BREE: A LOW-COST, FULLY FABRICABLE AND REPRODUCIBLE PLATFORM FOR REAL-WORLD BIMANUAL ROBOTIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling data has been a key driver of recent advances in embodied intelligence, yet real-world robotic data remain fragmented across heterogeneous embodiments and physical platform realizations. Even nominally identical hardware can produce different observation and action distributions due to variations in calibration, sensing, workspace geometry, and task environments. This makes independently collected data difficult to aggregate and reuse. Here we introduce *BREE*, a low-cost (∼$600) and fully reproducible bimanual robotic platform built on dual SO-101 arms. BREE standardizes the embodiment, sensing configuration, workspace, and calibration process, supported by *BREE-Assets* and *BREE-Data*. *BREE-Assets* provides standardized and fully fabricable task objects, fixtures, environmental components, including a dedicated physical calibrator for cross-instance alignment. *BREE-Data* provides eight standardized bimanual manipulation tasks with demonstrations and evaluation protocols based on BREE-Assets. Extensive cross-instance evaluations demonstrate that policies trained on data from one BREE instance can be directly transferred to another, while aggregating data collected from multiple instances can improve policy performance. That is, BREE exhibits *cross-instance transfer* and *multi-instance scaling* properties, providing a practical path toward scalable real-world robot learning across distributed platforms. We will open-source the complete BREE stack.

1 INTRODUCTION

Recent advances in embodied intelligence have increasingly benefited from scaling real-world robotic data. Large-scale datasets and generalist robot policies, such as Open X-Embodiment O'Neill et al. (2024), DROID Khazatsky et al. (2024), and Octo Ghosh et al. (2024), have shown that broader and more diverse robotic experience improves policy transfer and generalization. However, unlike image or text data that can be readily aggregated from distributed sources, robotic data are tightly coupled to the physical platforms on which they are collected, including their embodiments, sensing configurations, and action spaces. Consequently, data independently collected across robotic platforms are often difficult to directly aggregate and reuse, making real-world data scaling substantially more challenging. This makes cross-platform data compatibility a key requirement for scaling real-world robot learning.

This incompatibility stems from both embodiment heterogeneity and differences in physical platform realization. Across robotic systems, variations in kinematics, joint configurations, end-effectors, sensing layouts, and action representations naturally lead to mismatched observation and action spaces. More importantly, even nominally identical hardware can produce different data distributions after independent assembly, as small deviations in robot placement, camera pose, joint calibration, workspace geometry, lighting, backgrounds, and task-object configurations can alter the mapping from observations to actions Khargonkar et al. (2024); Huang et al. (2026). This issue becomes more pronounced in bimanual manipulation, where coordinated behaviors depend critically on the relative geometry between the two arms and their shared workspace. Therefore, hardware identity alone is insufficient for cross-platform data compatibility, and reproducible robot learning requires standardization of the complete physical setup.

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.

Table 1: **Reproducibility and multi-instance capabilities of real-world robotic manipulation benchmarks.** *Cross-Instance Eval.* denotes evaluation across physical instances, and *Multi-Instance Aggregation* denotes joint training with data from multiple instances.

Benchmark	Dual Arm	Cost	Reprod. Testbed	Fabricable Task Assets	Platform Calibration	Cross-Instance Eval.	Multi-Instance Aggregation
REPLAB	✗	∼$2K	✓	✗	✓	✗	✓
RAMP	✗	–	✗	✓	✗	✗	✗
FurnitureBench	✗	–	✓	✓	✓	✓	✗
SceneReplica	✗	–	✓	✗	✓	✗	✗
FMB	✗	–	✓	✓	△	✗	✗
VLA-REPLICA	✗	∼$1.05K	✓	✗	✓	✓	✗
ArmnetBench	✓	∼$477	△	✗	✗	✗	✗
BREE	✓	**∼$600**	✓	✓	✓	✓	✓

△ denotes partial support; – denotes unreported cost.

