enables scalable robot data collection, but raw 3D assets provide only geometry, lacking the semantic, interactive, and physical knowledge needed to specify where and how robots should act. In this work, we present **ScalableAnno**, a general automatic annotation framework that converts passive 3D assets into manipulation-ready assets with structured, diverse, and executable manipulation labels. ScalableAnno is built around two complementary pipelines. First, a unified visual-language annotation pipeline using vision-language reasoning to infer object semantics, interaction constraints, and 3D-grounded cues, providing human-prior guidance for identifying meaningful interaction regions. Second, a fully automatic and massively parallel physics annotation pipeline grounds these priors in each asset's geometry and physical constraints through candidate generation, geometry optimization and trajectory generation. This pipeline produces diverse and executable action annotations, including grasp poses, dexterous contacts, articulation waypoints, insertion directions, hanging affordances, and navigation targets. Based on the generated annotations, we further build an asynchronous parallel simulation data-collection system across diverse objects, tasks, and robot embodiments. Experiments demonstrate that ScalableAnno achieves superior annotation efficiency, data-collection efficiency, and task success rates over existing annotation and data-generation pipelines,

while also supporting downstream tasks such as affordance detection, robotic VQA, and visual instruction finetuning.

Table of Contents

CHAPTER 1

Introduction

Figure 1.1. Overview of ScalableAnno. The framework converts raw 3D assets into manipulation-ready annotations across four asset families: rigid objects, articulated objects, deformable objects, and room-scale navigation scenes.

Training general-purpose robotic agents requires large-scale and diverse robot data across objects, embodiments, and tasks [1–4]. Collecting such data in the real world is expensive and difficult to scale, while simulation offers a more scalable alternative when supported by suitable 3D assets and tools [5–8]. However, raw 3D assets typically encode only shape and appearance: they tell a simulator how an object looks, but not where to grasp it, which part can move, or where insertion or hanging is feasible. The bottleneck for scalable robot learning is therefore not the number of assets, but the lack of *actionable* annotations that turn passive geometry into manipulation-ready assets.

In this thesis we present **ScalableAnno**, a framework that automatically converts raw 3D assets into manipulation-ready assets through three complementary types of annotations: *language* annotations that describe object semantics and interactions, *visual* annotations that provide 3D-grounded part, keypoint, and affordance cues, and *action* annotations that encode executable manipulation parameters such as grasp poses, articulation waypoints, and navigation targets. The framework combines two pipelines: a vision-language pipeline that injects human interaction priors into each asset, and a physics-grounded action pipeline that turns those priors into executable, validated labels. Built on top of the annotations, we further demonstrate a parallel simulation data-collection system and several annotation-derived applications.

The main contributions of this thesis are:

- A unified visual-language annotation pipeline that converts raw 3D assets into hierarchical language descriptions and 3D-grounded visual annotations (parts, keypoints, affordances).
- A physics-grounded action annotation pipeline that turns these priors into diverse, executable action labels covering grasping, articulation, insertion, hanging, deformable manipulation, and navigation.
- A demonstration that the resulting annotations support scalable simulation data collection and downstream applications such as affordance detection and robot VQA, with experimental evidence on a heterogeneous audited evaluation suite.

The remainder of this thesis is organized as follows. Chapter 2 reviews related work on automatic robot data collection and manipulation annotation. Chapter 3 describes

the ScalableAnno framework. Chapter 4 describes downstream usages of the generated annotations. Chapter 5 reports experimental results, and Chapter 6 concludes.

CHAPTER 2

Related Work

Automatic data collection for robot learning. Automatic robot data collection broadly falls into annotation-based and RL-based methods. Annotation-based pipelines such as InternData-A1 [9], RoboTwin 2.0 [10], and [11] rely on manually curated task templates, hand-designed grasp candidates, or per-asset configuration. RL-based pipelines such as RoboGen [12] and [13–17] avoid manual labels but require reward engineering and substantial compute, and may yield behaviors that drift from human priors. In contrast, ScalableAnno turns raw 3D assets into reusable, skill-consumable annotations and uses them as inputs for downstream data collection, decoupling annotation from any specific data generator.

