RoboCOIN: An Open-Sourced Bimanual Robotic Data COllection for INtegrated Manipulation

Shihan Wu [1,2*†] Xuecheng Liu [1*†] Shaoxuan Xie [1*†] Pengwei Wang [1*] Xinghang Li [1*]
Bowen Yang [3] Zhe Li [3] Kai Zhu [3] Hongyu Wu [1,4] Yiheng Liu [1,4] Zhaoye Long [1,5] Yue Wang [1]
Chong Liu [1,4] Dihan Wang [1,4] Ziqiang Ni [1] Xiang Yang [1] You Liu [1] Ruoxuan Feng [1,6] Runtian Xu [1,4]
Lei Zhang [1,7] Denghang Huang [1,8] Chenghao Jin [1,9] Anlan Yin [1,10] Xinlong Wang [1] Zhenguo Sun [1]
Mengfei Du [1] Mingyu Cao [1] Xiansheng Chen [1] Hongyang Cheng [1] Xiaojie Zhang [1] Junkai Zhao [1]
Cheng Chi [1] Sixiang Chen [1,11] Huaihai Lyu [1,7] Xiaoshuai Hao [1] Yankai Fu [1,11] Yequan Wang [1] Bo Lei [1]
Dong Liu [1] Xi Yang [1] Yance Jiao [1] Tengfei Pan [1] Yunyan Zhang [1] Songjing Wang [1] Ziqian Zhang [1]
Xu Liu [1] Ji Zhang [12] Caowei Meng [13] Zhizheng Zhang [13,1] Jiyang Gao [14] Song Wang [15] Xiaokun Leng [15]
Zhiqiang Xie [16] Zhenzhen Zhou [16] Peng Huang [17] Wu Yang [17] Yandong Guo [18] Yichao Zhu [18]
Suibing Zheng [19] Hao Cheng [20] Xinmin Ding [21] Yang Yue [22] Huanqian Wang [22] Chi Chen [22]
Jingrui Pang [1,22] YuXi Qian [23] Haoran Geng [24] Lianli Gao [2] Haiyuan Li [4] Bin Fang [4,1] Gao Huang [22,1]
Hao Dong [11,1] He Wang [11,12,1] Hang Zhao [22,13] Yadong Mu [11,1] Di Hu [6,1] Hao Zhao [22,1]
Shanghang Zhang [11,1‡] Yonghua Lin [1‡] Zhongyuan Wang [1‡] and Guocai Yao [1†‡]

[1] Beijing Academy of Artificial Intelligence [2] University of Electrical Science and Technology of China
[3] Ant Digital Technologies, Ant Group [4] Beijing University of Posts and Telecommunications
[5] Harbin Institute of Technology [6] Renmin University of China [7] Chinese Academy of Sciences
[8] Huazhong University of Science and Technology [9] University of Cambridge [10] Harbin Engineering University
[11] Peking University [12] Southwest Jiaotong University [13] Galbot [14] Galaxea [15] Leju Robotics [16] Agilex Robotics
[17] TQ-Artisan [18] AI^2 Robotics [19] Realman Robotics [20] Booster Robotics [21] DORA Community
[22] Tsinghua University [23] Stanford University [24] University of California, Berkeley
* Equal Contribution, † Project Leaders, ‡ Corresponding Authors
shihan.wu.koorye@outlook.com, gcyao1@baai.ac.cn

https://FlagOpen.github.io/RoboCOIN/

Fig. 1: Overview of our RoboCOIN dataset.

***Abstract*—Bimanual manipulation is essential for achieving human-like dexterity in robots, but the large-scale and diverse bimanual robot datasets remain scarce due to hardware heterogeneity across robotic platforms. To address the challenge, we present RoboCOIN, a comprehensive multi-embodiment bimanual manipulation dataset with over 180,000 demonstrations collected from 15 distinct robotic platforms. The dataset covers 16 scenarios including residential, commercial, working environments, with 421 tasks systematically organized by bimanual coordination patterns and object properties. Our key innovation is a hierarchical capability pyramid that provides multi-level annotations, spanning trajectory-level concepts, segment-level subtasks, and frame-level kinematics. We further develop CoRobot, a comprehensive processing framework featuring Robot Trajectory Markup Language (RTML) for quality assessment, automated annotation generation, and unified multi-embodiment management. Extensive experiments demonstrate the reliability and effectiveness of RoboCOIN in multi-embodiment bimanual learning, with significant performance improvements across various model architectures and robotic platforms. The complete dataset and framework are open-sourced and publicly available for further research purposes.**

I. Introduction

Bimanual manipulation stands as a fundamental capability enabling robots to perform complex, human-like tasks in unstructured real-world environments such as manufacturing[28], home assistance[16], and logistics[12]. High-quality bimanual demonstration data is critically important for training data-driven robot learning methods-including imitation learning[40] and reinforcement learning[8]-that rely on large-scale, diverse datasets to generalize across tasks and environments. Such data enables models to learn complex coordination strategies, understand physical interactions, and develop robust control policies for bimanual systems.

Despite significant progress in data-driven robotics, existing datasets remain limited in diversity and structure, particularly for bimanual manipulation tasks. The heterogeneity across bimanual platforms makes it difficult to collect large-scale, diverse bimanual datasets that are broadly applicable. Moreover, existing datasets primarily supply action trajectories for imitation learning, but omit the structural learning of the manipulation process. Consequently, these limitations constrain the development of models that can generalize and adapt to new tasks or physical environments.

As depicted in Figure 1, we introduce *RoboCOIN*, a large-scale, multi-embodiment bimanual manipulation dataset containing over 180,000 demonstrations across 421 distinct tasks. These demonstrations were collected from 15 distinct robotic platforms, including dual-arm robots and humanoids with both parallel grippers and dexterous hands. All data were acquired via human teleoperation to ensure high-quality, and multi-view observations and language annotations are provided for each demonstration. A two-dimensional task taxonomy based on action coordination and object flexibility organizes the demonstrations into a comprehensive grid, enabling systematic task design and progressive skill acquisition.

To enable effective structural learning across diverse embodiments, RoboCOIN introduces a hierarchical capability pyramid with three structured annotation levels: (a) *trajectory-level* annotations capture global concepts and task objectives for holistic planning; (b) *segment-level* annotations decompose tasks into executable subtasks with temporal alignment for structured reasoning; and (c) *frame-level* annotations provide dense details including kinematic states and action labels for precise control. This multi-resolution framework supports learning from high-level conceptual understanding to low-level control, facilitating advanced reasoning and generalization in bimanual manipulation.

To efficiently support the data construction and deployment infrastructure of RoboCOIN, we developed the CoRobot framework comprising three key components: (a) a Robot Trajectory Markup Language (RTML) checker that validates trajectory properties including motion smoothness and task-stage consistency to ensure physical and semantic integrity across platforms; (b) a hierarchical annotation toolchain combining visual language models with rule-based methods to automate scene description and state transition labeling; (c) and a unified platform which extends LeRobot[7] to provide unified multi-embodiment control, data management, and deployment capabilities, establishing a foundational infrastructure for scalable multi-embodiment learning. This integrated infrastructure ensures consistent data quality across diverse hardware platforms while providing the necessary tools for scalable multi-embodiment learning.