Existing real-world benchmarks have made certain progress toward accessible and reproducible robot evaluation. REPLAB Yang et al. (2019) explored low-cost reproducible manipulation, while RAMP Collins et al. (2024) and FurnitureBench Heo et al. (2023) introduced more standardized task and environment settings. SceneReplica Khargonkar et al. (2024) and VLA-REPLICA Huang et al. (2026) further investigated scene reconstruction and cross-platform evaluation consistency through visual alignment and calibration. However, as summarized in Table 1, existing platforms typically address only partly the requirements for scalable real-world data collection. Highly standardized platforms often incur substantial hardware and setup costs, while low-cost alternatives may still lack strict physical reproducibility. More importantly, most existing efforts focus on reproducible evaluation, whereas cross-instance data interoperability remains insufficiently studied. It is still unclear whether data independently collected from different physical realizations can be directly transferred and aggregated for joint training. This leaves a critical gap between reproducible real-world platforms and scalable real-world data collection.

To bridge this gap, we introduce BREE, a low-cost and fully reproducible real-world platform for bimanual robotic manipulation. BREE is designed around the principle that independently constructed platforms should generate compatible data rather than merely support comparable evaluation. Toward this goal, BREE standardizes the robotic embodiment, sensing configuration, workspace, physical assets, and calibration process, with a dedicated physical calibrator to reduce cross-instance geometric variation. Contributions are summarized as follows.

- We introduce *BREE*, a low-cost and fully reproducible bimanual robotic platform built on dual SO-101 arms. BREE standardizes the embodiment, sensing configuration, workspace, and calibration process.
- As part of BREE, we provide *BREE-Assets*, a standardized and fully fabricable collection of physical components for real-world manipulation, including task objects, fixtures, and environmental elements with explicit geometries and specifications. In addition, BREE-Assets includes a calibration aid for maintaining consistent joint-coordinate mappings across robot arm instances.
- Based on BREE-Assets, we construct *BREE-Data*, a standardized real-world dataset and benchmark comprising eight bimanual manipulation tasks of varying practical difficulty, with demonstrations and evaluation protocols for policy training and assessment.
- We demonstrate BREE's *cross-instance transfer* and *multi-instance scaling* properties. That is, policies trained on data from one BREE instance can be directly evaluated on another, while jointly training on data aggregated from multiple instances improves performance in general, validating scalable real-world robot learning across distributed platforms.

2 Related Work

Robotic Manipulation Benchmarks. Robotic manipulation benchmarks such as ManiSkill2 Gu et al. (2023) and CALVIN Mees et al. (2022) define tasks and experimental conditions for policy learning and evaluation. LIBERO Liu et al. (2023) structures knowledge-transfer evaluation around changes in spatial relationships, objects, and task goals. RoboTwin Chen et al. (2026) en-

