CALIBALL: OFFLINE EYE-TO-HAND CALIBRATION FOR HETEROGENEOUS ROBOT DATASETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-embodiment robot learning requires action representations with consistent semantics across robot platforms. Existing representations suffer from platform-specific inconsistencies, while current solutions often maintain embodiment-specific action heads or learn latent action spaces without explicitly establishing a shared geometric reference. Recent approaches adopt camera-centric motion representations to address this issue, but missing camera extrinsic annotations limit their coverage across existing datasets. We present CalibAll, an optimization-based, robot-independent pipeline for offline camera extrinsic annotation. CalibAll initializes extrinsics via temporal Perspective-n-Point (PnP) using a tracked end-effector point, then refines the estimate by aligning rendered and observed robot masks through differentiable rendering. It additionally derives camera-frame tool center point (TCP) poses from robot and gripper URDFs to support action unification. We apply CalibAll to 16 datasets spanning four robot platforms, producing approximately 97K calibrated episodes across single-arm and bimanual systems. Downstream experiments show that using the annotated dataset for cross-embodiment pretraining improves policy performance in both simulation and real-world tasks.

1 INTRODUCTION

Enabling robots to learn and generalize across different embodiments is a central challenge in scaling robot learning (Wang et al., 2024; Zha et al., 2026; Black et al., 2024). A core requirement for cross-embodiment learning is a unified action representation (Zheng et al., 2025b; Kim et al., 2024; Black et al., 2024) that carries consistent semantics regardless of the robot platform. However, this is far from trivial, as different robots have vastly different kinematic structures, joint configurations, and coordinate conventions.

Existing action representations each have significant limitations for cross-embodiment unification: joint angles are incompatible in dimensionality and semantics across robots, end-effector poses depend on the robot base frame and are often recorded at the flange rather than the more grasp-relevant tool center point (TCP) pose, and delta actions suffer from inconsistent local coordinate conventions. To circumvent these issues, prior work either employs embodiment-specific action heads (Bjorck et al., 2025; Team et al., 2024; Wang et al., 2024) that maintain separate output spaces per robot, or learns shared latent actions (Ye et al., 2024; Bu et al., 2025; Zheng et al., 2025a) that abstract away low-level differences. These approaches facilitate learning from heterogeneous data, but do not necessarily establish a shared geometric interpretation of actions across embodiments.

To establish a common geometric reference across embodiments, recent large-scale robot pretraining methods adopt **camera-centric motion representations**, expressing end-effector or semantic-anchor motion in the camera frame (Team et al., 2026; Yuan et al., 2026). For single-arm systems, this reduces dependence on robot-specific base frames; for bimanual systems, expressing both arms in the same camera frame captures their relative spatial configuration.

Applying these representations to existing robot datasets requires camera extrinsics to transform robot-base coordinates into camera coordinates. However, these annotations are missing from many robot manipulation datasets (O'Neill et al., 2024; Liu et al., 2024b; Wu et al., 2024), limiting the availability of camera-frame supervision. Recent pretraining methods account for this missing information: Qwen-RobotManip falls back to robot-base-relative actions when calibration is unavailable (Yuan et al., 2026), while UCAG-P masks camera-frame components that cannot be recovered without

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

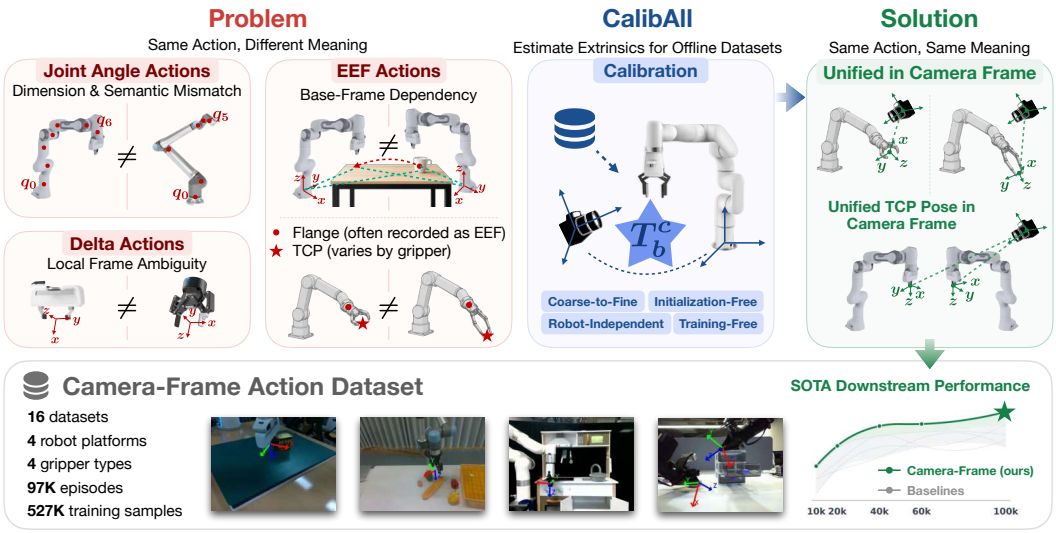

Figure 1: Robot datasets differ in action dimensionality, reference frames, and end-effector definitions, posing challenges for cross-embodiment learning. CalibAll estimates camera extrinsics for existing datasets and additionally provides camera-frame TCP poses to support action alignment. We annotate 16 datasets across four robot platforms and evaluate the utility of the estimated extrinsics through downstream policy learning in simulation and the real world.

metric depth or calibration (Team et al., 2026). Recovering camera extrinsics from existing datasets would therefore extend geometric supervision to data that these methods cannot fully exploit.

Yet existing calibration methods face practical limitations. Rendering-based approaches align rendered robot masks with observations but often require physical robot access or reliable initialization (Chen et al., 2023; Hong et al., 2024). Learning-based keypoint methods estimate camera poses via Perspective-n-Point (PnP), but rely on robot-specific annotations or training data (Lu et al., 2023; 2025). These requirements hinder scalable offline calibration across heterogeneous robot datasets.

To this end, we present CalibAll, an optimization-based, robot-independent pipeline for annotating robot demonstrations in existing datasets with camera extrinsics. CalibAll follows a coarse-to-fine calibration procedure. The coarse stage initializes camera extrinsics via temporal PnP using a tracked end-effector point. The refinement stage improves extrinsic accuracy by aligning rendered and observed robot masks through differentiable rendering. After calibration, CalibAll also derives camera-frame tool center point (TCP) poses from robot and gripper URDFs to support action unification.

Using CalibAll, we annotate 16 datasets spanning four robot platforms (Franka, UR5e, XArm7, and AgileX ALOHA), covering both single-arm and bimanual systems and yielding approximately 97K calibrated episodes. Each calibrated episode is associated with one third-person camera view and wrist-camera views are not counted separately. We further demonstrate that cross-embodiment pretraining on our annotated dataset improves downstream policy performance in both simulation and real-world tasks.

Our contributions are threefold:

- We introduce CalibAll, an optimization-based, robot-independent pipeline for offline camera extrinsic annotation, combining temporal PnP initialization with differentiable rendering-based refinement.
- We apply CalibAll to 16 datasets across four robot platforms, producing approximately 97K calibrated episodes with camera extrinsic annotations and additional camera-frame TCP poses to support cross-embodiment action unification.
- We evaluate the annotated dataset through downstream cross-embodiment pretraining and demonstrate improved policy performance in both simulation and real-world tasks.

2 Related Work

Extrinsic Calibration. Hand-eye calibration estimates the relative pose between the camera and the robot. In this work, we focus on the eye-to-hand setting, where an external camera observes the robot and the goal is to estimate the camera extrinsic with respect to the robot base. Traditional online methods rely on markers or require access to the physical robot during calibration (Ilonen & Kyrki, 2011; Tsai et al., 1989; Park & Martin, 1994; Daniilidis, 1999). Recent markerless methods such as EasyHeC (Chen et al., 2023) and EasyHeC++ (Hong et al., 2024) use differentiable rendering to align rendered robot masks with observed images, but they still require robot access and are sensitive to initialization. Offline methods such as DREAM (Lee et al., 2020), CtRNet (Lu et al., 2023), and CtRNet-X (Lu et al., 2025) estimate camera poses from collected images, but rely on robot-specific keypoint annotations or training data. In contrast, CalibAll targets offline datasets and estimates camera extrinsics in a training-free and robot-independent manner through a coarse-to-fine pipeline.