To validate the RoboCOIN dataset's multi-embodiment applicability, we developed a general enhancement method that incorporates the hierarchical capability pyramid into diverse architectures while preserving their original network structures and parameters, enabling high-level conceptual understanding and low-level feedback control. Comprehensive evaluation across multiple models and robotic platforms demonstrates consistent performance improvements in bimanual manipulation tasks spanning RoboCOIN's multi-dimensional task space. Furthermore, analysis of RTML-validated data reveals inherent limitations in human-teleoperated demonstrations, underscoring the need for unified data standards to advance multi-embodiment learning. The key contributions of this work can be summarized as follows:

- **Large-Scale, Multi-Embodiment Bimanual Dataset.** We introduce RoboCOIN, a comprehensive dataset featuring over 180,000 demonstrations across 421 tasks, collected from 15 distinct robotic platforms.
- **Hierarchical Capability Pyramid.** We propose a hierarchical capability pyramid with trajectory-level, segment-level, and frame-level descriptions, enabling multi-resolution learning from high-level global concepts to low-level control.
- **Integrated Data Processing Framework.** We develop a unified data processing framework CoRobot, including RTML-based assessment, an automated annotation toolchain, and a platform for unified multi-embodiment dataset management and robot deployment.

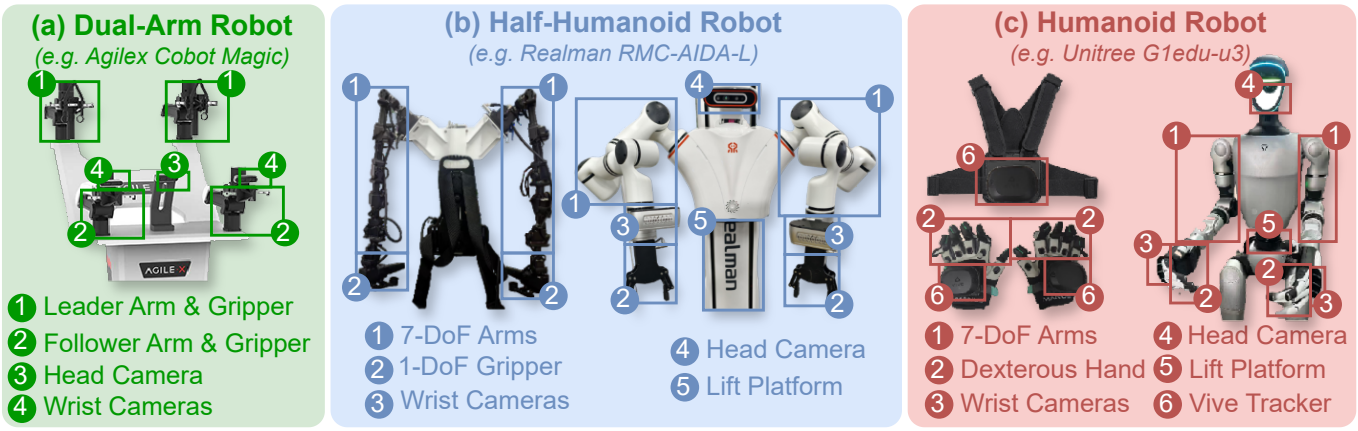

Fig. 2: Data collection platforms of our RoboCOIN. (a) **Dual Robot (e.g. Agilex Cobot Magic).** A bimanual robot with two 6-DoF arms and parallel grippers, capable of performing complex bimanual tasks. (b) **Half-Humanoid Robot (e.g. Realman RMC-AIDA-L).** Left: wearable teleoperation device for human demonstration, right: bimanual robot with parallel grippers. (c) **Humanoid Robot (e.g. Unitree G1edu).** Left: wearable motion capture device for human demonstration, right: humanoid robot with advanced dexterous manipulation capabilities. Head and wrist cameras provide multi-view visual observations.

TABLE 1: Robotic platforms used in RoboCOIN dataset collection.

Type	Name	Arm DoF	Camera Configuration	Gripper Type	Teleoperation Method
Dual-Arm	Agilex Cobot Magic[2]	2×6	Head + Wrist	Parallel Gripper	Isomorphic Arm
	Agilex Split ALOHA[2]	2×6	Head + Wrist	Parallel Gripper	Isomorphic Arm
	Galaxea R1 Lite[14]	2×6	Head + Wrist	Parallel Gripper	Isomorphic Arm
Half-Humanoid	Realman RMC-AIDA-L[32]	2×7	Head + Wrist	Parallel Gripper	Exoskeleton
	Agibot G1[1]	2×7	Head (w/ Depth) + Wrist + Back	Parallel Gripper	Virtual Reality
	AI^2 AlphaBot 2[30]	2×7	Head + Wrist + Chest	Dexterous Hand	Virtual Reality / Isomorphic Arm
	AI^2 AlphaBot 1s[30]	2×7	Head + Wrist + Chest	Dexterous Hand	Virtual Reality / Isomorphic Arm
	Galbot G1[15]	2×7	Head + Wrist	Parallel Gripper	Motion Capture
	Tianqing A2[33]	2×7	Head (w/ Depth) + Wrist + Back	Parallel Gripper	Virtual Reality
	Realman Rs-02[32]	2×7	Head (w/ Depth) + Wrist	Parallel Gripper	Exoskeleton
	Realman Rs-01[32]	2×7	Head (w/ Depth) + Wrist	Parallel Gripper	Exoskeleton
	Airbot MMK2[3]	2×6	Head + Wrist + Third-Person	Dexterous Hand	Virtual Reality
	Leju Kuavo 4 LB[31]	2×7	Head + Wrist	Dexterous Hand	Virtual Reality
Humanoid	Leju Kuavo 4 Pro[31]	2×7	Head + Wrist	Dexterous Hand	Virtual Reality
	Unitree G1edu-u3[37]	2×7	Head + Wrist	Dexterous Hand	Motion Capture / Exoskeleton

II. Related Work

Robotic Learning Datasets. The evolution of robot learning has been significantly driven by the availability of diverse and scalable demonstration datasets. Early robot learning efforts, constrained by hardware limitations, primarily collected data in simulation environments such as Meta-world[41], LIBERO[22], and CALVIN[24], but they often struggle to transfer to real-world scenarios due to sim-to-real gaps[43]. While aggregating real-world, multi-embodiment trajectories to enhance policy generalization, the Open-X-Embodiment dataset remains limited to single-arm tasks, restricting its applicability to complex bimanual interactions in real-world scenarios. A notable addition is the π_0 dataset[5], which offers an extensive collection of bimanual demonstrations featuring long-horizon tasks. However, the π_0 dataset remains proprietary and closed-source, limiting its accessibility to the broader research community. More recently, datasets such as AgiBot World and Galaxea Open-World have emerged to address the need for large-scale bimanual data in open-world settings. While AgiBot World is notable for its immense scale and industrial-grade data quality collected across multiple scenarios, Galaxea Open-World distinguishes itself with high-quality, fine-grained annotations from a unified mobile bimanual platform across numerous real-world scenes. While these newer datasets expand the scale of bimanual data, they are often constrained by their reliance on a single robot embodiment due to commercial considerations, limiting multi-embodiment applicability.