Automatic annotation for manipulation. Prior automatic annotation methods typically target a single label type such as grasps [16, 18, 19], contacts, affordances, or articulation waypoints [20, 17], and are often restricted to a specific object category or embodiment. Recent work on simulation-side annotation, including GenManip [21], GenieSim 3.0 [22], and MolmoBot [23], broadens this scope but still focuses on tabletop manipulation with limited deformable, dexterous, or room-scale coverage. ScalableAnno unifies skill-consumable language, visual, and action labels across grasping, dexterous and bimanual manipulation, articulation, insertion, hanging, deformable manipulation, and navigation under a single annotation pipeline.

CHAPTER 3

Method

ScalableAnno takes a raw 3D asset—ranging from a single object to a whole room—and produces three types of annotations: language descriptions, 3D-grounded visual annotations, and physics-validated action labels. The framework is built from two complementary pipelines. A *visual-language annotation pipeline* (Section 3.1) extracts human interaction priors and grounds them in each asset's geometry. A *physics-based action annotation pipeline* (Section 3.2) converts those priors into executable, validated action labels covering rigid, articulated, deformable, and room-scale interactions. Because the visual-language pipeline is the main technical focus of this thesis, it is described in detail; the physics pipeline is summarized at a higher level.

3.1. Hierarchical Visual-Language Annotation Pipeline

The goal of the visual-language pipeline is to enrich each asset with human-prior reasoning that is grounded in its geometry. For each asset it produces two types of output: **language annotations**, which describe the asset and its interactions in natural language, and **visual annotations**, which mark interaction-relevant regions on the 3D surface. Annotations are generated at both an asset level and a room level, and linked across levels through shared instance identifiers.

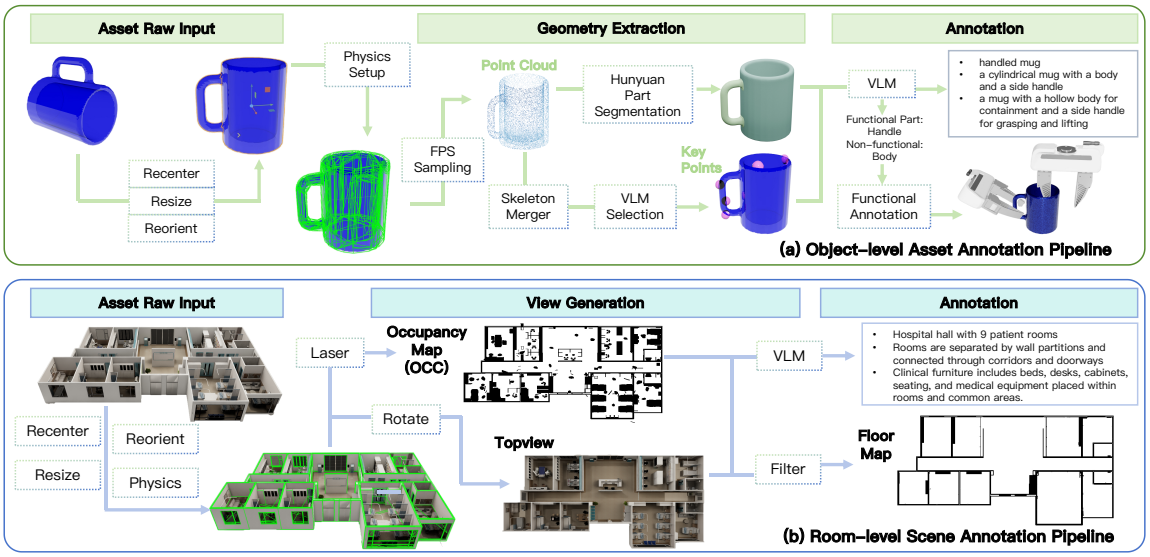

Figure 3.1. Hierarchical visual-language annotation pipeline. For each asset the pipeline produces language descriptions and 3D-grounded visual annotations (parts, keypoints, affordances), and composes them into room-level annotations.