Cross-Embodiment Representations and Camera Geometry. Large robot policies learn from heterogeneous datasets whose action representations differ in joint dimensionality, coordinate frames, and end-effector definitions (Fu et al., 2024; Zhao et al., 2023; Intelligence et al., 2025a; Black et al., 2024; O'Neill et al., 2024; Zitkovich et al., 2023; Kim et al., 2024). Existing approaches accommodate these differences through embodiment-specific action heads (Bjorck et al., 2025; Team et al., 2024; Wang et al., 2024) or shared latent actions (Ye et al., 2024; Bu et al., 2025; Zheng et al., 2025a). A complementary line of work incorporates camera geometry to connect visual observations with robot motion, supporting human-to-robot learning, bimanual manipulation, and viewpoint-robust policies (Wang et al., 2023; Lepert et al., 2025; Fang et al., 2025; Jiang et al., 2025; Zhang et al., 2025). More recently, Qwen-RobotManip expresses end-effector motion deltas in camera coordinates (Yuan et al., 2026), while UCAG-P predicts camera-centric wrist and grasp-center motion before translating it into embodiment-specific controls (Team et al., 2026). However, geometric supervision depends on available calibration: Qwen-RobotManip falls back to robot-base-relative actions when calibration is missing, whereas UCAG-P masks unavailable geometric components. Our work complements these methods by recovering camera extrinsics from existing robot datasets, providing geometric annotations for downstream learning.

3 Method

We present CalibAll, an offline annotation pipeline for estimating camera extrinsics in existing robot manipulation datasets. The pipeline consists of three stages: preprocessing (section 3.2), camera extrinsic estimation (section 3.3), and postprocessing (section 3.4). We then apply CalibAll to annotate a collection of datasets spanning multiple robot platforms (section 3.5).

3.1 Problem Setup

Given an offline robot manipulation dataset, CalibAll estimates its camera extrinsics. We denote the rigid transformation from the robot base frame b to the camera frame c by T_b^c, satisfying $P_c = T_b^c P_b$. Here, P_b and P_c represent the same 3D point in homogeneous coordinates in the two frames, respectively.

To estimate this transformation, the pipeline uses two types of recorded data: (1) RGB video sequences, (2) Joint angles at each timestamp.

3.2 Preprocessing

Before running the calibration pipeline, several preprocessing steps are required.

Camera Intrinsic Estimation. Camera intrinsics K are needed for camera extrinsic estimation (section 3.3). When the dataset provides intrinsics directly, we use them as-is. Otherwise, we look up the intrinsics from the known camera model (e.g., RealSense D435, ZED 2). If neither is available, we estimate the intrinsics using off-the-shelf methods (Wang et al., 2025).

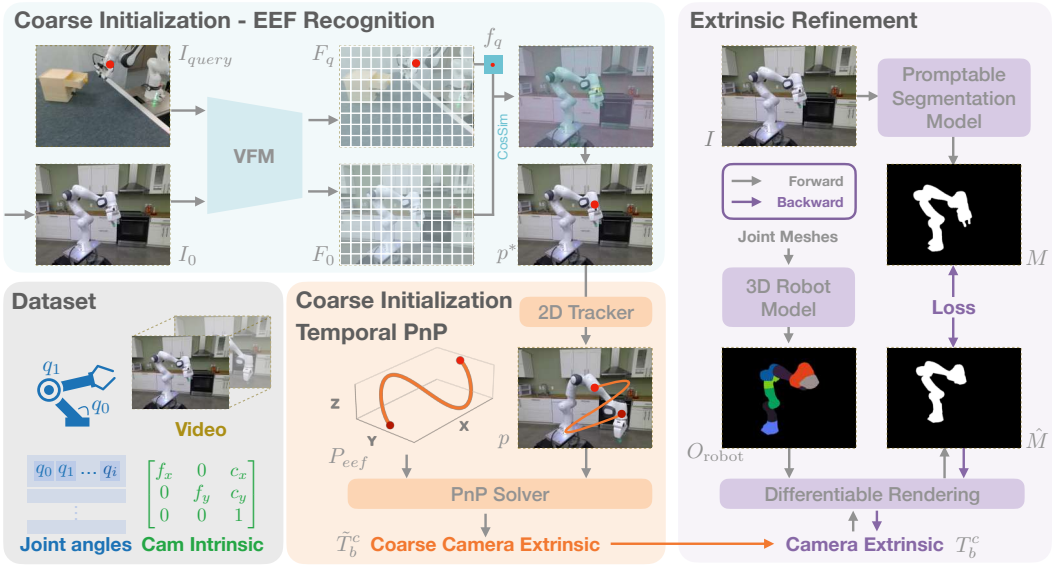

Figure 2: **Overview of CalibAll.** It follows a coarse-to-fine calibration pipeline that achieves training-free, stable, and accurate camera extrinsic estimation across diverse datasets and robot platforms. CalibAll first use EEF Recognition to obtain the end-effector tracking point, then apply temporal PnP to estimate a coarse extrinsic, and finally perform extrinsic refinement to obtain an accurate result.

Video Clip Selection. The coarse initialization stage (section 3.3.1) relies on tracking the end-effector throughout the video. If the gripper leaves the camera field of view during the sequence, the tracker may lose the target and produce erroneous trajectories. To ensure tracking quality, we manually select a video clip for each scene in which the gripper remains fully visible. This clip is used for calibration, while the resulting extrinsic applies to the entire scene since the camera-robot relative pose is fixed.

Bimanual Robot Handling. For bimanual robots where both arms are defined in a single URDF with a fixed spatial relationship, we only need to calibrate one arm; the extrinsic of the other arm is automatically determined by the known inter-arm transform. For setups where the two arms are independently mounted and their relative position is not known, we calibrate each arm separately by estimating an independent extrinsic $T_{b_i}^c$ for each arm base b_i. In both cases, since both arms are observed by the same camera, the camera frame naturally encodes their spatial relationship.

3.3 Camera Extrinsic Estimation

We estimate the camera extrinsic T_b^c through a coarse-to-fine approach: a temporal PnP stage produces a stable initial estimate, which is then refined via differentiable rendering-based optimization.

3.3.1 Coarse Initialization

EEF Recognition. The coarse initialization stage requires a tracking point p^* on the end-effector (EEF). Although this point can be manually annotated, we explore an automatic alternative using cross-image correspondence from a vision foundation model (VFM) (Tang et al., 2023; Meng et al., 2024). Given a reference mark p_q on an EEF image, we extract its feature and find the most similar location in the first frame I_0:

$$p^* = \arg\max_{(u,v)} \text{Cos}\Big(f_q, F_0(u,v)\Big), \tag{1}$$

where f_q and F_0 are VFM features of the reference mark and the target frame, respectively. Since EEFs share common visual characteristics across robots, this enables stable localization across diverse platforms with a single mark. Representative localization examples are shown in fig. 4.

Temporal PnP. Using a 2D point tracking model (Karaev et al., 2024; 2025) with the mark p^* as input, we track the EEF throughout the video to obtain a 2D trajectory p. Meanwhile, we compute the corresponding 3D EEF trajectory P_{eef} via forward kinematics from the joint angles Q and the robot model O. An PnP solver (Lepetit et al., 2009) then estimates the camera extrinsic:

$$\tilde{T}_b^c = \mathrm{PnP}\left(p, P_{eef}, K\right). \tag{2}$$

Since p^* may not precisely correspond to the 2D projection of P_{eef} and the tracked trajectory contains inevitable noise, the result is a coarse approximation. Nevertheless, it reliably captures the camera position and viewing direction, serving as an effective initialization for the subsequent refinement. We provide qualitative visualizations in fig. 5 and quantitative ablations in table 2.