Large-scale Robotic Learning Policies. The development of robot learning policies has evolved significantly from specialized, small-scale models to large-scale, generalist systems, especially in the domain of bimanual manipulation. Early methods such as Action Chunking with Transformers (ACT)[42] and Diffusion Policy[9] demonstrated acceptable performance on specific tasks using small-scale datasets. However, both methods were limited by their training data's scale and diversity, which spurred the development of generalist Vision-Language-Action (VLA) models via large-scale datasets. For instance, Octo[36] trained on the Open X-Embodiment dataset-supports both language and goal-image conditioning, facilitating zero-shot adaptation to novel tasks

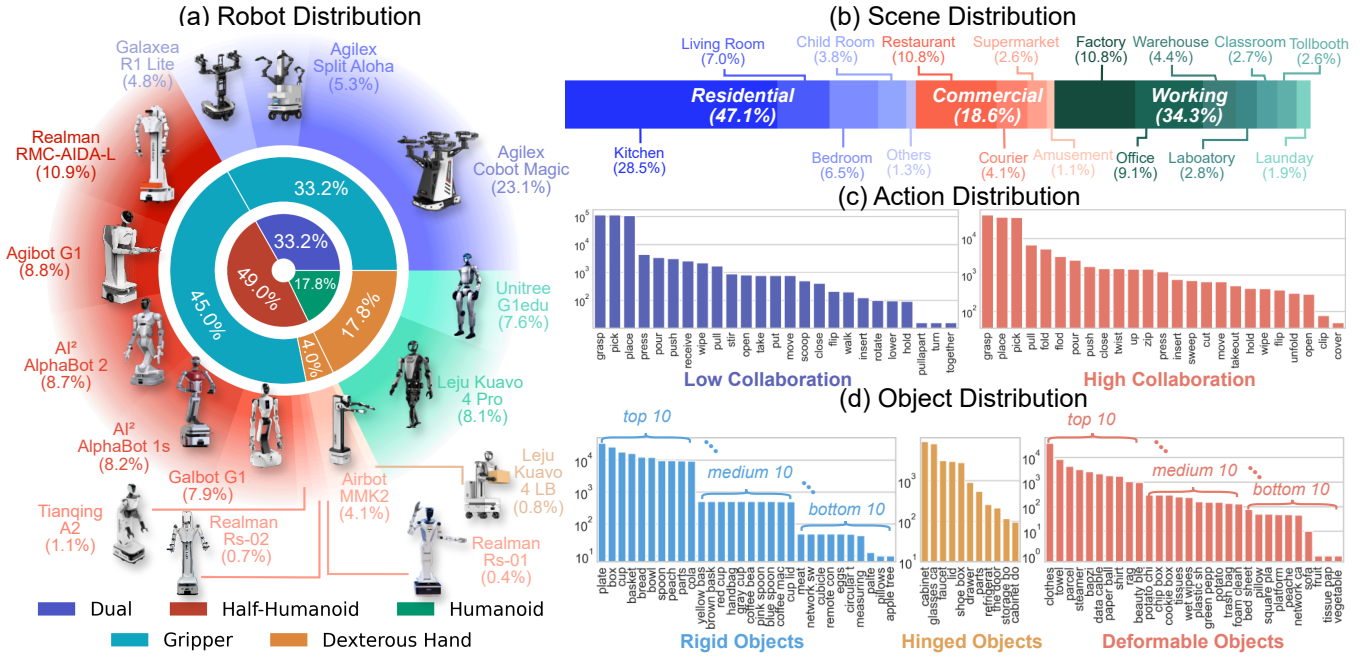

Fig. 3: Overview statistics of the RoboCOIN dataset. RoboCOIN incorporates (a) 15 distinct robotic platforms, including bimanual, half-humanoid, and humanoid robots with grippers and dexterous hands; (b) diverse environments such as residential, commercial, and working scenes; (c) 36 action types categorized by collaboration levels; and (d) 432 object types spanning rigid, articulated, and deformable categories.

and robots. In contrast, OpenVLA[21] adopts an autoregressive architecture based on large language models (LLMs), tokenizing continuous robot actions into discrete representations compatible with its language model backbone. Robotics Diffusion Transformer (RDT)[23] is a pioneering diffusion-based foundation model for bimanual manipulation. RDT is capable of learning from heterogeneous modalities and diverse robot architectures through its scalable transformer architecture and physically interpretable unified action space. Meanwhile, π_0[5] employs a flow-matching architecture, combining a vision-language model (VLM) for perception and semantic reasoning with a separate action expert network dedicated to generating continuous, high-precision motor commands. The GR00T-N1[25] model established a dual-system architecture for humanoid robots, where system-2 interprets the environment and system-1 generates real-time motor actions, with its successor GR00T-N1.5[26] introducing key enhancements including an upgraded vision-language model and a future state alignment objective (FLARE). Alternatively, specialized models like GO-1[6] and G0[18] for specific robot platforms achieve effective single-embodiment bimanual control through multi-stage training and hierarchical systems. Building upon the RoboBrain 2.0[34] dataset by incorporating real robot action data, RoboBrain-X0[35] achieves integrated perception-to-execution capabilities and zero-shot cross-embodiment generalization through unified modeling of vision, language, and action. Despite their scale, these models often lack structured understanding of task hierarchies, limiting their ability to reason about complex bimanual manipulation tasks.

III. RoboCOIN Dataset

The RoboCOIN dataset provides a multi-embodiment benchmark for bimanual manipulation, integrating 14 robotic platforms, a multi-dimensional task taxonomy based on action coordination and object flexibility, and a hierarchical capability pyramid spanning trajectory-level, segment-level, and frame-level annotations. This structure supports learning from high-level concepts to low-level control, facilitating advanced reasoning and generalization in bimanual manipulation.

A. Data Collection and Storage

The RoboCOIN framework leverages a diverse set of 14 robotic platforms for comprehensive data acquisition, encompassing bimanual, half-humanoid, and humanoid configurations. Figure 2 illustrates three representative platforms: bimanual robots (e.g. Agilex Cobot Magic), half-humanoid robots (e.g. Realman RMC-AIDA-L), and humanoid robots (e.g. Unitree G1edu). The framework employs teleoperation to ensure high-quality data collection, utilizing methods such as (a) leader-follower isomorphic arms, (b) exoskeletons, and (c) motion capture systems. The complete list of robotic platforms is detailed in Table 1. The platforms are equipped with a comprehensive suite of sensors. These capture multimodal data streams (RGB and depth) from multiple camera views (e.g., head, wrist, third-person, chest and back), along with the robot's kinematic state (including joint angles, end-effector