3.1.1. Hierarchical Language Annotation

Asset-level language. For each object asset we feed multi-view RGB renderings to a vision-language model (Qwen3-VL [24]) and prompt it to generate three levels of description: (i) a *semantic phrase* that names the object and its functional category; (ii) a *functional sentence* that summarizes how the object is typically used; and (iii) a *part-aware paragraph* that enumerates manipulable parts and feasible interactions. These hierarchical descriptions provide priors for both visual grounding and action generation: the semantic phrase anchors the asset to a category, the functional sentence specifies plausible high-level interactions, and the part-aware paragraph identifies sub-parts that the visual annotation stage should localize.

Room-level language. For room-scale scenes we additionally render a selection of room views and a labeled top-view occupancy map, and query the same VLM for a three-level room description: a *layout summary*, a *furniture-and-zone description*, and a *dense scene context* paragraph. These descriptions capture spatial relations between objects, navigable regions, and task-relevant interaction areas, and complement the object-level annotations when the asset participates in mobile or scene-level tasks.

3.1.2. Hierarchical Visual Annotation

Asset-level visual annotation. For each object asset we reconstruct a fused RGB-D point cloud from multi-view renderings and produce three types of 3D-grounded annotations: (i) *semantic keypoints*, obtained by farthest-point sampling on the surface and then having the VLM select task-relevant points conditioned on the asset-level language description; (ii) *part masks*, obtained through a native 3D part decomposition using P3-SAM [25] and X-Part [26]; and (iii) *affordance regions*, obtained by projecting language-described interaction regions onto the 3D surface and refining them against the part masks. Together, these provide 3D anchors that subsequent physics-based action generation can ground its labels on.

Room-level visual annotation. For room-scale scenes we construct multi-height occupancy maps from simulated LiDAR and ray-cast observations, and derive floor-plan and wall annotations by filtering out movable objects. These visual annotations provide the geometric context required for navigation, exploration, and scene-level action generation.

Cross-level composition. For each room-scale scene we isolate object instances, annotate each instance independently using the object-centric pipeline above, and then

transform the resulting annotations back to the global frame using scene instance identities. Each room object is therefore associated with both its own semantic, visual, and action-relevant annotations and with the surrounding room-level descriptions, all linked through shared identifiers.

3.2. Physics-based Action Annotation Pipeline

The physics pipeline turns the grounded priors from the visual-language stage into structured, executable action labels. We summarize the design here; details on each skill are provided in our project materials.

Unified action schema. Rather than committing to a single canonical action per object, we organize action annotations as a hierarchical *candidate bank*. Given an asset, its grounded anchors (parts, keypoints, affordance regions, and scene regions) are combined with compatible skill types to produce multiple feasible candidates. Each candidate stores a skill type, a target on the asset, skill-specific parameters such as pose, contact, or direction, and an optional trajectory together with feasibility metadata. This one-to-many schema lets downstream modules choose among diverse, physically feasible, and functionally consistent labels.

Generation, optimization, and validation. For each skill the pipeline runs four stages. (i) *Candidate target generation* localizes skill-compatible regions using the grounded annotations—for example, grasp regions for grasping, handles for articulation, or navigation-mesh points for approach—and samples concrete targets filtered by geometry and embodiment constraints. (ii) *Trajectory generation* produces an initial waypoint sequence from the target, chosen according to the skill's constraint source (object property,

articulated motion, interaction direction, or scene trajectory). (iii) *Trajectory optimization* refines parameters and waypoints under task, contact, collision, kinematic, and smoothness costs [27, 28]. (iv) *Physics validation* executes optimized candidates in parallel simulation, using floating end-effectors for most object-centric skills and full manipulators for embodiment-dependent skills, with perturbations to test robustness. Validated candidates are further expanded by local perturbation and symmetry-aware augmentation to increase diversity while preserving the same functional affordance.

The result is a manipulation-ready asset whose annotation bank contains diverse, validated labels across skills such as grasping, dexterous grasping, bimanual grasping, articulation, insertion, hanging, deformable manipulation, and navigation.

CHAPTER 4

Downstream Tasks

ScalableAnno turns passive 3D assets into reusable visual-language-action annotations that can drive several downstream uses for robot learning. This chapter focuses on the largest of these uses—large-scale simulation data collection—and briefly describes additional applications that the same annotations enable, illustrated in Fig. 4.1.