3.3.2 Rendering-based Refinement

We refine the coarse estimate via rendering-based optimization that exploits the full robot geometry. The key idea is to iteratively adjust the camera pose until the rendered robot mask aligns with the observed one, following the render-and-compare paradigm (Kundu et al., 2018).

We employ a promptable segmentation foundation model (Carion et al., 2025; Liu et al., 2024a; Ren et al., 2024; Ravi et al., 2024) to obtain the robot arm mask M for each image. We then adopt a differentiable rendering pipeline (Laine et al., 2020) to render the corresponding robot mask $\hat{M}$ under the current camera pose:

$$\hat{M} = \min\left(1, \sum_l \pi\left(\boldsymbol{T_b^c}\, T_l^b l\right)\right), \tag{3}$$

where π is the differentiable mask renderer, l denotes a robot link, and T_l^b is the link-to-base transformation computed via forward kinematics. Masks are rendered per link and aggregated, with the min operator ensuring valid binary occupancy.

The camera extrinsic $\boldsymbol{T}_b^c$ is optimized via gradient descent:

$$L(T_b^c) = \left\| M - \hat{M} \right\|_2 = \left\| M - \min\left(1, \sum_l \pi\left(\boldsymbol{T_b^c}\, T_l^b l\right)\right) \right\|_2. \tag{4}$$

Rendering-based optimization is prone to local minima with poor initialization (table 3). If the initial camera pose fails to capture the robot, the rendered mask collapses to zero, producing no gradients and halting optimization. The coarse initialization from section 3.3.1 is therefore essential for stable convergence (table 2, table 3).

3.4 Postprocessing

Beyond camera extrinsics, CalibAll provides additional annotations through the following postprocessing steps.

Standardized Camera-Frame Action Conversion. To support cross-embodiment learning, we additionally provide camera-frame TCP poses that reduce differences in reference frames and end-effector definitions across datasets.

This conversion consists of three steps: (1) Convert the raw flange EEF pose into a tool center point (TCP) pose. Since different grippers have different geometries, the flange pose does not necessarily correspond to the actual grasping point. We define the action position as the TCP, located at the tip of the closed gripper fingers, and obtain the flange-to-TCP offset $T_{\mathrm{tcp}}^{\mathrm{flange}}$ from the gripper URDF. (2) Standardize the TCP local frame. We introduce a gripper-specific transform S_{tcp} such that the z-axis points along the gripper forward direction and the y-axis points along the gripper closing direction. This makes the TCP rotation semantics consistent across robot platforms. (3) Transform the standardized TCP pose from the robot base frame to the camera frame:

$$T_{\mathrm{tcp}}^c = T_b^c\, T_{\mathrm{flange}}^b\, T_{\mathrm{tcp}}^{\mathrm{flange}}\, S_{\mathrm{tcp}}, \tag{5}$$

where T_b^c is the camera extrinsic, T_{flange}^b is the flange pose in the robot base frame computed by forward kinematics, $T_{\mathrm{tcp}}^{\mathrm{flange}}$ is the flange-to-TCP transform obtained from the gripper URDF, and

S_{tcp} is the gripper-specific transform that standardizes the TCP local frame. For bimanual robots, we apply the same conversion independently to each arm using its own extrinsic $T_{b_i}^c$, so that both arms are represented in the shared camera frame.

Unified Action Dimension. To enable joint training of single-arm and bimanual data, we use a common 14-dimensional action layout. Each arm contributes a 7-dimensional camera-frame action (3D position + 3D rotation + 1D gripper). For bimanual robots, the first seven dimensions correspond to the arm appearing on the left side of the reference image, and the remaining seven correspond to the arm on the right. For single-arm robots, the action occupies the first seven dimensions, with the remaining dimensions padded with zeros. This layout allows single-arm and bimanual data to be mixed in a single training batch without architectural changes.

Auxiliary Annotations. Beyond camera extrinsics and camera-frame actions, CalibAll also produces auxiliary annotations, including masks and bounding boxes for both the robot and gripper, 3D TCP trajectories in the camera frame, and corresponding 2D TCP trajectories in the image space as shown in section F. These annotations are generated using the estimated camera extrinsics and robot URDF. Specifically, the URDF provides the robot geometry and kinematic structure, while T_b^c aligns the robot model with the camera frame. We render the robot in the image space to obtain masks and bounding boxes, and project the 3D TCP trajectory onto the image plane to obtain 2D trajectories. These annotations can support downstream tasks such as visual grounding (Liu et al., 2024a), robot-aware policy learning (Feng et al., 2025), and trajectory-conditioned manipulation (Tan et al., 2026).

3.5 Large-scale Dataset Construction

We apply CalibAll to annotate camera extrinsics in a collection of existing robot manipulation datasets. Although CalibAll reduces the need for robot-specific training data, fully automatic annotation remains challenging in practice. Visual tracking may fail under severe self-occlusion, and foundation-model masks can be inaccurate in cluttered scenes or under challenging viewpoints. To improve annotation quality, we adopt a human-in-the-loop protocol: when tracking fails, we manually provide suitable initial tracking points; when mask prediction is inaccurate, we manually select the correct mask. Fully automatic calibration across these challenging conditions remains future work.

As shown in table 5, the annotated data come from three major sources: RoboMIND (Wu et al., 2024), RDT (Liu et al., 2024b), and Open X-Embodiment (OXE) (O'Neill et al., 2024). They cover 16 datasets, four robot platforms, and four gripper types, including both single-arm robots (Franka, UR5e, and XArm7) and bimanual robots (AgileX ALOHA). In total, we obtain approximately 97K calibrated episodes and 527K frame-level training samples. Each calibrated episode is associated with one third-person camera view; wrist-camera views are not counted separately. Each sample includes the estimated camera extrinsic T_b^c, along with derived TCP poses and auxiliary annotations described in section 3.4.

4 Experiments

4.1 Calibration Experiments

We first evaluate the calibration accuracy of CalibAll by comparing with existing offline methods.

Experimental Setup. We evaluate camera extrinsic estimation on the DREAM dataset (Lee et al., 2020), which contains approximately 57K real-world images of the Franka Panda robot. We compare CalibAll with existing offline calibration methods, including DREAM (Lee et al., 2020) variants (F, Q, H), RoboPose (Labbé et al., 2021), CtRNet (Lu et al., 2023), and CtRNet-X (Lu et al., 2025). We report Average Distance (ADD) and Area-Under-the-Curve (AUC). Given 3D points $\{p_i\}_{i=1}^n$ in the robot base frame, we compute their alignment error under the predicted extrinsic $\tilde{T}_b^c$ and ground-truth extrinsic T_b^c:

$$D_i = \left\| \tilde{T}_b^c p_i - T_b^c p_i \right\|_2, \quad \text{ADD} = \frac{1}{n}\sum_{i=1}^{n} D_i, \quad \text{AUC} = \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n} \mathbf{1}(D_i < \tau_t), \tag{6}$$

Table 1: Calibration Results on DREAM (Lee et al., 2020) and comparison with offline calibration methods.

	AUC ↑	ADD (m) ↓	Train-Free	Init-Free	Robot in-and-out	Robot-Independent
DREAM-F (Lee et al., 2020)	60.740	113.029	✗	✓	✗	✗
DREAM-Q (Lee et al., 2020)	56.988	59.284	✗	✓	✗	✗
DREAM-H (Lee et al., 2020)	68.584	17.477	✗	✓	✗	✗
RoboPose (Labbé et al., 2021)	80.094	0.020	✗	✗	✗	✗
CtRNet (Lu et al., 2023)	85.962	0.020	✗	✓	✗	✗
CtRNet-X (Lu et al., 2025)	86.231	0.014	✗	✓	✓	✗
CalibAll (ours)	**97.642**	**0.008**	✓	✓	✓	✓

where τ_t is the t-th distance threshold and T is the number of thresholds.