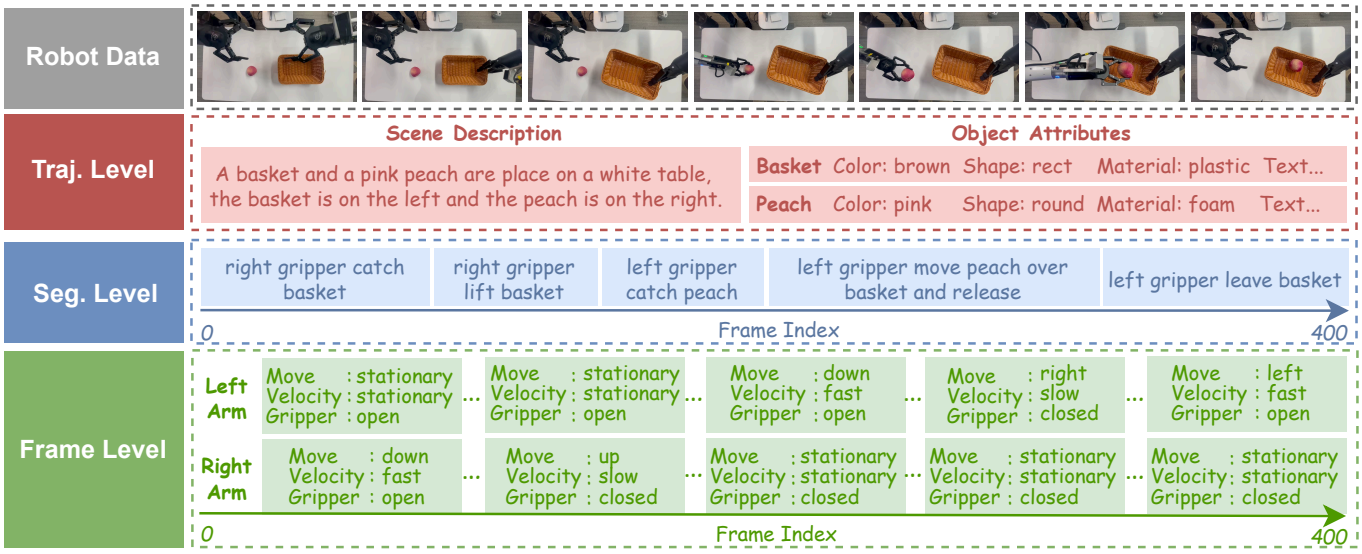

Fig. 4: The RoboCOIN framework introduces a hierarchical capability pyramid, structured across three levels: (a) trajectory-level annotations define the global concepts and task objectives; (b) segment-level annotations decompose the task into executable subtasks; and (c) frame-level annotations provide dense low-level details such as motion trajectories and gripper states. All annotations are temporally synchronized to form a cohesive data structure.

TABLE 2: Comparison of existing real-world datasets for robot manipulation. All data is drawn from the original paper or RoboMIND paper. †not a dataset in itself, but an *aggregation* of existing datasets.

Dataset	Arm	Embodiment	Trajectory	Task	Skill	Dexterous	Annotation	Collection Method
Pinto and Gupta[29]	Dual	1	50k	n/a	1	✗	No	Scripted
RoboNet[10]	Single	1	162k	n/a	n/a	✗	No	Scripted
MT-Opt[19]	Single	1	800k	12	1	✗	No	Scripted
BridgeData[11]	Single	1	7.2k	71	4	✗	No	Human Teleoperation
BC-Z[17]	Single	1	26k	100	3	✗	No	Human Teleoperation
RH20T[13]	Single	1	13k	140	33	✗	No	Human Teleoperation
RoboSet[4]	Single	1	98k	38	6	✗	No	30% Human / 70% Scripted
BridgeData V2[38]	Single	1	60k	n/a	13	✗	No	85% Human / 15% Scripted
DROID[20]	Single	1	76k	n/a	86	✗	No	Human Teleoperation
Open X-Embodiment†[27]	Single+Dual	22	1.4M	160k	217	✗	No	Dataset Aggregation
RoboMIND[39]	Single+Dual	4	107k	479	38	✓	Flat	Human Teleoperation
AgiBot World Beta[6]	Dual	1	1M	217	87	✗	Flat	Human Teleoperation
Open Galaxea[18]	Dual	1	50k	150	58	✓	Flat	Human Teleoperation
RoboCOIN	**Dual**	**15**	180K+	421	36	✓	**Hierarchical**	Human Teleoperation

poses, and gripper articulation). Essential environmental parameters, such as platform elevation and workspace geometry, are also recorded. All kinematic measurements adhere to standardized conventions: distances are in meters, end-effector orientations are represented by 6D rotation matrices within a unified left-handed coordinate system, and the gripper state is normalized to a continuous value from 0 to 1. Strict temporal synchronization across all data streams is maintained through timestamp alignment, ensuring consistency between visual observations and kinematic states.

B. Statistics and Taxonomy

The RoboCOIN dataset is built on a multi-embodiment foundation to ensure broad applicability across diverse robotic platforms. Figure 3(a) shows the overall distribution of robotic platforms in RoboCOIN, encompassing dual-arm, half-humanoid, and humanoid types. The 15 distinct platforms offer rich morphological diversity for complex bimanual manipulation. The dataset emphasizes half-humanoid robots as a mainstream architecture balancing human-like form with practical hardware requirements, while dual-arm systems offer a cost-effective solution for basic bimanual coordination and humanoid platforms deliver advanced dexterity with fully articulated hands. As shown in Figure 3(b), data collection spans 16 distinct scenarios, categorized into residential, commercial, and working environments. Residential settings constitute the majority, as they exhibit the greatest diversity of tasks and objects and are most intimately connected to daily human life. Each category contains finer subdivisions, with commercial scenarios including restaurant, courier, supermarket, and amusement settings, thereby capturing a wide range of real-world applications.

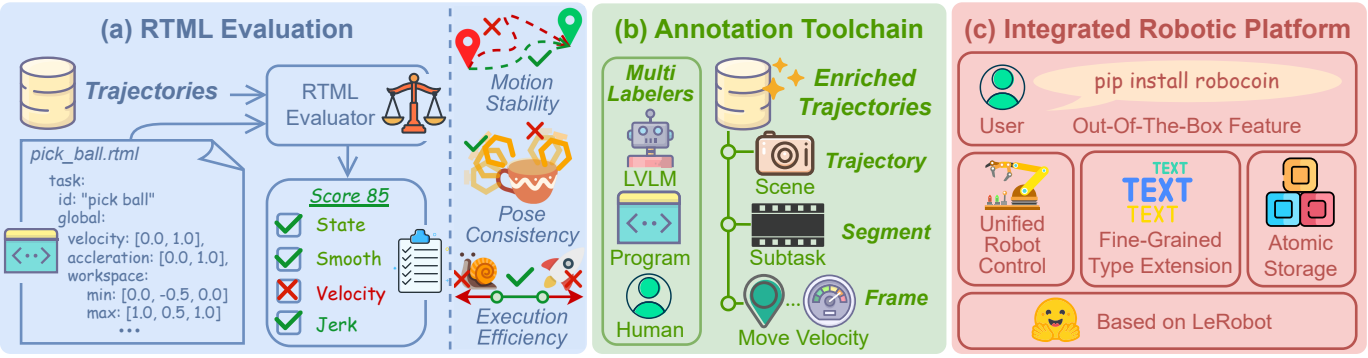

Fig. 5: Overview of our CoRobot data processing framework. (a) Robot Trajectory Markup Language (RTML) for automated trajectory validation. (b) Semi-automatic annotation toolchain for generating rich and hierarchical task descriptions. (c) An out-of-the-box integrated robotic platform for unified robot control and multi-embodiment data management.

RoboCOIN employs a two-dimensional task taxonomy that organizes manipulation scenarios along action coordination (Figure 3(c)) and object flexibility (Figure 3(d)). 36 action patterns are categorized according to the degree of bimanual coordination required, distinguishing between low-coordination tasks (where arms operate largely sequentially) and high-coordination tasks (featuring partial or fully parallel arm movements). Similarly, 432 objects are classified along a spectrum of mobility ranging from rigid bodies with fixed poses to articulated objects with constrained motion (e.g., hinges and joints) and finally to fully deformable objects that may undergo significant shape changes during manipulation. This integrated framework facilitates the creation of diverse and incrementally challenging tasks that support skill development across different robotic platforms and real-world settings.

C. Hierarchical Capability Pyramid

As illustrated in Figure 4, the hierarchical capability pyramid in RoboCOIN encompasses three levels of structured annotations: trajectory-level, segment-level, and frame-level, enabling multi-resolution learning from high-level conceptual understanding to low-level control.