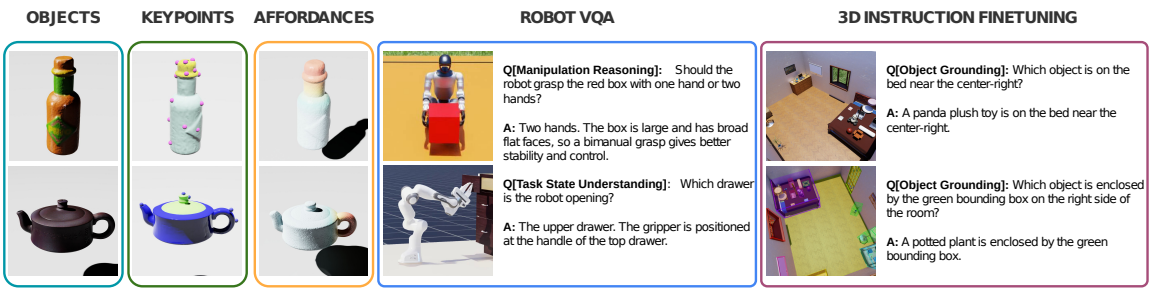

Figure 4.1. Downstream applications supported by ScalableAnno. The generated visual-language-action annotations can be consumed by atomic skills for simulation data collection, and converted into supervision for affordance and keypoint detection, robot reasoning VQA, and 3D visual-language instruction tuning.

4.1. Large-scale Simulation Data Collection

The action annotations are designed to serve as executable interfaces for simulation-based robot data collection. We build an **atomic-skill library** aligned with our annotation schema, where each skill consumes the action parameters produced by the pipeline: grasp poses, target parts, waypoints, insertion directions, hanging anchors, and navigation goals.

The library covers tabletop, bimanual, whole-body, dexterous, and mobile manipulation, and atomic skills can be composed into long-horizon tasks.

Data collection itself runs as a set of **heterogeneous parallel rollouts**. Each simulation environment independently samples assets, tasks, object poses, and scene layouts, with domain randomization over pose, lighting, material, camera, and scene configuration. For each rollout we use asynchronous cuRobo-v2 planning for goal reaching and obstacle avoidance [27, 28]. For each randomized scene the planner retrieves candidates from the annotation bank, solves goal-set inverse kinematics, and executes the feasible solution with the lowest planning cost. Candidates that repeatedly fail simulation validation are removed from the bank, so that the retained annotations stay executable under diverse robot–object configurations.

This setup is not the core contribution of the thesis: it is included to demonstrate that ScalableAnno's annotations can be directly consumed by an existing data-collection pipeline without further per-asset engineering.

4.2. Other Annotation-derived Applications

Beyond data collection, the same annotation bank supports several supervision-style applications. **Keypoint and affordance learning** reuses the visual annotations as keypoint supervision and physics-validated action annotations as affordance supervision, which can be projected to images or point clouds for perception training. **Robot reasoning VQA** can be generated as a by-product of simulation rollouts: during execution we log grounded labels such as the selected object, part anchor, inverse-kinematics solution, and approach side, which yield runtime QA supervision that cannot be derived from static

assets alone. **3D visual-language instruction tuning** can also benefit, since simulation assets and rollouts provide dense grounding labels (3D and 2D boxes, part segmentations) and the language annotations provide object relations, room layouts, and task contexts, yielding instruction–response pairs for 3D spatial reasoning and scene-level QA.

These applications are presented as illustrative downstream uses rather than full benchmarks; the main empirical evidence is reported in Chapter 5.

CHAPTER 5

Experiments

This chapter evaluates ScalableAnno as an annotation pipeline, focusing on three questions: (Q1) can the framework convert heterogeneous raw 3D assets into manipulation-ready annotations at scale? (Q2) how good are the visual-language annotations relative to direct VLM or affordance-only baselines? (Q3) can policies trained from annotation-enabled simulation rollouts transfer to the real world?

5.1. Setup

Asset pool and audited suite. We process a heterogeneous asset pool of 17,005 assets drawn from 9 source families, covering rigid objects, articulated objects, deformable or garment assets, and room-scale scenes. All quality and success-rate results are reported on an *audited evaluation suite* sampled from this pool. The suite is stratified by valid asset–skill pairs across grasping, dexterous grasping, bimanual grasping, bimanual dexterous grasping, articulation, insertion, hanging, deformable manipulation, and navigation/approach-target generation.