Results. As shown in table 1, CalibAll achieves an AUC of 97.642 and ADD of 0.008 m, substantially outperforming all baselines. The best prior method, CtRNet-X, reaches 86.231 AUC and 0.014 m ADD. CalibAll improves the AUC by over 11 points while being the only method that is simultaneously training-free, initialization-free, and robot-independent.

Ablation study. We ablate the two stages of the camera extrinsic estimation pipeline on the DREAM dataset (table 2). Using coarse initialization alone achieves an AUC of 82.050 and ADD of 0.071 m, indicating that temporal PnP already provides a reasonably good estimate. Using rendering-based refinement alone with random initialization fails completely (AUC of 0), confirming that the optimization is prone to local minima without a good starting point. The full pipeline (coarse + refine) achieves the best result, demonstrating that both stages are essential and complementary.

Table 2: Ablation study on DREAM (Lee et al., 2020).

	AUC ↑	ADD (m) ↓
Coarse only	82.050	0.071
Fine only	0	0.5+
Coarse + Fine	**97.642**	**0.008**

Convergence analysis of refinement. To further investigate the convergence behavior of the refinement stage, we conduct a controlled experiment (table 3). We select ground-truth camera extrinsics, add varying levels of rotational and positional noise, and run the refinement from these noisy initializations. For each noise level, we conduct 60 trials and record the number of successful convergences. As shown in table 3, when the initialization error is within approximately 10° rotation and 7.5 cm translation, the refinement converges reliably (green region). Beyond this range, the success rate drops sharply (red region). Notably, our coarse initialization stage produces estimates with an average rotation error of 4.0° and position error of 5.5 cm on the DREAM dataset, both well within the convergence region, explaining why the full pipeline achieves stable results.

Table 3: Convergence regions of extrinsic refinement.

Rot (°)	**Pos Noise (cm)**				
	0.1–2.5	2.5–5	5–7.5	7.5–10	10–15
1–5	60/60	60/60	56/60	41/60	19/60
5–10	60/60	58/60	49/60	25/60	16/60
10–15	54/60	47/60	38/60	24/60	16/60
15–20	39/60	36/60	16/60	18/60	11/60
20–25	28/60	21/60	14/60	19/60	6/60

Visualization results. We test CalibAll on DROID (Khazatsky et al., 2024) in section D, a dataset with provided camera extrinsics, and find that CalibAll provides more precise calibration results than the original dataset annotations, yielding better alignment between the rendered robot model and the observed image. We also visualize the EEF recognition results in fig. 4, as well as the rendered robot models obtained using the camera extrinsics estimated by the coarse and refinement stages as shown in fig. 5. We additionally visualize 10 representative cases from our Dataset as shown in section F, including projected 2D TCP trajectories, standardized local frames, and rendered bounding boxes and masks. These examples demonstrate that CalibAll produces accurate and scalable annotations across diverse robot platforms, camera viewpoints, and manipulation scenes.

Table 4: Success rate (%) on RobotTwin (Chen et al., 2025) Clean.

Pretrain Type	10k	20k	40k	60k	80k	100k	Best
From Scratch	12.8	23.5	32.9	33.0	31.2	42.4	42.4
Raw Mixed	19.4	26.6	37.7	33.3	**33.6**	36.1	37.7
Prompt	23.7	32.3	37.7	39.4	29.4	38.2	39.4
Action Head	19.1	22.8	28.5	36.8	31.4	33.0	36.8
Camera-Frame (ours)	**24.7**	**33.4**	**41.9**	**42.8**	29.7	**48.0**	**48.0**

4.2 Simulation Experiments

To evaluate the downstream utility of our annotated dataset, we use the derived TCP actions for cross-embodiment pretraining and evaluate the resulting policies in simulation and real-world tasks.

Pretraining Variants. We compare five variants. **From Scratch** uses only SFT on the target benchmark without cross-embodiment pretraining. **Raw Mix** directly mixes datasets with their original action representations, padded to the same dimensionality. **Prompt** uses the same action representations as Raw Mix and additionally specifies robot embodiment information in the language prompt, as detailed in section A. **Action Head** uses dataset-specific heads to encode and decode each dataset's action representation. **Camera-Frame (ours)** uses the TCP actions derived from CalibAll annotations.

Experimental Setup. We use StarVLA (Community, 2026) as the training framework. All variants are evaluated on the RoboTwin benchmark. We use the RoboTwin (Chen et al., 2025) as benchmark, which contains 2,500 trajectories for training. All baselines are pretrained on the original datasets, whereas **Camera-Frame** is pretrained on our Dataset.

Camera-frame pretraining achieves the best performance. As shown in table 4, **Camera-Frame** achieves the highest success rate in 5 out of 6 SFT checkpoints, reaching the best overall performance of 48.0% at 100k steps. This improves over From Scratch by 5.6%, Raw Mix by 10.3%, Prompt by 8.6%, and Action Head by 11.2% in terms of best success rate.

Naive cross-embodiment pretraining can lead to negative transfer. Raw Mix performs worse than From Scratch at convergence, with 36.1% versus 42.4% at 100k steps. This suggests that directly mixing data from different robots without action representation unification can introduce negative transfer, since the model must fit inconsistent action semantics under a shared action space. Action Head also underperforms From Scratch in terms of best success rate, achieving only 36.8%, indicating that using dataset-specific heads alone does not fully resolve the underlying inconsistency. Prompt improves over Raw Mix and Action Head, reaching 39.4% best success rate, but still falls behind From Scratch and Camera-Frame, suggesting that embodiment information in the language prompt is insufficient to eliminate action-space misalignment.

Camera-frame pretraining maintains strong performance across SFT steps. Camera-Frame consistently performs well throughout the SFT process, achieving the best results at 10k, 20k, 40k, 60k, and 100k steps. This stable advantage suggests that the unified camera-frame action representation provides a more transferable pretraining signal across embodiments. In particular, Camera-Frame already reaches 24.7% at 10k steps and continues to improve to 48.0% at 100k steps, indicating both faster adaptation and stronger final performance.

4.3 Real-Robot Experiments

Experimental Setup. We evaluate real-world closed-loop performance in a few-shot tabletop setting on an ALOHA platform. The policy observes one third-person camera, two wrist cameras, the task prompt, and proprioceptive states. We compare four variants: From Scratch, Raw Mix, Camera-Frame, and $\pi_{0.5}$ (Intelligence et al., 2025b) as a reference pretrained policy.

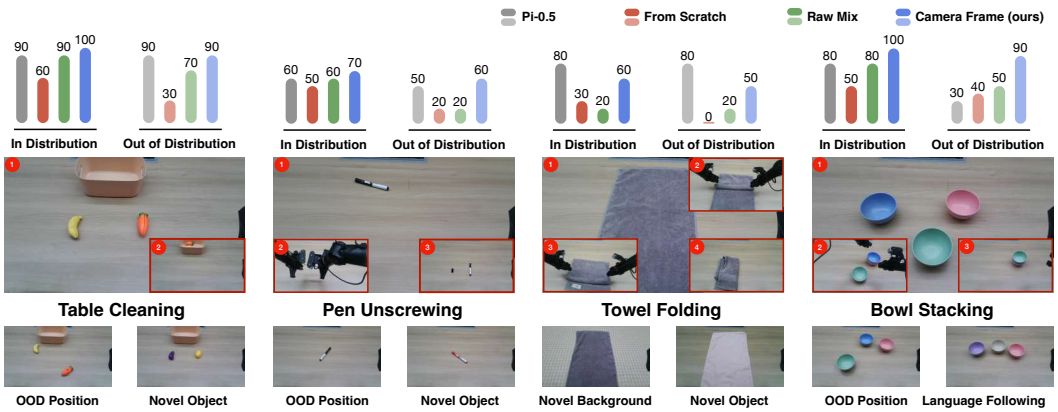

Figure 3: **Real-World Experiments.** The top charts report results from 10 in-distribution (ID) trials and 10 out-of-distribution (OOD) trials. The bottom panel illustrates the OOD test setup.