Trajectory-level Concepts. Trajectory-level annotations represent the concepts of a task by describing the overall scene configuration. This includes scene description (environment settings, object placements), and detailed attributes (e.g., color, shape, material, texture, and size). The resulting integrated representation supports global spatial and physical reasoning across the task sequence.

Segment-level Subtasks. Segment-level annotations decompose tasks into specific subtasks, which may temporally overlap to accommodate dual-arm operations. Each segment is aligned with specific video frames and includes step-by-step instructions. Annotations also explicitly label exception cases, such as grasping failures, to support robust error handling. This structured decomposition facilitates learning of temporal reasoning, task planning, and error recovery.

Frame-level Kinematics. At the finest granularity, frame-level annotations describe the kinematic state for each video frame using natural language. This includes the motion parameters (e.g., direction, velocity, acceleration) for both arm end-effectors, and the status of grippers or dexterous hands (such as open/close state or transitioning movements). This high-density kinematic data enables real-time intrinsic feedback control and precise motion execution.

D. Comparison with Existing Datasets

As shown in Table 2, RoboCOIN stands out from existing robot datasets through its open and generalizable design for complex bimanual manipulation. Most available datasets are restricted to single-arm robots (e.g., BridgeData V2[38], DROID[20]) or mixed embodiments with limited bimanual coverage (e.g., Open X-Embodiment[27], RoboMIND[39]). Even dedicated dual-arm datasets such as AgiBot World Beta[6] and Open Galaxea[18] remain confined to single robotic platforms. In contrast, RoboCOIN incorporates 15 diverse robot platforms, covering dual-arm, half-humanoid, and full-humanoid configurations, with both parallel grippers and dexterous hands. Moreover, RoboCOIN introduces a hierarchical capability pyramid with multi-level annotations, enabling structured learning from high-level concepts to low-level control.

IV. CoRobot Data Processing Framework

For the efficient construction of the RoboCOIN dataset, we developed a CoRobot, an integrated data processing framework, as illustrated in Figure 5. This framework integrates three core components: (a) a Robot Trajectory Markup Language (RTML) for automated trajectory quality assessment, (b) a semi-automatic annotation toolchain for generating hierarchical capability pyramid annotations, and (c) an out-of-the-box robotic platform for unified multi-embodiment control and data management.

A. Robot Trajectory Markup Language Evaluation

While critical for VLA model training, high-quality data collection is challenged by distribution shifts arising from human operators' varying expertise and preferences, which

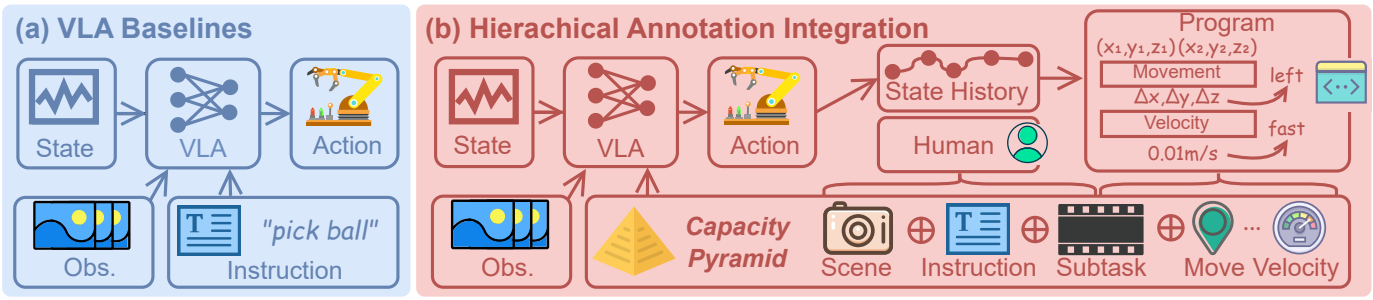

Fig. 6: Model architecture. (a) VLA Baselines. (b) Hierarchical Annotation Integration (HAI). While inference, HAI incorporates hierarchical annotations via human instructions with real-time context generated automatically through phase change detection and state history summarization, without altering the original architecture or parameters.

deleteriously impact model performance. To address this problem at the source, we propose the Robot Trajectory Markup Language (RTML), a domain-specific language designed to convert expert rules into standardized, machine-readable, and configurable constraints. This ensures that collected robot trajectory data is consistent, reliable, and adheres to defined quality principles. The design of RTML is based on three key principles for high-quality robot motion:

- **Motion Stability.** Trajectories should be smooth and predictable, avoiding sudden changes that could lead to instability or inaccuracies.
- **Pose Consistency.** During critical task phases, the robot's end-effector pose must meet task-specific constraints to ensure successful execution.
- **Execution Efficiency.** A trajectory must balance speed and precision, avoiding excessive haste or hesitation that could compromise task performance.

RTML is defined using the YAML format for readability and ease of use. It constrains trajectories from two perspectives: (a) global constraints that apply to the entire trajectory, defining motion characteristics including workspace boundaries, velocity limits, acceleration limits, and duration limits; and (b) local constraints that divide the trajectory into sequential phases (e.g. approach, grasp, place), defining override parameters and orientation tolerances for each phase. Furthermore, a RTML evaluator is provided to automatically assess trajectory quality against the defined constraints. The output includes detailed reports and an overall quality score, providing quantitative support for data selection and filtering. The detailed specification of RTML can be found in the Appendix A.

B. Annotation Toolchain

We developed a comprehensive toolchain that integrates large language models, rule-based tools, and human annotation to support hierarchical robotic data annotation. For trajectory-level annotation, object positions in the scene are first obtained via an object detection tool and then converted into natural language using a large language model. At the segment level, keyframes marking important behavioral changes are automatically identified by rule-based tools and later refined manually. For frame-level labels, we use a rule-based tool that applies a sliding window to state sequences to quantify motion between frames, which is then converted into text labels using predefined thresholds (e.g., categorizing minimal movement as "stationary"). This integrated approach streamlines the annotation process while ensuring high accuracy and consistency across multiple levels, enabling the efficient creation of large-scale, detailed datasets for complex robotic tasks.

C. Integrated Robotic Platform

The heterogeneity of robot hardware and data formats poses a major challenge for multi-embodiment learning. To address this challenge, we introduce an integrated robotic platform for unified control and data management. Built on LeRobot[7], it provides a robust infrastructure for this purpose and is characterized by three key features:

- **Unified Robot Control.** The platform integrates official SDKs for various robot platforms and supports generic control via ROS, enabling seamless operation across diverse robot hardware.
- **Fine-Grained Type Extension.** The platform enhances data handling capabilities by supporting segment-level and frame-level text annotations, facilitating detailed task breakdowns and state representations.
- **Atomic Storage.** The platform employs an atomic storage strategy, partitioning datasets into minimal subsets based on factors like embodiment, task, and environment. These subsets can be dynamically combined using tags, reducing download burdens and improving resource efficiency.