Coverage. On the audited evaluation suite the pipeline generates on average 2,315 candidates per attempted asset–skill pair, of which 615 pass physics validation and 538 are retained in the final annotation bank (a 26.6% validation pass rate and 23.2% retention rate). Aggregated over the full asset pool this yields roughly 100M physics-validated action annotations across 18 atomic skills.

5.2. Visual-Language Annotation Quality

The visual-language stage provides the priors that the physics stage grounds, so its quality is the main lever for the overall pipeline. We evaluate it on a manually annotated subset of the audited suite [29–32], which provides references for language descriptions, functional parts, keypoints, affordance regions, and scene-level spatial cues.

Complete visual-language bundles are scored by human raters on a 0–100 scale (reported as mean ± standard error) along five axes: Semantic, 3D Grounding, Coverage, Action-readiness, and Overall. We additionally report aggregate matching scores against manual references for the generated parts, affordances, and keypoints. We compare against three variants: **Direct VLM** (no 3D refinement), **Aff.-only + heur.** (affordance heuristics without language reasoning), and **w/o 3D refine** (language priors without 3D-grounding refinement).

Table 5.1. Visual-language and visual annotation quality on the audited evaluation suite. The left block reports mean ± standard-error human ratings for complete visual-language bundles. The right block reports aggregate matching scores for generated part, affordance, and keypoint annotations.

	Visual-Language Bundle Quality					**Visual Annotation Quality**		
Method	Semantic	3D Grounding	Coverage	Action	Overall	Part	Afford.	Keypoint
Direct VLM	84.2±1.1	61.5±1.9	68.7±1.7	64.1±1.8	69.6±1.4	72.8	68.5	70.2
Aff.-only + heur.	76.8±1.5	70.2±1.8	73.1±1.6	67.5±1.7	71.9±1.4	70.5	79.3	68.8
w/o 3D refine	87.5±0.9	73.4±1.6	77.8±1.4	75.2±1.5	78.5±1.2	82.6	78.1	80.8
Ours	**93.1±0.6**	**89.4±0.8**	**90.2±0.7**	**91.0±0.8**	**90.9±0.6**	**91.5**	**89.0**	**90.3**

As shown in Table 5.1, the full pipeline produces the best visual-language bundles. **Direct VLM** is competitive on semantics but weaker on 3D grounding and action-readiness, since it has no mechanism to anchor descriptions to 3D parts. **Aff.-only + heur.** improves

affordance localization but loses part and keypoint quality without language reasoning. **w/o 3D refine** degrades across all three visual annotation types, confirming that explicit 3D grounding is needed even when language priors are available. The combined pipeline improves overall bundle quality from 69.6 to 90.9 on the human-rated scale.

5.3. Real-World Transfer

As a sanity check on the executability of annotation-enabled rollouts, we train manipulation policies purely from simulation rollouts generated through our annotation bank (with domain randomization) and evaluate them zero-shot on representative real-world tasks. We report success rates over 20 trials per task on a single platform.

Table 5.2. Zero-shot real-world transfer of policies trained from annotation-enabled simulation rollouts. Success counts are reported over 20 trials per task.

Metric	Grasp	DexGrasp	Insertion	Open Drawer	Close Lid	Long-horizon
Success / 20	18	15	14	16	17	12

Table 5.2 shows that policies trained only on annotation-enabled simulation rollouts transfer to the real platform on basic atomic skills (Grasp, DexGrasp, Insertion, Open Drawer, Close Lid) with success rates between 14/20 and 18/20. Performance drops on the long-horizon task (12/20), as expected for compositional execution without real-world fine-tuning. These numbers are meant as a feasibility check rather than a benchmark, but they support the claim that the annotation bank produces executable, transfer-friendly trajectories.