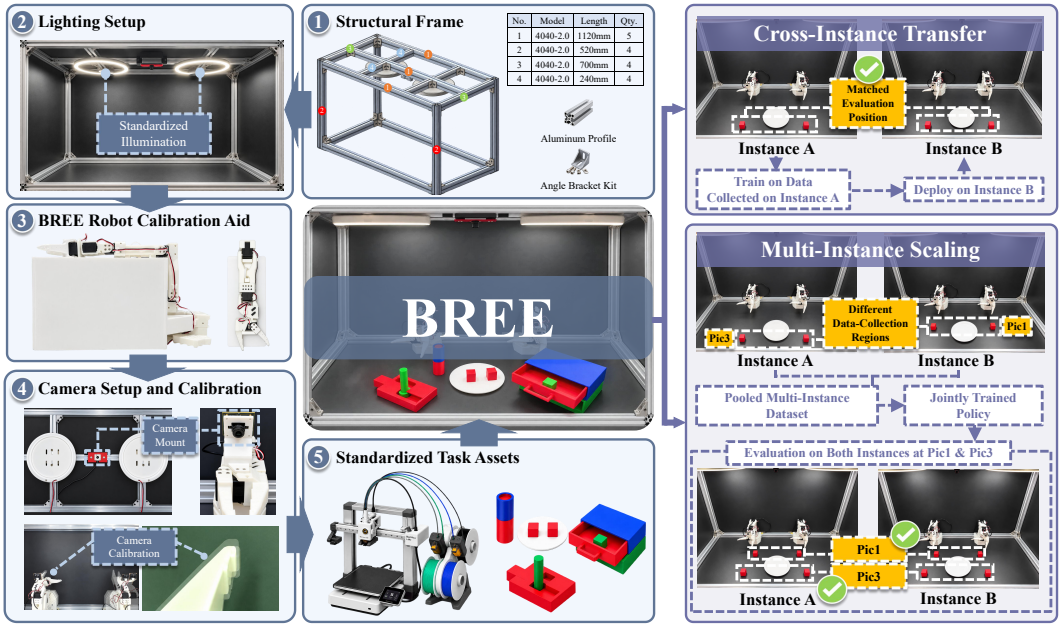


Figure 1: **Overview of BREE.** BREE standardizes the physical platform, robot and camera calibration, and task assets to reduce variations across independently assembled bimanual systems. This cross-instance consistency enables policies and data to be reused across BREE instances, supporting direct policy transfer and scalable multi-instance data collection.

ables large-scale demonstration generation for bimanual manipulation in simulation. Simulation benchmarks allow policies to be trained and evaluated under shared task and environment configurations. Reproducing these conditions across physical setups is more challenging, which can limit data compatibility and the comparability of evaluation results.

Real-World Robotic Manipulation Benchmarks. Real-world manipulation benchmarks aim to provide consistent physical conditions for evaluating robot learning methods. Early platforms focused on standardizing the robot and workspace while reducing the cost of real-world evaluation. For example, REPLAB Yang et al. (2019) uses a low-cost standardized robot cell with dedicated calibration procedures. As manipulation tasks became more diverse, benchmark design also began to standardize the task environment and physical assets. RAMP Collins et al. (2024) provides fabricable components for robotic assembly, while FurnitureBench Heo et al. (2023) and FMB Luo et al. (2025) define standardized environments and task objects for more complex manipulation. A further challenge is reproducing these physical conditions across independently constructed setups. SceneReplica Khargonkar et al. (2024) addresses this problem through replicable real-world scenes, while VLA-REPLICA Huang et al. (2026) uses platform calibration to support evaluation across physical instances. The availability of multiple low-cost setups also enables parallel real-world evaluation, as explored by ArmnetBench Selvaraj et al. (2026). As summarized in Table 1, existing benchmarks provide different levels of testbed reproducibility, task-asset standardization, platform calibration, and cross-instance evaluation. The aggregation of demonstrations collected independently across physical instances, however, has received considerably less attention.

3 The BREE Framework

BREE standardizes the physical setup, calibration, task assets, data collection, and evaluation to reduce cross-instance variation across independently assembled bimanual robotic systems. Fig. 1 outlines the two components, BREE-Assets and BREE-Data, and the workflow from platform construction to data collection and evaluation. BREE-Assets specifies the structural components, calibration procedures, and task assets for platform construction. BREE-Data defines the manipulation tasks, demonstration collection procedures, and evaluation protocols. These specifications establish a common basis for cross-instance transfer and multi-instance scaling.

Table 2: **Approximate Bill of Materials for the BREE platform.**

Component	Specification	Qty.	Total cost ($)
SO-101 arms	Leader–Follower pairs, STS3215 servos, 7.4 V (1/345, 1/191, and 1/147 gearing)	2	450.00
Global camera	DCXW100, 2.1 mm, 100°	1	9.00
Wrist cameras	DCXW100, 2.8 mm, 90°	2	18.00
Structural frame	4040-2.0 aluminum profiles, connectors, mounting hardware, clamps, and workspace panels	1 set	96.14
Lighting system	LED strips, power supply, controller, and wiring	1 set	28.80
3D-printed parts	Task objects, fixtures, and physical calibrator	1 set	22.40
Total (USD)			**624.34**