Our integrated robotic platform offers an out-of-the-box solution for multi-embodiment robot control and dataset management, significantly lowering the barrier to entry for researchers and practitioners in the field of robotic learning. It is fully open-source and encourages further development within the research community. [1]

V. Experiments and Analysis

This study aims to answer the following key research questions (RQs) through a series of experiments in real-world environments:

[1] https://github.com/FlagOpen/CoRobot

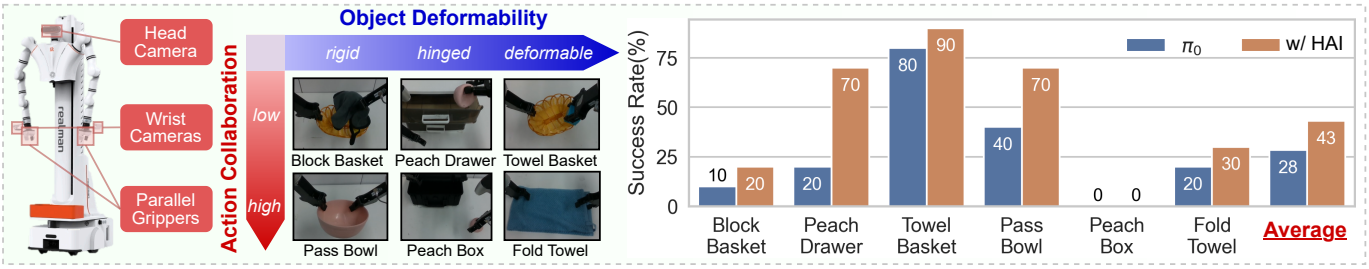

Fig. 7: The experimental task design and results of Realman RMC-AIDA-L + π_0. It follows an identical task design (left), with the results (right) showing the success rates for π_0 with and without HAI.

RQ1. How does the RoboCoin dataset perform on different VLA and multi-embodiment bimanual platforms?
RQ2. Does the hierarchical capability pyramid improve VLA models, and in what aspects?
RQ3. To what extent does the Robot Trajectory Markup Language (RTML) contribute to improving data quality and model performance?

A. Experiment Setup

Hierarchical Annotation Integration for VLAs. We propose hierarchical annotation integration (HAI) to improve robotic policy learning by adding hierarchical information to standard Vision-Language-Action (VLA) models. HAI uses annotations at three levels: trajectory-level scene context, segment-level subtask instructions, and frame-level state descriptions. As shown in Figure 6(b), these annotations come from the RoboCOIN dataset and are integrated into the VLA model as additional input tokens. This approach enriches the model's input with layered contextual information without modifying its core architecture. During training the model uses the full set of annotations, while during operation it combines human instructions with real-time context generated automatically through phase change detection and state history summarization. This allows the model to leverage hierarchical knowledge from high-level concepts to low-level control, enhancing its ability to perform complex bimanual tasks.

Evaluated VLA Models. We evaluated two VLAs using their recommended default hyperparameter settings:

- π_0[5]. A flow-matching VLA model trained on the proprietary π_0 dataset, combining a vision-language model for perception and reasoning with an action expert network for continuous motor commands.
- **GN00T N1.5**[26]. A diffusion-based VLA model trained on the Galaxea Open-World dataset, featuring a hierarchical architecture that separates high-level planning from low-level skill execution.

To achieve parameter-efficient adaptation, π_0 was fine-tuned using LoRA with $r = 16$, while GN00T N1.5 used partial fine-tuning of only the diffusion module and projector. The detailed training settings are provided in the Appendix B.

Multi-Embodiment Evaluation Platforms. To validate multi-embodiment adaptability, we selected one representative platform from each of two morphology categories:

- **Realman RMC-AIDA-L**[32], a half-humanoid robot with two 7-DoF arms mounted on a lift platform, equipped with parallel grippers.
- **Unitree G1**[37], a humanoid robot with two 7-DoF arms and dexterous hands, with full-body mobility.

All platforms were configured with head-mounted cameras and dual-wrist cameras for visual perception.

B. Impact of Hierarchical Annotation Integration

To address **RQ1** regarding the generalization of the RoboCOIN dataset, we evaluated π_0 on the Realman RMC-AIDA-L. The experimental setups are illustrated in Figure 7. Tasks were designed along a two-dimensional grid varying action coordination difficulty and object flexibility, following the taxonomy outlined in Sec. III-B. Evaluation of the π_0 model on the Realman RMC-AIDA-L platform showed competent performance on simple tasks, exemplified by an 80% success rate in the "place the towel into the basket" task. However, the π_0 model on Realman RMC-AIDA-L showed substantially lower performance on complex tasks, achieving only 40% success in "pass the bowl" and 20% in "place the peach into the drawer and close it." These results demonstrate that the RoboCOIN dataset enables effective evaluation of VLA models across diverse bimanual platforms, while also revealing significant performance variations across tasks of different complexity levels.

In response to **RQ2**, we integrated Hierarchical Annotation Integration (HAI) into both VLA models. Performance improvements were observed across task difficulties: on the Realman RMC-AIDA-L with π_0, success rates increased from 80% to 90% for the simple "place the towel into the basket" task, while the complex "place the peach into the drawer and close it" task saw a more substantial gain from 20% to 70%. These results collectively demonstrate that HAI effectively enhances VLA model performance across both simple and complex tasks, with particularly significant gains observed in challenging scenarios requiring precise coordination and object manipulation. The consistent improvements across task complexities confirm the value of hierarchical annotations for robust robotic policy learning.

C. Impact of RTML

To address **RQ3**, we conducted experiments on two tasks using the unitree g1 platform: a single-arm task "pick the

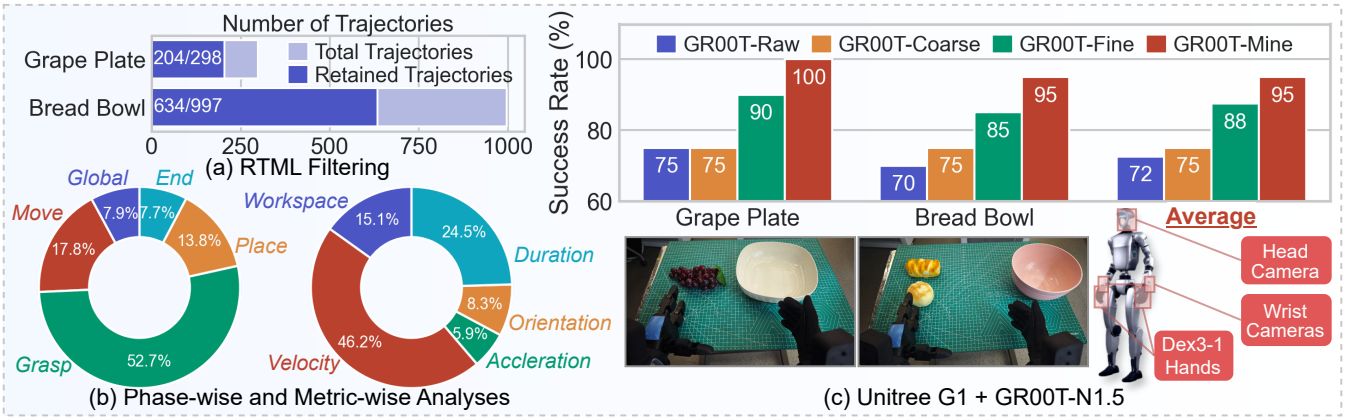

Fig. 8: Impact of RTML on data quality and model performance. (a) RTML Filtering. The amount of data removed by RTML in the two tasks. (b) Phase-wise and Metric-wise Analyses. The former identifies failure-prone operational stages, while the latter pinpoints frequently violated constraints. (c) Unitree G1 + GR00T-N1.5. The performance of the two tasks across the four fine-tuning settings: GR00T-Raw, GR00T-Coarse, GR00T-Fine, and GR00T-Mine, as measured by success rate.