CHAPTER 6

Conclusion

This thesis presented **ScalableAnno**, a framework for automatically converting raw 3D assets into manipulation-ready annotations. The framework combines a hierarchical visual-language annotation pipeline that injects human interaction priors and grounds them in each asset's geometry, with a physics-based action annotation pipeline that turns those priors into diverse, executable, and validated action labels across rigid, articulated, deformable, and room-scale interactions. Built on top of the resulting annotation bank, we demonstrated a parallel simulation data-collection system as well as several annotation-derived applications. Experiments on a heterogeneous audited evaluation suite showed that the full pipeline produces substantially higher-quality visual-language annotations than direct VLM or affordance-only baselines, and policies trained from annotation-enabled simulation rollouts transferred zero-shot to a real platform on a set of representative manipulation tasks.

Limitations and future work. The current pipeline depends on the quality of the visual-language model used for asset-level reasoning, and its physics validation is limited to the simulator's contact and deformation models. Future directions include richer affordance priors from interaction videos, tighter coupling between language and physics validation, and broader real-world evaluation on long-horizon and mobile manipulation tasks.

References

[1] Physical Intelligence. Pi-0.7: A steerable generalist robotic foundation model with emergent capabilities. arXiv preprint, 2026. CorpusID: 287607456.

[2] NVIDIA. Gr00t n1: An open foundation model for generalist humanoid robots. arXiv:2503.14734, 2025.

[3] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent

Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. `https://arxiv.org/abs/2310.08864`, 2023.

[4] Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. Droid: A large-scale in-the-wild robot manipulation dataset. arXiv:2403.12945, 2024.

[5] Guo Ye, Zexi Zhang, Xu Zhao, Shang Wu, Haoran Lu, Shihan Lu, and Han Liu. Learning to feel the future: Dreamtacvla for contact-rich manipulation. *ArXiv*, abs/2512.23864, 2025. URL `https://api.semanticscholar.org/CorpusID:284350273`.

[6] Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, Yutong Liang, Dylan Goetting, Chaoyi Xu, Haozhe Chen, Yuxi Qian, Yiran Geng, Jiageng Mao, Weikang Wan, Mingtong Zhang, Jiangran Lyu, Siheng Zhao, Jiazhao Zhang, Jialiang Zhang,

Chengyang Zhao, Haoran Lu, Yufei Ding, Ran Gong, Yuran Wang, Yuxuan Kuang, Ruihai Wu, Baoxiong Jia, Carlo Sferrazza, Hao Dong, Siyuan Huang, Yue Wang, Jitendra Malik, and Pieter Abbeel. Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning. *ArXiv*, abs/2504.18904, 2025. URL `https://api.semanticscholar.org/CorpusID:277741767`.

[7] Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Munoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

[8] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Jose Martinez, Hang Yin, Michael Lingelbach, Minjune Hwang, Ayano Hiranaka, Sujay Garlanka, Arman Aydin, Sharon Lee, Jiankai Sun, Mona Anvari, Manasi Sharma, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Yunzhu Li, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, Fei-Fei Li, and Salesforce AI Research. Behavior-1k: A human-centered, embodied ai benchmark with 1, 000 everyday activities and realistic simulation. *ArXiv*, abs/2403.09227, 2024. URL `https://api.semanticscholar.org/CorpusID:268520018`.

[9] Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, Yaping Li, Ping Wang, Junhao Cai, Jia Zeng, Hao Dong, and Jiangmiao Pang. Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy. *ArXiv*, abs/2511.16651, 2025. URL `https://api.semanticscholar.org/CorpusID:283110021`.

[10] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, XianLian Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *ArXiv*, abs/2506.18088, 2025. URL `https://api.semanticscholar.org/CorpusID:279999539`.

[11] Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 27649–27660, June 2025.

[12] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2023.

[13] Wenbo Wang, Fangyun Wei, QiXiu Li, Xi Chen, Yaobo Liang, Chang Xu, Jiaolong Yang, and Baining Guo. Mobilemanibench: Simplifying model verification for mobile manipulation. *arXiv preprint arXiv:2602.05233*, 2026.

[14] Sizhe Yang, Yiman Xie, Zhixuan Liang, Yang Tian, Jia Zeng, Dahua Lin, and Jiangmiao Pang. Ultradexgrasp: Learning universal dexterous grasping for bimanual robots with synthetic data, 2026. URL `https://arxiv.org/abs/2603.05312`.

[15] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3891–3902, 2023.

[16] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11359–11366. IEEE, 2023. URL `https://arxiv.org/abs/2210.02697`.