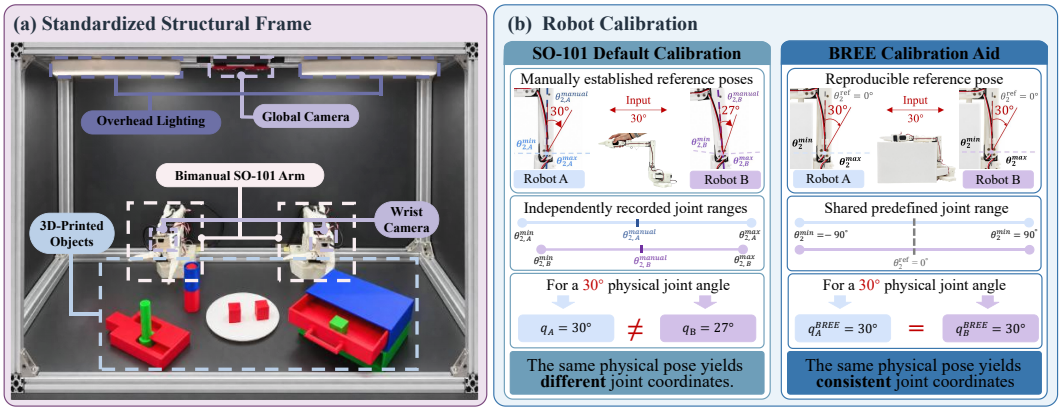


Figure 2: **BREE-Assets: Structural Frame and Calibration.** (a) Standardized structural frame for reproducible platform construction. (b) BREE Calibration Aid for consistent joint-coordinate mappings across robot instances.

3.1 BREE-Assets

BREE-Assets defines the standardized physical assets and construction specifications from platform geometry to task objects; a BREE instance is one physical realization built accordingly. The structural frame establishes a common mounting reference, on which robot and camera calibration align joint-coordinate mappings and visual observations across instances. Lighting and imaging settings reduce variation during image acquisition. Shared 3D-printable models complete the task setup by specifying consistent shapes and dimensions for objects and fixtures.

Structural Frame. As shown in Fig. 2(a), a rigid aluminum-profile frame establishes the nominal geometry of BREE through predefined mounting locations for two SO-101 arms, one global camera, two wrist-mounted cameras, overhead lighting, and a fixed workspace. Its dimensions and mounting interfaces constrain component placement and establish a common physical reference for robot and camera calibration. The rigid structure allows the platform to be relocated as a single unit while limiting configuration changes. To reproduce this geometry across independently assembled instances, we specify the frame dimensions, materials, and mounting coordinates, with assembly instructions and fabrication files to be released alongside the benchmark. Table 2 lists the main platform components and their approximate costs, totaling approximately $600.

Robot Calibration. The standard SO-101 calibration relies on manually established reference poses and independently recorded joint ranges, which can lead to inconsistent joint-coordinate mappings across robot instances, as illustrated in Fig. 2(b). BREE instead uses the BREE Calibration Aid, a physical fixture that establishes a reproducible reference pose $\boldsymbol{\theta}^{\mathrm{ref}}$. Instance-specific homing offsets align the calibrated encoder positions at this reference:

$$P_{j,r}(\theta_j^{\mathrm{ref}}) = c_j, \qquad \forall r, \tag{1}$$

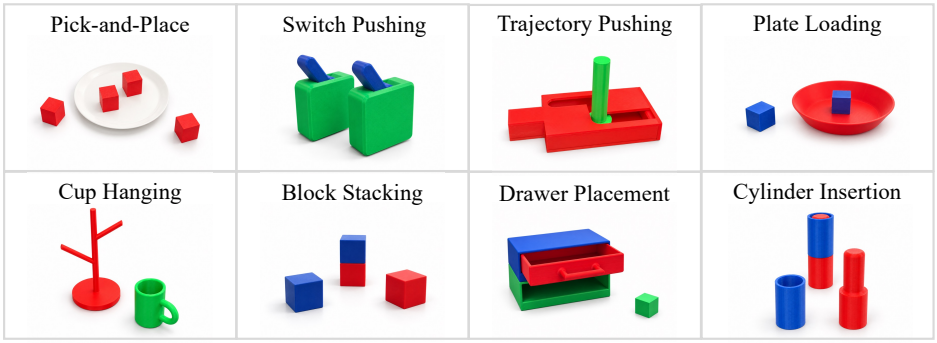


Figure 3: **BREE-Assets: Task Assets.** Standardized objects and fixtures are provided for all eight tasks to maintain consistent task setups across independently assembled platforms.