We evaluate four tasks: Table Cleaning, Pen Unscrewing, Towel Folding, and Bowl Stacking as shown in fig. 3. Each task includes both in-distribution and OOD settings. The OOD tests introduce novel objects, unseen object positions, different object instances or backgrounds, and longer-horizon continuous manipulation. For Bowl Stacking, the robot must additionally follow the color order specified by the language instruction.

Results. We report real-robot results under both ID and OOD settings. Across all tasks, Camera-Frame achieves the highest average success rate of 77.5%, outperforming $\pi_{0.5}$ (70.0%), Raw Mix (51.3%), and From Scratch (35.0%). This advantage is also pronounced under OOD conditions, where Camera-Frame reaches 72.5% average success, compared with 62.5% for $\pi_{0.5}$, 40.0% for Raw Mix, and 22.5% for From Scratch. These results show that camera-frame action pretraining provides a more transferable action prior and improves real-world generalization compared with naive cross-dataset action mixing. The gains are especially clear in spatially grounded and language-conditioned tasks. For Bowl Stacking, Camera-Frame achieves 100% success in ID and 90% in OOD, substantially outperforming Raw Mix. Although $\pi_{0.5}$ performs better on Towel Folding, Camera-Frame still consistently surpasses From Scratch and Raw Mix. Overall, these results show that Camera-Frame provides a robust and generalizable pretraining strategy for real-world manipulation.

The results demonstrate the value of our annotated dataset for cross-embodiment pretraining, with improved downstream policy performance in both simulation and real-world tasks.

5 Conclusion

We present CalibAll, an optimization-based, robot-independent pipeline for estimating camera extrinsics from offline robot datasets. The pipeline combines temporal PnP initialization with differentiable rendering-based refinement and additionally provides TCP poses to support downstream learning. We apply CalibAll to 16 datasets across four robot platforms, producing approximately 97K calibrated episodes. Experiments demonstrate the utility of these annotations for cross-embodiment pretraining, with improved policy performance in both simulation and real-world tasks.

AI Use Statement

We used generative AI tools to provide feedback on experimental methodology, assist with interpreting reported results, check consistency between tables and text, and improve manuscript wording. The authors take responsibility for the final content of the paper.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Linghao Chen, Yuzhe Qin, Xiaowei Zhou, and Hao Su. Easyhec: Accurate and automatic hand-eye calibration via differentiable rendering and space exploration. *RA-L*, 2023.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Konstantinos Daniilidis. Hand-eye calibration using dual quaternions. *IJRR*, 1999.

Hongjie Fang, Chenxi Wang, Yiming Wang, Jingjing Chen, Shangning Xia, Jun Lv, Zihao He, Xiyan Yi, Yunhan Guo, Xinyu Zhan, et al. Airexo-2: Scaling up generalizable robotic imitation learning with low-cost exoskeletons. *arXiv preprint arXiv:2503.03081*, 2025.

Yao Feng, Hengkai Tan, Xinyi Mao, Guodong Liu, Shuhe Huang, Chendong Xiang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist bimanual manipulation. *arXiv preprint arXiv:2507.12898*, 2025.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Zhengdong Hong, Kangfu Zheng, and Linghao Chen. Easyhec++: Fully automatic hand-eye calibration with pretrained image models. In *IROS*, 2024.

Jarmo Ilonen and Ville Kyrki. Robust robot-camera calibration. In *ICAR*, 2011.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025a.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. a vision-language-action model with open-world generalization. *0.5: a vision-language-action model with open-world generalization*, 2025b.

Tianchong Jiang, Jingtian Ji, Xiangshan Tan, Jiading Fang, Anand Bhattad, Vitor Guizilini, and Matthew R Walter. Do you know where your camera is? view-invariant policy learning with camera conditioning. *arXiv preprint arXiv:2510.02268*, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *ICCV*, 2025.

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.

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.

Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018.

Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Single-view robot pose and joint angle estimation via render & compare. In *CVPR*, 2021.

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ToG*, 2020.

Timothy E Lee, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Oliver Kroemer, Dieter Fox, and Stan Birchfield. Camera-to-robot pose estimation from a single image. In *ICRA*, 2020.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025.

Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *IJCV*, 2009.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024a.

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*, 2024b.

Jingpei Lu, Florian Richter, and Michael C Yip. Markerless camera-to-robot pose estimation via self-supervised sim-to-real transfer. In *CVPR*, 2023.

Jingpei Lu, Zekai Liang, Tristin Xie, Florian Richter, Shan Lin, Sainan Liu, and Michael C Yip. Ctrnet-x: Camera-to-robot pose estimation in real-world conditions using a single camera. In *ICRA*, 2025.

Lingchen Meng, Shiyi Lan, Hengduo Li, Jose M Alvarez, Zuxuan Wu, and Yu-Gang Jiang. Segic: Unleashing the emergent correspondence for in-context segmentation. In *ECCV*, 2024.

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

Frank C Park and Bryan J Martin. Robot sensor calibration: solving ax= xb on the euclidean group. *T-RO*, 1994.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024.

Huajie Tan, Enshen Zhou, Zhiyu Li, Yijie Xu, Yuheng Ji, Xiansheng Chen, Cheng Chi, Pengwei Wang, Huizhu Jia, Yulong Ao, et al. Robobrain 2.5: Depth in sight, time in mind. *arXiv preprint arXiv:2601.14352*, 2026.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *NeurIPS*, 2023.

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.

Xiaomi Embodied Intelligence Team, Shaoqing Xu, Fang Li, Guozhi Zhan, Zhixiang Duan, Yuhan Wang, Yuechen Luo, Shengyin Jiang, Hanbing Li, Zhiying Du, et al. One policy, many embodiments: Unified camera-centric action geometry pre-training for heterogeneous embodied manipulation. *arXiv preprint arXiv:2608.26058*, 2026.

Roger Y Tsai, Reimar K Lenz, et al. A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration. *T-RO*, 1989.

Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Advances in neural information processing systems*, 37:124420–124450, 2024.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026.

Lihan Zha, Asher J Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z Ren, and Anirudha Majumdar. Lap: Language-action pre-training enables zero-shot cross-embodiment transfer. *arXiv preprint arXiv:2602.10556*, 2026.

Tianyi Zhang, Haonan Duan, Haoran Hao, Yu Qiao, Jifeng Dai, and Zhi Hou. Grounding actions in camera space: Observation-centric vision-language-action policy. *arXiv preprint arXiv:2508.13103*, 2025.

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.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025a.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025b.

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

A Training Details

We use StarVLA (Community, 2026) as the policy training framework for all variants. Cross-embodiment pretraining is conducted for 200K steps on 32 A100 GPUs. For both simulation and real-robot experiments, we fine-tune all variants on the target benchmark for 100K SFT steps on 8 A100 GPUs, using the same training protocol, batch size, and optimization hyperparameters for a fair comparison.

For the **Action Head baseline**, we use dataset-specific action heads in both the action encoder and decoder. Specifically, each dataset has an independent action head for mapping its original action representation into the latent action space, as well as a corresponding decoder head for mapping the latent representation back to the dataset-specific action space.

For the **Prompt baseline**, we specify the robot embodiment information in the language instruction while keeping the original action representations unchanged. Specifically, we use the following prompt template:

```
The robot is [robot name] with [single/dual] arm.  Please
predict the actions to execute the following task:  [task].
```

B Dataset Overview

Table 5: Summary of annotated datasets.