grape and place into the plate" and a bimanual task "push the bowl and place bread pieces into it". As shown in Figure 8(a), RTML filtering removed an average of 35.3% of low-quality trajectories across both tasks, indicating that a substantial portion of human demonstrations contain inconsistencies that could adversely affect policy learning. As shown in Figure 8(a) (b), we analyzed trajectories from two tasks using two complementary approaches: phase-wise and metric-wise analyses. The former identified failure-prone operational stages (e.g., move, grasp, place, etc.) by assessing segments against phase-specific constraints, while the latter pinpointed frequently violated constraints (e.g., workspace, velocity, accleration, etc.) through evaluation against individual metrics. The phase-wise analysis revealed that the majority of trajectory failures occurred during the grasping phase, accounting for 52.7% of disqualifications, followed by the moving phase at 17.8%. On the other hand, the metric-wise analysis indicated that velocity violations were the most common cause of trajectory disqualification, responsible for 46.2% of failures, followed by duration violations at 24.5%. The complementary insights from both analyses highlight critical areas for improving demonstration quality, particularly in enhancing grasping techniques and adhering to velocity constraints during task execution.

To further investigate the impact of RTML on model performance, we further evaluated how RTML-filtered data affects policy learning by comparing four fine-tuning settings:

- **GR00T-Raw**: model trained on the original dataset.
- **GR00T-Coarse**: model trained on data filtered only by global RTML constraints.
- **GR00T-Fine**: model trained on data filtered by both global and phase-wise constraints.
- **GR00T-Mine**: model trained on data filtered by RTML and augmented with mined high-quality trajectory segments from other tasks.

As shown in Figure 8(c), the four configurations exhibited a clear progressive improvement in average success rates. GR00T-Coarse showed a modest 3% gain over the GR00T-Raw baseline, indicating that basic global filtering provides limited benefits. In contrast, GR00T-Fine achieved a more substantial 16% improvement, underscoring the significant impact of phase-level constraints on trajectory quality. The highest performance was attained by GR00T-Mine, which reached a 23% total gain by further incorporating relevant high-quality segments from other tasks. These results collectively demonstrate that fine-grained trajectory validation and integration contribute more substantially to policy performance than dataset scale alone, highlighting the critical role of RTML in enhancing robotic learning outcomes. More experimental details and analyses can be found in the Appendix C.

D. Limitations and Future Work

While our approach demonstrates promising results in bimanual robotic manipulation, several limitations remain to be addressed in future work. The annotation toolkit, though designed to reduce cost, may still introduce errors and requires manual verification. The RTML framework relies on empirically set thresholds, which may not generalize across all scenarios. Furthermore, our study does not include mixed-embodiment training or cross-embodiment policy transfer experiments. To address these limitations, we plan to develop more intelligent RTML filtering strategies, potentially using statistical or learning-based methods, and integrate RTML into the data collection process for real-time supervision. We will also enhance the CoRobot framework to support more modalities and robot platforms, improving its generality. Finally, we intend to conduct mixed-embodiment experiments to develop powerful bimanual policies that can transfer across different robotic platforms. These efforts are expected to strengthen the generality, efficiency, and practical utility of our framework, advancing its use in complex multi-embodiment bimanual manipulation tasks.

VI. Conclusion

We present RoboCOIN, a large-scale multi-embodiment dataset which integrates 15 robotic platforms, over 180,000 demonstrations, with 421 tasks and multiple scenarios. RoboCOIN introduces a hierarchical capability pyramid comprising trajectory-level, segment-level, and frame-level annotations, enabling structured learning from high-level concepts to low-level control. To facilitate dataset construction, we develop CoRobot, an integrated data processing framework featuring a Robot Trajectory Markup Language (RTML) for automated trajectory quality assessment, a semi-automatic annotation toolchain, and an out-of-the-box robotic platform for unified multi-embodiment control and data management. Extensive experiments demonstrate RoboCOIN's effectiveness in enhancing VLA model performance across diverse bimanual platforms, with hierarchical annotations and RTML significantly improving task success rates.

References

[1] Agibot. https://www.agibot.com/products/G1.

[2] AgileX. https://global.agilex.ai/products/cobot-magic.

[3] Airbot. https://airbots.online/mmk2.

[4] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795. IEEE, 2024.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. Pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[6] 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. *arXiv preprint arXiv:2503.06669*, 2025.

[7] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

[8] Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Zongqing Lu, Stephen McAleer, Hao Dong, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5150–5163, 2022.

[9] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[10] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[11] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[12] Wolfgang Echelmeyer, Alice Kirchheim, and Eckhard Wellbrock. Robotics-logistics: Challenges for automation of logistic processes. In *2008 IEEE International Conference on Automation and Logistics*, pages 2099–2103. IEEE, 2008.

[13] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023.

[14] Galaxea. https://galaxea-ai.com/products/R1-Lite.

[15] Galbot. https://www.galbot.com/g1.

[16] Anna Henschel, Guy Laban, and Emily S Cross. What makes a robot social? a review of social robots from science fiction to a home or hospital near you. *Current Robotics Reports*, 2(1):9–19, 2021.

[17] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[18] 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. *arXiv preprint arXiv:2509.00576*, 2025.

[19] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

[20] 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. *arXiv preprint arXiv:2403.12945*, 2024.

[21] 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. *arXiv preprint arXiv:2406.09246*, 2024.

[22] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking

knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[23] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[24] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

[25] NVIDIA, Nikita Cherniadev Johan Bjorck andFernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

[26] NVIDIA, Nikita Cherniadev Johan Bjorck andFernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. https://research.nvidia.com/labs/gear/gr00t-n1_5/, 2025.

[27] 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 *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[28] Mikkel Rath Pedersen, Lazaros Nalpantidis, Rasmus Skovgaard Andersen, Casper Schou, Simon Bøgh, Volker Krüger, and Ole Madsen. Robot skills for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing*, 37: 282–291, 2016.

[29] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.

[30] AI2 Robotics. https://ai2robotics.com/, .

[31] Leju Robotics. https://www.lejurobot.cn/zh/application/kuavo-my, .

[32] Realman Robotics. https://realmanrobotics.com/, .

[33] Tianqing Robotics. https://tqartisan.com/productDetails?type=A2, .

[34] BAAI RoboBrain Team. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

[35] BAAI RoboBrain Team. https://github.com/FlagOpen/RoboBrain-X0, 2025.

[36] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[37] Unitree. https://www.unitree.com/g1/.

[38] 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 *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[39] 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. *arXiv preprint arXiv:2412.13877*, 2024.

[40] Fan Xie, Alexander Chowdhury, M De Paolis Kaluza, Linfeng Zhao, Lawson Wong, and Rose Yu. Deep imitation learning for bimanual robotic manipulation. *Advances in neural information processing systems*, 33: 2327–2337, 2020.

[41] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[42] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[43] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pages 737–744. IEEE, 2020.

APPENDIX A
ROBOT TRAJECTORY MARKUP LANGUAGE SPECIFICATION

The Robot Trajectory Markup Language (RTML) is a domain-specific language designed to standardize the specification and evaluation of robot trajectory quality. It is structured in YAML format for readability and ease of use. An RTML document consists of two main sections: global constraints and local stage constraints. Global constraints apply to the entire trajectory, while local stage constraints define specific requirements for individual phases of the trajectory. For global and local constraints, RTML supports various parameters including workspace boundaries, velocity limits, acceleration limits, orientation tolerances, temporal duration limits, etc. An example RTML specification for the task "pull bowl storage bread" is provided in Listing 1.