where $P_{j,r}$ is the homing-offset-corrected encoder position of joint j on robot instance r, and c_j is its shared reference position. For each revolute joint, BREE converts the calibrated encoder position into a joint-angle representation as

$$q_{j,r}^{\mathrm{BREE}} = \left(P_{j,r} - \frac{P_j^{\min} + P_j^{\max}}{2} \right) \frac{360^\circ}{4095}, \tag{2}$$

where $P_j^{\min}$ and $P_j^{\max}$ are predefined calibration limits shared across BREE instances. Under consistent joint directions and encoder scales, the shared reference positions and calibration ranges establish a common joint-coordinate definition across instances, while allowing homing offsets to compensate for installation-dependent encoder alignment. Detailed joint ranges, homing-offset computation, asymmetric wrist-roll calibration, gripper calibration, and offset-limit handling are provided in Appendix A.1.

Camera Installation and Calibration. BREE reduces camera variation across physical instances by addressing camera geometry, imaging conditions, and acquisition stability. Camera geometry is reproduced using predefined visual references, with the workspace and robot configuration for the global camera and the gripper for each wrist camera. Given consistent camera geometry, wrist-camera imaging settings are adjusted to reduce differences caused by illumination and reflections. The acquisition settings are then configured to reduce dynamic variations during data collection, including exposure-induced changes in frame rate. See Appendix A.2 for detailed procedures and parameters.

Fabricable Task Assets. Consistent task definitions alone do not guarantee identical physical objects across independently reproduced benchmark setups. BREE provides shared 3D-printable models for the objects and fixtures used in all eight tasks, allowing task assets with the same geometry and dimensions to be fabricated locally. This avoids dependence on specific commercial objects and makes the physical task setup reproducible across BREE instances. The complete task suite is illustrated in Fig. 3.

3.2 BREE-Data

Building on the standardized physical setup defined by BREE-Assets, BREE-Data establishes a common protocol for demonstration collection and policy evaluation across instances. The task suite specifies the manipulation behaviors, initial conditions, and success criteria. Within each task, demonstrations are collected using shared observation modalities, state and action representations, and sampling procedures. Policies trained on these demonstrations are evaluated under matched task configurations across instances. Combined with the physical specifications of BREE-Assets, this protocol enables systematic investigation of cross-instance transfer and multi-instance scaling.

Task Design. Bimanual manipulation involves different coordination relationships between the two arms, which can place distinct demands on policy learning and execution. We design eight tasks to represent three common coordination patterns, including synchronized, asymmetric, and

Table 3: **Policy performance under hardware changes between training and test environments, with 10 trials per test.** For robot-arm experiments, the specified calibration is used consistently during both demonstration collection and evaluation.

Calibration	Training Env.	Evaluation Env.	Purpose	Success (%)
Robot Arm				
BREE	BREE Inst. A	BREE Inst. A (Arm A)	Baseline	100
	BREE Inst. A	BREE Inst. A (Arm B)	Arm replacement	90
	BREE Inst. A	BREE Inst. B (Arm A)	Platform replacement	100
Standard SO-101	BREE Inst. A	BREE Inst. A (Arm A)	Baseline	70
	BREE Inst. A	BREE Inst. A (Arm B)	Arm replacement	0
Global Camera				
None	BREE Inst. A	BREE Inst. A (Camera A)	No camera calibration	0
BREE	BREE Inst. A	BREE Inst. A (Camera rotation)	Camera perturbation	100
	BREE Inst. A	BREE Inst. A (Camera B)	Camera replacement	100

sequential behaviors. Across these patterns, the tasks cover object transfer, constrained motion, and precise placement within a shared workspace. Each task is defined by fixed objects, fixtures, initial configurations, and success criteria, with its physical assets standardized through BREE-Assets.