#	Source	Dataset	Robot Platform	# Third-Person Cameras	# Samples
1	RoboMIND	ur_1rgb	UR5e + Robotiq	1	25,002
2	RoboMIND	franka_3rgb	Franka	3	50,637
3	RoboMIND	agilex_3rgb	AgileX ALOHA	1	9,298
4	RDT	rdt_aloha	AgileX ALOHA	1	6,061
5	OXE	nyu_franka_play	Franka	2	730
6	OXE	berkeley_autolab_ur5e	UR5e + Robotiq	1	1,000
7	OXE	mutex	Franka	1	1,500
8	OXE	toto	Franka	1	901
9	OXE	qut_dexterous	Franka	1	812
10	OXE	franka-play	Franka	1	456
11	OXE	mimic_play	Franka	1	378
12	OXE	sailor	Franka	1	250
13	OXE	kaist_nonprehensile	Franka	1	201
14	OXE	ucsd_kitchen	XArm7	1	150
15	OXE	utokyo_xarm	XArm7	1	92
16	OXE	BUDS	Franka	1	50
				Total	97,518

Table 5 summarizes the datasets annotated by CalibAll. The annotated data are collected from three major sources, including RoboMIND, RDT, and Open X-Embodiment (OXE), and cover 16 datasets across four robot platforms: Franka, UR5e, XArm7, and AgileX ALOHA. For each dataset, we report the data source, dataset name, robot platform, number of available third-person cameras, and number of annotated episodes. In total, our annotation covers 97,518 episodes, providing camera extrinsics and auxiliary annotations for both single-arm and bimanual robot datasets.

C Visualization of the CalibAll Pipeline

We visualize the key stages of CalibAll from three aspects. First, we show the EEF recognition results across different datasets and robot platforms in fig. 4. The VFM similarity heatmaps and the selected tracking points show that a single annotated reference mark can reliably localize EEF tracking points across diverse robots.

Next, we visualize the extrinsics estimated by the coarse stage on offline datasets with different robot platforms, as shown in fig. 5. Although the rendered masks still exhibit small deviations from the

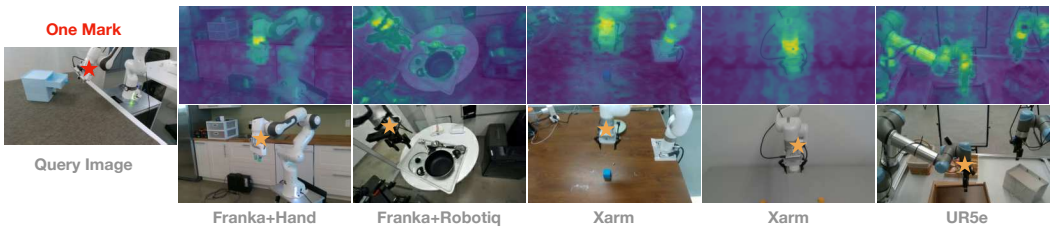

Figure 4: **Qualitative result of EEF recognition on Franka, xArm and UR5e.** The first row shows the heatmaps obtained from feature matching. The second row visualizes the selected tracking point based on the maximum similarity.

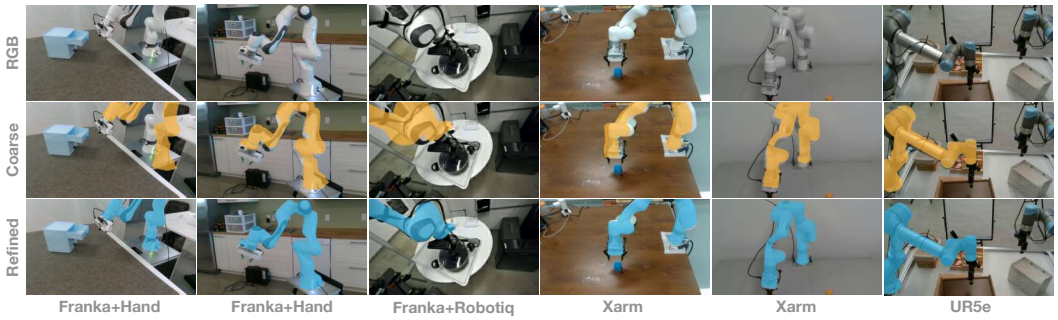

Figure 5: **Qualitative result of coarse initialization and extrinsic refinement on Franka, xArm and UR5e.** The first row presents the source RGB images. The second row shows the rendered result using the camera extrinsic obtained from the automatic coarse initialization approach. The last row shows the rendered result of final camera extrinsic produced by CalibAll.

observed robot masks, they already capture the camera viewpoint and robot location reasonably well, providing an effective initialization for refinement.

Finally, as shown qualitatively in fig. 5, the refined extrinsics produce accurate robot-image alignment across different datasets and robot platforms. Together, these results validate the design of CalibAll: the coarse stage provides a stable initialization, while the refinement stage improves the estimate to achieve accurate offline camera extrinsic calibration.

D Calibration Experiment on DROID Dataset

We conducted experiments on the more complex DROID (Khazatsky et al., 2024) dataset. DROID is a diverse robot manipulation dataset containing 76k demonstration trajectories, or 350 hours of interaction data, collected across 564 scenes and 86 tasks. In addition, DROID records camera extrinsics for each scene obtained via traditional marker-based calibration (Tsai et al., 1989) during data collection, which solves the equation $AX = XB$.

From this dataset, we selected 20 scenes for testing. We report the success rates for each stage across these 20 scenes, as shown in the table 6. EEF Recognition is highly robust across all 20 scenarios, consistently achieving good initial tracking points. However, due to the complexity of the scenes and the movements of robot, tracking failed in 5 out of the 20 scenarios, causing CalibAll to fail. This observation indicates that tracking performance is the main bottleneck of the current pipeline. Nevertheless, for the 15 successful tracking scenarios, 14 achieved accurate camera extrinsic predictions. We further evaluated these 14 scenarios and found that the average position error was only 4.4 cm, with a rotation error of 3.9°. These results demonstrate the stability and effectiveness of CalibAll under complex conditions.

We further visualized the masks rendered using the ground truth camera extrinsics recorded in the DROID dataset and the masks rendered using the camera extrinsics estimated by CalibAll in fig. 6. The ground truth in DROID sometimes exhibits errors, whereas the results estimated by CalibAll

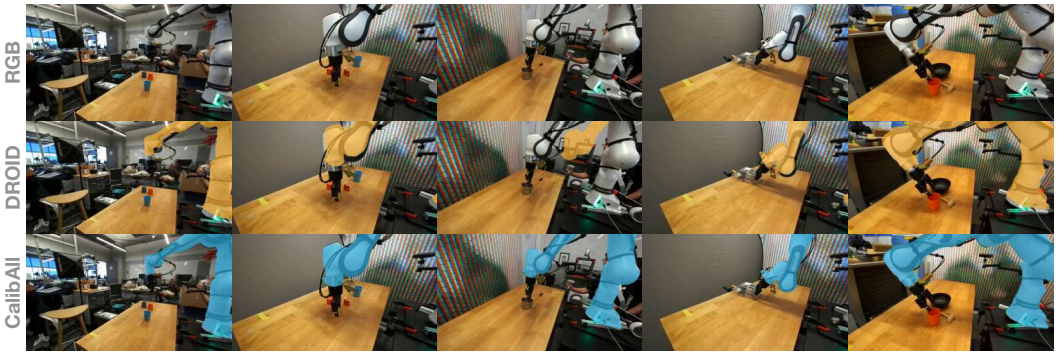

Figure 6: **Qualitative results on DROID (Khazatsky et al., 2024).** Rendering result using camera extrinsics in DROID (Khazatsky et al., 2024) obtained via traditional marker-based calibration during data collection and those estimated training-free and offline by CalibAll (ours).

Table 6: **Results on DROID (Khazatsky et al., 2024).** We evaluate CalibAll on 20 sequences. Point denotes the success rate of end-effector mark recognition, Track denotes the success rate of obtaining valid 2D trajectories, and Refine denotes the success rate of camera extrinsic refinement. E_r and E_p report the mean rotation and translation errors over the successful trials.

Success Rate			Refinement Error	
Point	Track	Refine	E_r (°)	E_p (cm)
20/20	15/20	14/20	3.9	4.4

are actually more accurate and convenient than those from the DROID dataset collected using the traditional method.

E Real-World Experiment Results

Table 7: Real-world evaluation results on four tasks. We report success rates (%) under both in-distribution (ID) and out-of-distribution (OOD) settings.