Listing 1: RTML Example "pull bowl storage bread"

```yaml
# RTML V1.0
task:
  id: "pull_bowl_storage_bread"

  # Global constraints
  global_constraints:
    velocity:
      linear:
        max: 0.5          # m/s
        mean_max: 0.3     # m/s
    acceleration:
      linear:
        max: 12.0          # m/s2

  # Local stage constraints
  stages:
    - id: "move_bowl_right"
      match_subtask: "Move the pink bowl to the center of table with right hand"
      constraints:
        workspace:
          right:
            min: [0.10, -0.40, 0.10]
            max: [0.25, -0.20, 0.30]
        velocity:
          linear:
            mean_max: 0.10
            std_max: 0.08
        idle_arm:
          arm: "left"
          velocity_linear_mean_max: 0.05
        temporal:
          duration_min: 2.0
          duration_max: 6.0

    - id: "grasp_long_bread_left"
      match_subtask: "Grasp the long bread with left hand"
      constraints:
        workspace:
          left:
            min: [0.05, -0.05, -0.05]
            max: [0.25, 0.35, 0.20]
        orientation:
          left:
            angular_mean_deviation_max: 0.8
            std_max: [0.5, 0.5, 0.8]
            angular_variance_max: 0.15
        velocity:
          linear:
            mean_max: 0.12
            std_max: 0.10
        idle_arm:
          arm: "right"
          velocity_linear_mean_max: 0.05
        temporal:
          duration_min: 2.0
          duration_max: 8.0

    - id: "place_long_bread_in_bowl"
      match_subtask: "Place the long bread in pink bowl with left hand"
      constraints:
        workspace:
```

```
        left:
          min: [0.05, -0.05, -0.05]
          max: [0.25, 0.35, 0.20]
      velocity:
        linear:
          mean_max: 0.15
          std_max: 0.15
      idle_arm:
        arm: "right"
        velocity_linear_mean_max: 0.05
      temporal:
        duration_min: 1.0
        duration_max: 4.0

  - id: "grasp_round_bread_left"
    match_subtask: "Grasp the round bread with left hand"
    constraints:
      workspace:
        left:
          min: [0.05, 0.00, -0.05]
          max: [0.25, 0.35, 0.20]
      orientation:
        left:
          angular_mean_deviation_max: 0.5
          std_max: [0.5, 0.5, 0.5]
          angular_variance_max: 0.15
      velocity:
        linear:
          mean_max: 0.12
          std_max: 0.10
      idle_arm:
        arm: "right"
        velocity_linear_mean_max: 0.05
      temporal:
        duration_min: 2.0
        duration_max: 8.0

  - id: "place_round_bread_in_bowl"
    match_subtask: "Place the round bread in pink bowl with left hand"
    constraints:
      workspace:
        left:
          min: [0.05, 0.00, -0.05]
          max: [0.25, 0.35, 0.20]
      velocity:
        linear:
          mean_max: 0.15
          std_max: 0.15
      idle_arm:
        arm: "right"
        velocity_linear_mean_max: 0.05
      temporal:
        duration_min: 1.0
        duration_max: 4.0

  - id: "End"
    match_subtask: "End"
    constraints:
      velocity:
        linear:
          mean_max: 0.12
          std_max: 0.12
      temporal:
        duration_max: 6.0
```

APPENDIX B
EXPERIMENT HYPERPARAMETERS

The hyperparameter settings for fine-tuning the evaluated VLA models are summarized in Table 3. All hyperparameters were selected based on the recommended settings from the original model papers, with adjustments made for optimal performance on the RoboCOIN dataset.

APPENDIX C
BOUNDARY EXPERIMENTS OF RTML FILTERING

To further illustrate the effectiveness of RTML filtering, we present boundary case examples. The boundary initial states are selected from trajectories that are close to the RTML constraints, representing challenging scenarios for trajectory quality assessment. As depicted in Figure 9, three representative boundary cases are considered:

- **Bread Rotated**. The bread is rotated at an extreme angle, making it difficult to grasp within the defined orientation tolerances.
- **Bowl at Edge**. The bowl is positioned at the edge of the workspace, challenging the robot's ability to reach and manipulate it without violating spatial constraints.
- **Bread Together**. The breads are placed very close together, requiring precise manipulation to avoid collisions while adhering to velocity and acceleration limits.

As shown in Table 4, even in these challenging cases, RTML filtering significantly improves model performance, with GR00T-Fine achieving a 35.0% success rate, and GR00T-Mine reaching 47.5%. While it sacrifices the coverage of edge scenarios, RTML effectively eliminates extreme cases and ensures the reliability of actions, thereby enhancing the robustness of the model.

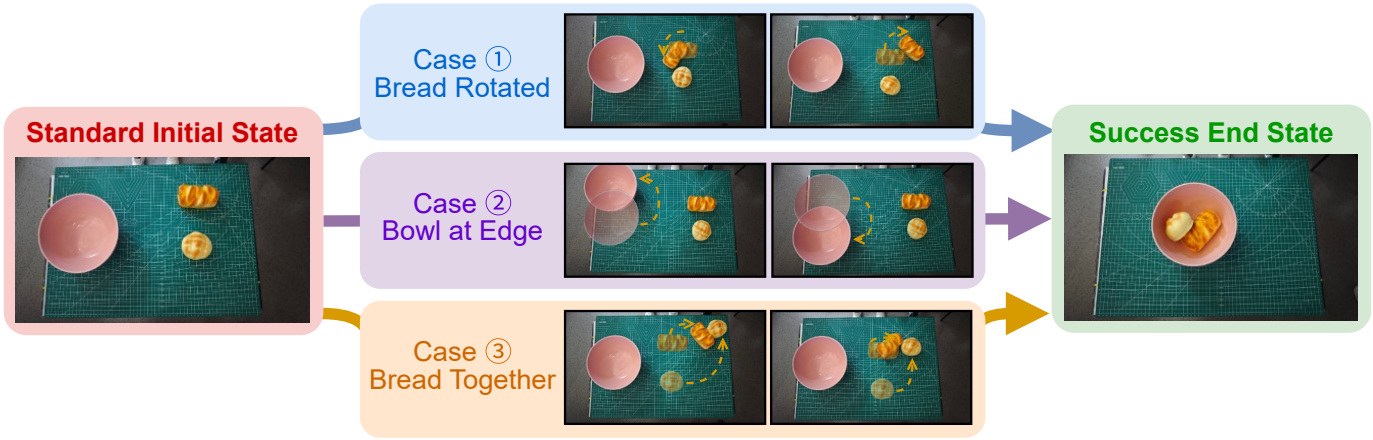

Fig. 9: Boundary cases of RTML filtering.

TABLE 3: Hyperparameter settings for fine-tuning the evaluated VLA models.

Hyperparameter	π_0	GR00T N1.5
Batch Size	32	32
Learning Rate	2.5e-5	1e-4
Optimizer	AdamW	AdamW
Weight Decay	0.01	0.01
Steps	30000	10000
Partial Fine-tuning	LoRA (r=16)	Diffusion & Projector

TABLE 4: RTML evaluation results for boundary cases.

Method	Success Rate
GR00T-Raw	27.5%
GR00T-Fine	35.0%
GR00T-Mine	**47.5%**