Cross-Instance Data Collection. Scaling data collection across physical instances requires the collection process to remain consistent when the physical instance changes. BREE standardizes the task instructions, observation modalities, robot state and action representations, and sampling procedure to define a common data interface across instances. BREE-Assets complements this interface through standardized calibration. The resulting collection protocol allows each BREE instance to generate demonstrations independently using the same task and policy interfaces.

Standardized Evaluation Protocol. BREE evaluates each task using a predefined set of object configurations drawn from the demonstration distribution. A set of reference scene images is provided to reproduce these configurations across independently assembled BREE instances. This protocol allows policies to be evaluated under matched task conditions across physical instances.

4 Experiment

We conduct real-world experiments to examine three aspects of BREE. (1) **Asset reproducibility (Sec. 4.2):** Can BREE provide reproducible policy evaluation under changes to physical assets and environmental conditions? (2) **Cross-instance reuse and scaling (Sec. 4.3):** Does this reproducibility enable policies and demonstrations to be reused and aggregated across physical instances? (3) **Benchmark evaluation (Sec. 4.4):** How do representative robot learning policies perform under the BREE evaluation protocol?

4.1 Experimental Setup

BREE Instances and Tasks. Experiments are conducted on two independently assembled BREE instances, denoted BREE Instance A and BREE Instance B, using tasks selected from the eight manipulation tasks in Section 3.2. BREE Instance A serves as the default evaluation instance unless otherwise specified. For compactness in tables, "Inst." denotes "Instance."

Benchmark Policies. We evaluate five representative policies representing transformer-based imitation learning, diffusion-based policies, and vision-language-action models, including ACT Zhao et al. (2023), SmolVLA Shukor et al. (2025), RDT Liu et al. (2025), X-VLA Zheng et al. (2026), and $\pi_{0.5}$ Black et al. (2025). The full benchmark evaluates all five policies using 50 demonstrations per task collected at 30 FPS, while ACT serves as the primary policy in the preceding experiments. Training and inference details are provided in Appendix C.

Evaluation Protocol. For the full benchmark, each task is evaluated at five predefined object configurations. For each configuration, the policy is evaluated with four inference seeds (0, 1, 2, and

Table 4: **Policy performance under controlled lighting variations.**

Internal Lighting	External Lighting	Enclosure	Trials	Success (%)
White	Enhanced	Installed	10	100
White	Dark	Installed	10	90
Yellow	Ambient	Installed	10	30
Off	Ambient	Removed	10	0



Figure 4: **Lighting conditions used to evaluate the controlled workspace.**

3), resulting in 20 rollouts per policy. We report two evaluation metrics, i.e., **full-task success** and **stage-wise success**. Full-task success measures whether the rollout completes the entire task within the execution budget. Stage-wise success is assessed at rollout termination and averaged across predefined stages, allowing completed stages to be superseded by valid subsequent stages. Detailed completion criteria, execution budgets, and invalid-trial handling are provided in Appendix B.2.

4.2 BREE-Assets Reproducibility

We study the reproducibility of BREE-Assets under hardware replacement and controlled environmental variations. ACT is used as the evaluation policy throughout these experiments.

Robot Arm Calibration. BREE calibration supports consistent policy execution across SO-101 units and standardized workbenches. As shown in Table 3, a BREE-calibrated replacement Arm B achieves similar performance (90% success) to Arm A (100%) using the same ACT policy trained on BREE Instance A with Arm A. Notably, deploying Arm A on BREE Instance B without recalibrating the arm achieves 100% success with the same policy trained on BREE Instance A. In contrast, the policy trained on data collected using standard SO-101 calibration drops from 70% to 0% success after arm replacement, despite using the same calibration procedure for both arms. These results support cross-arm policy execution and calibration reuse across BREE instances.