[17] Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21190–21200, 2023.

[18] Jiayi Chen, Yubin Ke, and He Wang. Bodex: Scalable and efficient robotic dexterous grasp synthesis using bilevel optimization, 2025. URL `https://arxiv.org/abs/2412.16490`.

[19] Mu Lin, Yi-Lin Wei, Jiaxuan Chen, Yuhao Lin, Shuoyu Chen, Jiangran Lyu, Jiayi Chen, Yansong Tang, He Wang, and Wei-Shi Zheng. Bidexgrasp: Coordinated bimanual dexterous grasps across object geometries and sizes, 2026. URL `https://arxiv.org/abs/2604.06589`.

[20] Yufei Wang, Ziyu Wang, Mino Nakura, Pratik Bhowal, Chia-Liang Kuo, Yi-Ting Chen, Zackory Erickson, and David Held. Articubot: Learning universal articulated object manipulation policy via large scale simulation. *ArXiv*, abs/2503.03045, 2025. URL `https://api.semanticscholar.org/CorpusID:276781877`.

[21] Ning Gao, Yilun Chen, Shuai Yang, Xinyi Chen, Yang Tian, Hao Li, Haifeng Huang, Hanqing Wang, Tai Wang, and Jiangmiao Pang. Genmanip: Llm-driven simulation for generalizable instruction-following manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12187–12198, 2025.

[22] Chenghao Yin, Da Huang, Di Yang, Jichao Wang, Nanshu Zhao, Chen Xu, Wenjun Sun, Linjie Hou, Zhijun Li, Junhui Wu, Zhaobo Liu, Zhen Xiao, Sheng Zhang, Lei Bao, Rui Feng, Zhenquan Pang, Jiayu Li, Qian Wang, and Maoqing Yao. Genie sim 3.0 : A high-fidelity comprehensive simulation platform for humanoid robot, 2026. URL `https://arxiv.org/abs/2601.02078`.

[23] Abhay Deshpande, Maya Guru, Rose Hendrix, Snehal Jauhri, Haoquan Fang, Wilbert Pumacay, Yejin Kim, Quinn Pfeifer, Ying-Chun Lee, Piper Wolters, Omar Rayyan, Mingtong Zhang, Karen Farley, Winson Han, Eli Vanderbilt, Dieter Fox, Ali Farhadi, Georgia Chalvatzaki, Dhruv Shah, and Ranjay Krishna. Molmob0t: Large-scale simulation enables zero-shot manipulation. Preprint, 2026. CorpusID: 286498371.

[24] Qwen Team. Qwen3.5-omni technical report, 2026. URL `https://arxiv.org/abs/2604.15804`.

[25] Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Yanwen Guo, Zhuo Chen, and Chunchao Guo. P3-sam: Native 3d part segmentation. *arXiv preprint arXiv:2509.06784*, 2025.

[26] Xinhao Yan, Jiachen Xu, Yang Li, Changfeng Ma, Yunhan Yang, Chunshi Wang, Zibo Zhao, Zeqiang Lai, Yunfei Zhao, Zhuo Chen, et al. X-part: High fidelity and structure coherent shape decomposition. *arXiv preprint arXiv:2509.08643*, 2025.

[27] Balakumar Sundaralingam, Adithyavairavan Murali, and Stan Birchfield. curobov2: Dynamics-aware motion generation with depth-fused distance fields for high-dof robots, 2026.

[28] 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. URL `https://arxiv.org/abs/2310.17274`.

[29] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019.

URL https://openaccess.thecvf.com/content_CVPR_2019/html/Mo_PartNet_A_Large-Scale_Benchmark_for_Fine-Grained_and_Hierarchical_Part-Level_3D_CVPR_2019_paper.html.

[30] Penghao Wang, Yiyang He, Xin Lv, Yukai Zhou, Lan Xu, Jingyi Yu, and Jiayuan Gu. Partnext: A next-generation dataset for fine-grained and hierarchical 3d part understanding. *arXiv preprint arXiv:2510.20155*, 2025.

[31] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. *arXiv preprint arXiv:2002.12687*, 2020.

[32] Yuandong Liu, Yan Wei, Yue Jiang, et al. Laso: Language-guided affordance segmentation on 3d object. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.