Task	Setting	Pi-0.5	From Scratch	Raw Mix	Camera Frame (ours)
Table Cleaning	ID	90	60	90	**100**
	OOD	90	30	70	**90**
Pen Unscrewing	ID	60	50	60	**70**
	OOD	50	20	20	**60**
Towel Folding	ID	**80**	30	20	60
	OOD	**80**	0	20	50
Bowl Stacking	ID	80	50	80	**100**
	OOD	30	40	50	**90**
Average	ID	77.5	47.5	62.5	**82.5**
	OOD	62.5	22.5	40.0	**72.5**
Overall Average	–	70.0	35.0	51.3	**77.5**

table 7 reports the detailed real-world evaluation results on four tabletop tasks: Table Cleaning, Pen Unscrewing, Towel Folding, and Bowl Stacking. For each task, we evaluate both in-distribution (ID) and out-of-distribution (OOD) settings, and report the success rate over 10 trials. The OOD settings introduce variations such as novel objects, unseen object positions, different object instances or backgrounds, and longer-horizon continuous manipulation.

Camera Frame achieves the best overall average success rate of 77.5%, outperforming Pi-0.5, From Scratch, and Raw Mix. The advantage is especially clear under OOD settings, where Camera Frame reaches 72.5% average success, compared with 62.5% for Pi-0.5, 40.0% for Raw Mix, and 22.5% for From Scratch. These results show that camera-frame action pretraining improves real-world generalization and provides a more transferable action representation across tasks.

F VISUALIZATION OF CAMERA-FRAME ACTION DATASET

We visualize representative annotations produced by CalibAll. Given the estimated camera extrinsic and robot URDF, CalibAll renders robot masks and bounding boxes in the image space, and projects the 3D TCP trajectory into the camera view to obtain the corresponding 2D trajectory. These visualizations show that our annotations are well aligned with the observed robot across diverse platforms, camera viewpoints, and manipulation scenes, demonstrating the scalability of the Camera-Frame Action Dataset.

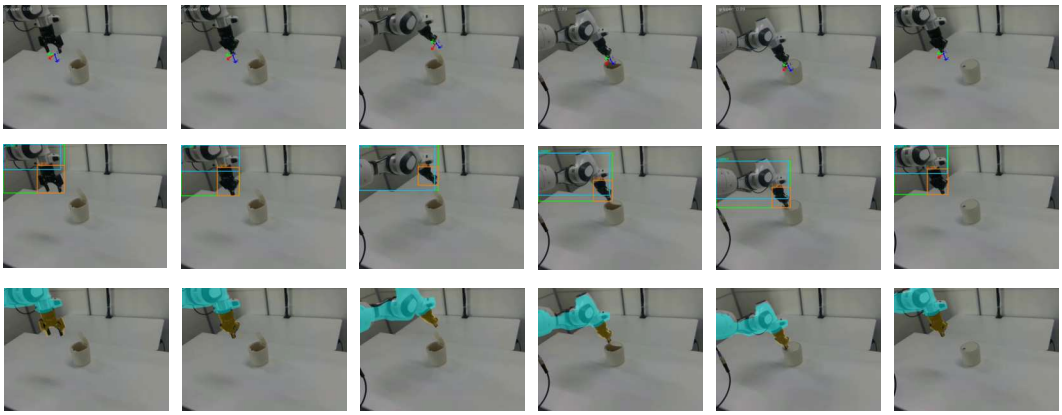

Figure 7: Data Annotation on **RoboMIND Franka**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.

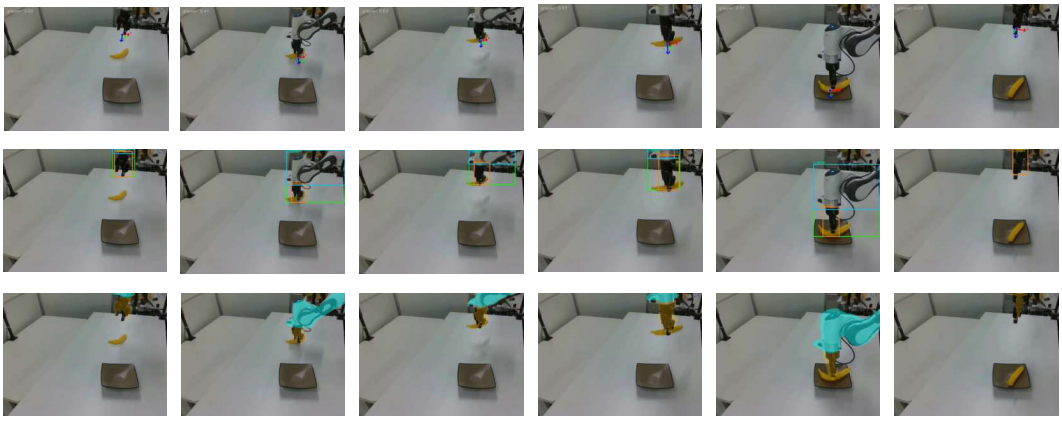

Figure 8: Data Annotation on **RoboMIND Franka**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.

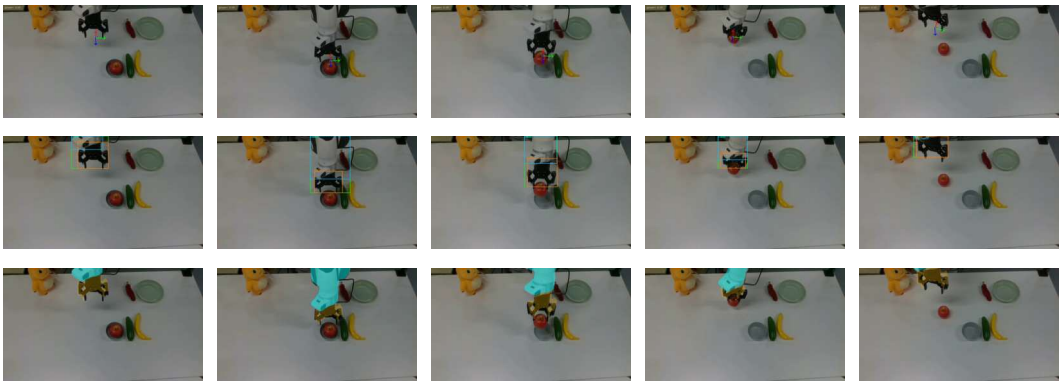

Figure 9: Data Annotation on **RoboMIND Franka**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.

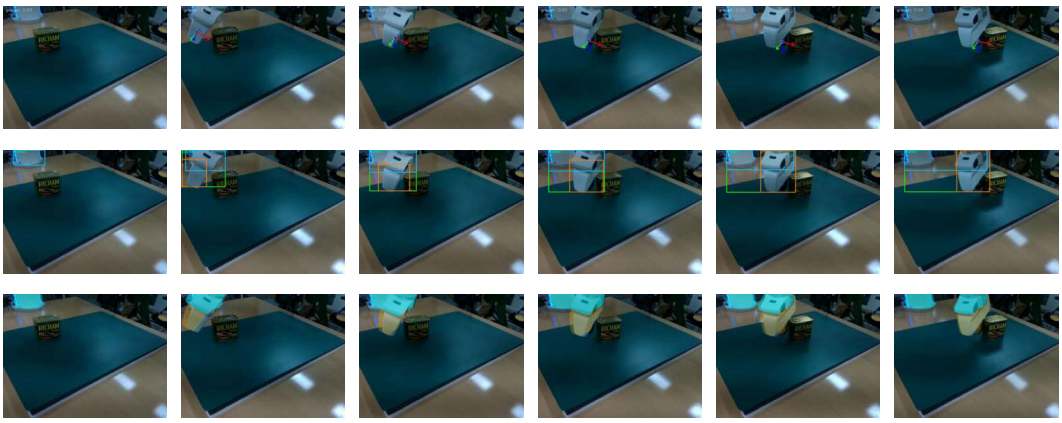

Figure 10: Data Annotation on **KAIST**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.