Global Camera Calibration. The structural frame defines a standardized configuration for the global camera, while small geometric differences can still arise during physical assembly. Without camera alignment, the policy fails, but achieves 100% success after BREE alignment. The same performance is maintained after replacing and realigning the global camera (Table 3). These results show that standardized mounting and alignment support reproducible policy performance across independently assembled BREE instances.

Lighting and Enclosure. BREE uses standardized internal lighting and an enclosed workspace to reduce external illumination variations. Using the same ACT checkpoint, success remains high under enhanced and dark external lighting (100% and 90%). In contrast, success drops to 30% with altered internal lighting and to 0% when relying only on ambient lighting (Table 4; Fig. 4). These results show that consistent lighting conditions are important for reproducible policy evaluation across BREE instances.

4.3 Cross-Instance Transfer and Multi-Instance Scaling

Cross-instance experiments test policy transfer and data reuse across independently assembled BREE instances. The main experiments use Task 1 in two spatially separated regions, Pic1 and Pic3 (Fig. 6), with the same predefined evaluation positions across instances.

Table 5: **Cross-instance policy transfer across independently assembled BREE instances.** Pick-and-Place uses Pic3 for training and evaluation; Trajectory Pushing follows the benchmark protocol.

Task	Evaluation Inst.	Training Data (# Demos)	Stage (%)	Full Task (%)
Pick-and-Place	BREE Inst. A	BREE Inst. A (25)	–	50
		BREE Inst. B (25)	–	45
		BREE Inst. A (25) + BREE Inst. B (25)	–	55
		BREE Inst. A (50)	–	65
	BREE Inst. B	BREE Inst. A (25) + BREE Inst. B (25)	–	55
		BREE Inst. A (50)	–	45
Trajectory Pushing	BREE Inst. A	BREE Inst. A (50)	55	15
	BREE Inst. B	BREE Inst. A (50)	42	0
		BREE Inst. A (50) + BREE Inst. B (5)	60	35

Cross-instance transfer. Direct transfer between independently assembled BREE instances is feasible without collecting data on the target instance. As shown in Table 5, a policy trained on 50 demonstrations from instance A achieves 45% success when directly deployed on instance B, demonstrating zero-shot cross-instance transfer without target-instance demonstrations or fine-tuning. For the Trajectory Pushing task, which involves a longer manipulation sequence, transfer from A to B reduces full-task success from 15% to 0%, showing that some performance variation remains across independently assembled BREE instances. However, the transferred policy still achieves 42% average stage success on B, compared with 55% on A. This indicates that the policy retains cross-instance transfer capability for part of the task, while fine-grained manipulation is more sensitive to residual instance differences. Surprisingly, jointly training ACT on 50 demonstrations from A and 5 additional demonstrations from B increases full-task success to 35% and average stage success to 60%. This result suggests that limited data from another BREE instance can compensate for residual instance differences and provides initial evidence for combining demonstrations across instances. We examine this effect more in the following multi-instance scaling experiment.

Multi-instance scaling. Combining demonstrations from two distinct spatial regions, Pic1 and Pic3, on the same BREE instance preserves the performance of the corresponding single-region policies (Table 6). We then distribute data collection across two BREE instances by retaining Pic1 demonstrations from A and collecting Pic3 demonstrations on B. Performance is maintained with this mixed-instance dataset, reaching 100% success at Pic1 and 60% at Pic3 on A. The resulting policy also achieves 60% at Pic3 on B. These results show that demonstrations can be collected independently across BREE instances and combined for training, supporting distributed data collection across physical platforms.

Table 6: **Multi-instance data scaling on Task 1 using ACT trained for 20k steps.** Success rates (%) are reported at two predefined spatial regions, Pic1 and Pic3.