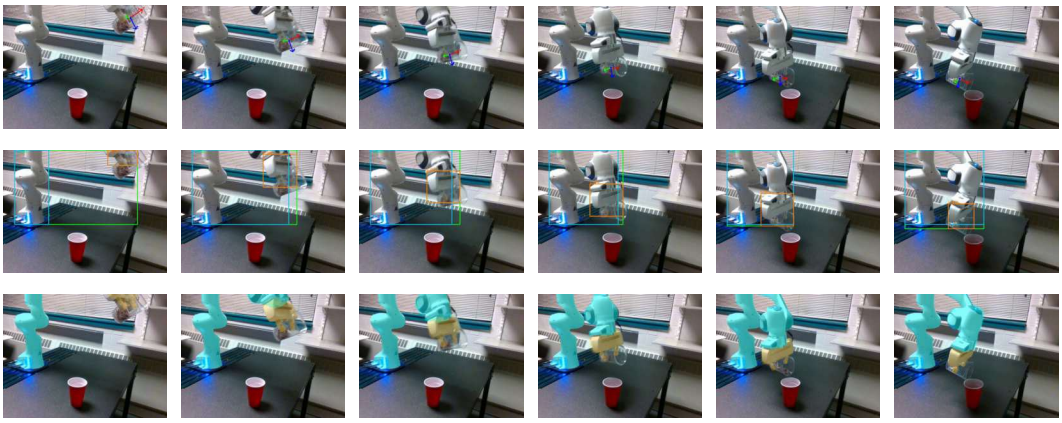

Figure 11: Data Annotation on **TOTO**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.

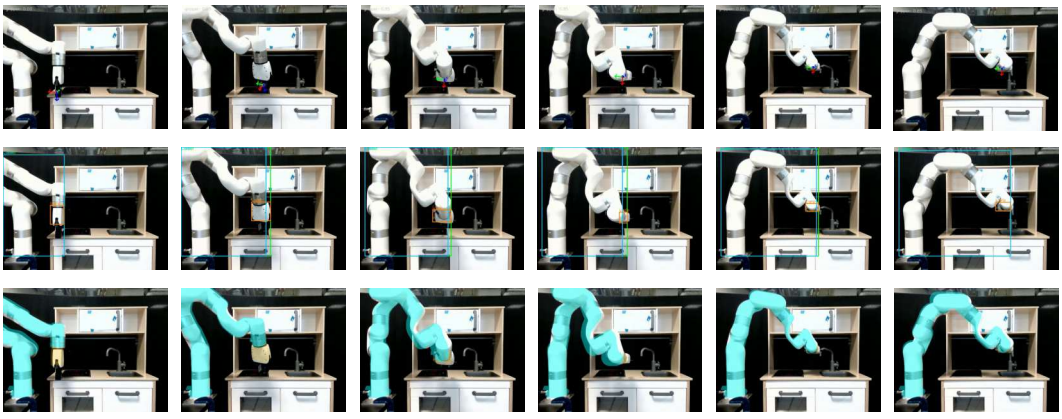

Figure 12: Data Annotation on **UCSD Kitchen**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.

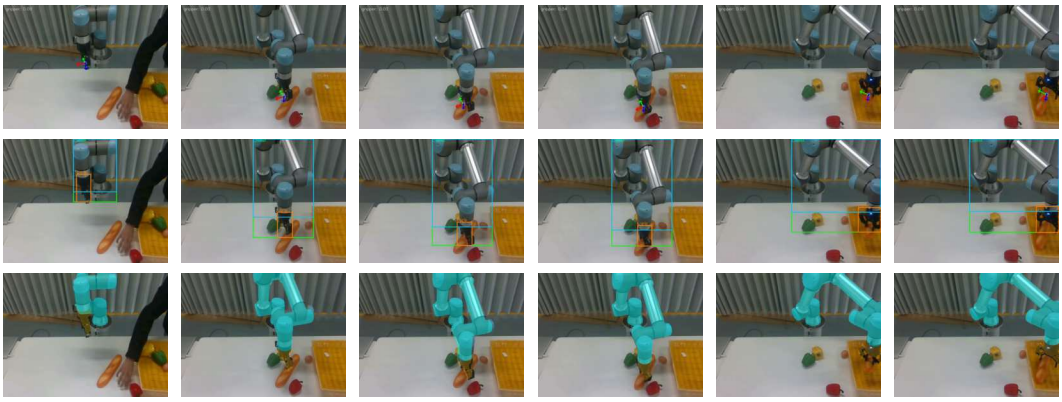

Figure 13: Data Annotation on **RoboMIND UR5e**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.

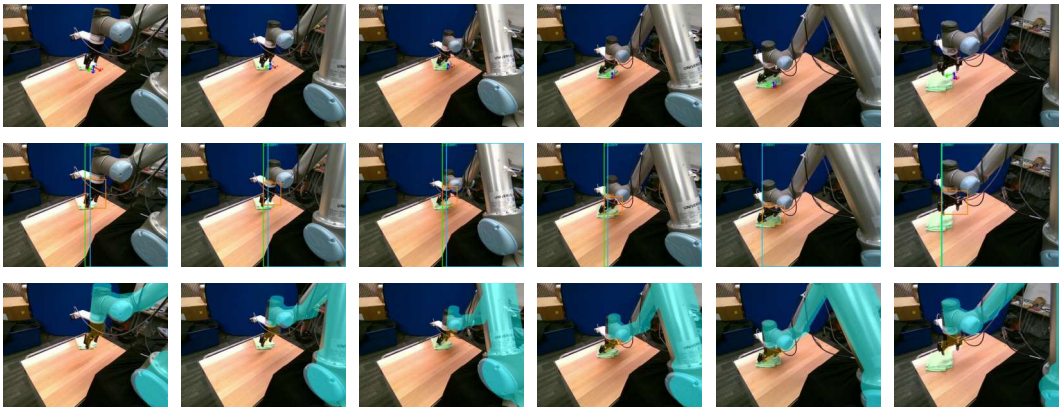

Figure 14: Data Annotation on **Berkeley UR5**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.

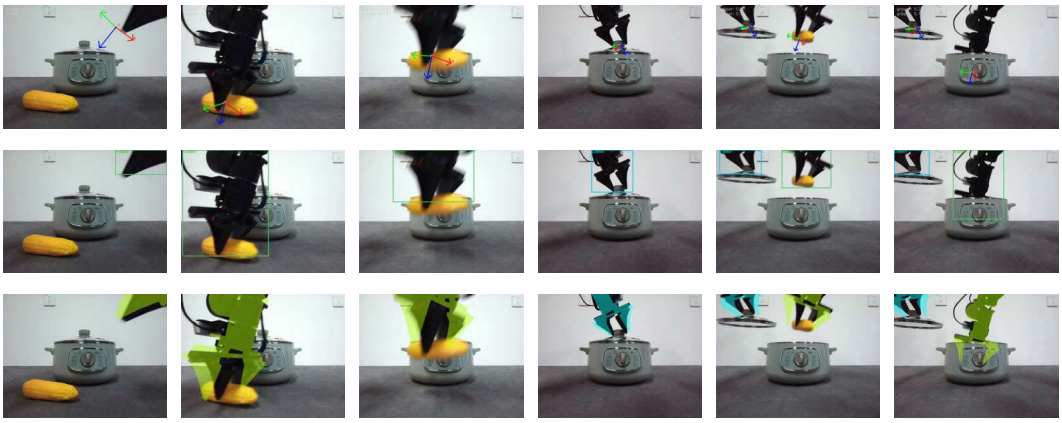

Figure 15: Data Annotation on **RoboMIND ALOHA**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.



Figure 16: Data Annotation on **RDT**. We visualize annotations generated by CalibAll: the first row shows 2D/3D end-effector trajectories with local coordinate axes, where red, green, and blue denote the local x, y, and z axes, respectively; the second row shows gripper and robot-arm bounding boxes; and the third row shows the corresponding segmentation masks.