Evaluation Instance	Training Data (# Demos)	Pic1	Pic3
BREE Inst. A	BREE Inst. A (Pic1, 25)	100	–
	BREE Inst. A (Pic3, 25)	–	50
	BREE Inst. A (Pic1, 25) + BREE Inst. A (Pic3, 25)	100	50
	BREE Inst. A (Pic1, 25) + BREE Inst. B (Pic3, 25)	100	60
BREE Inst. B	BREE Inst. B (Pic3, 25)	–	55
	BREE Inst. A (Pic1, 25) + BREE Inst. B (Pic3, 25)	100	60

4.4 Benchmark Evaluation

Benchmark Discrimination and Stage-Level Evaluation. Table 7 shows that policy performance varies substantially across the coordination types covered by BREE. Differences are relatively small on synchronized tasks but become more pronounced on asymmetric and sequential tasks. For example, on Cylinder Insertion, $\pi_{0.5}$ and SmolVLA achieve 100% and 95% full-task suc-

Table 7: **Benchmark performance on the BREE task suite.** Each entry reports *average stage success (%) / full-task success* (%), with unweighted averages across the eight tasks in the final row.

ID	Task	Coordination	Stages	ACT	RDT	SmolVLA	X-VLA	$\pi_{0.5}$
Synchronized Coordination								
01	Pick-and-Place	Synchronized	1	60.0/60	40.0/40	60.0/60	70.0/70	55.0/55
02	Switch Pushing	Synchronized	1	70.0/70	45.0/45	80.0/80	70.0/70	75.0/75
Asymmetric Coordination								
03	Trajectory Pushing	Asymmetric	3	55.0/15	53.3/40	71.7/65	75.0/50	85.0/85
04	Plate Loading	Asymmetric	2	65.0/30	67.5/50	97.5/95	62.5/30	92.5/85
05	Cup Hanging	Asymmetric	2	35.0/5	15.0/0	70.0/45	32.5/10	75.0/50
Sequential Coordination								
06	Block Stacking	Sequential	3	31.7/5	13.3/0	50.0/20	48.3/5	50.0/30
07	Drawer Placement	Sequential	3	46.7/20	38.3/25	58.3/45	50.0/35	76.7/65
08	Cylinder Insertion	Sequential	3	66.7/15	68.3/40	98.3/95	48.3/10	100.0/100
Average				53.8/27.5	42.6/30.0	73.2/63.1	57.1/35.0	76.1/68.1

Table 8: **Benchmark reproduction across BREE instances.** The same policy checkpoints are evaluated on BREE Instance A and B, reporting *average stage success / full-task success* (%).

Coordination	Task	Instance	ACT	RDT	SmolVLA	X-VLA	$\pi_{0.5}$
Synchronized	Switch Pushing	A	70.0/70.0	45.0/45.0	80.0/80.0	70.0/70.0	75.0/75.0
		B	70.0/70.0	40.0/40.0	75.0/75.0	45.0/45.0	75.0/75.0
Asymmetric	Plate Loading	A	65.0/30.0	67.5/50.0	97.5/95.0	62.5/30.0	92.5/85.0
		B	60.0/20.0	62.5/35.0	82.5/65.0	62.5/25.0	85.0/70.0
Sequential	Cylinder Insertion	A	66.7/15.0	68.3/40.0	98.3/95.0	48.3/10.0	100.0/100.0
		B	66.7/5.0	65.0/20.0	86.7/80.0	43.3/5.0	98.3/95.0

cess, whereas ACT, RDT, and X-VLA achieve 15%, 40%, and 10%. This variation highlights the value of covering different forms of bimanual coordination when comparing policies. Average stage success and full-task success provide complementary information. ACT and RDT achieve similar average stage success on Cylinder Insertion (66.7% and 68.3%) despite different full-task success (15% and 40%). Detailed results in Appendix C.1 further locate these differences at individual stages. These results show that BREE distinguishes policy performance across task structures while providing stage-level information for more detailed evaluation.

Benchmark Reproducibility across BREE Instances. Beyond the controlled experiments above, we validate cross-instance reproducibility using full-benchmark checkpoints on representative tasks from three task categories. As shown in Table 8, policies retain much of their learned task behavior when deployed from BREE Instance A to B without retraining. On the synchronized task, most policies maintain comparable performance across instances. On multi-stage tasks, policies (e.g., RDT) largely preserve performance in earlier stages, while degradation becomes more pronounced in later stages, leading to lower full-task success (40% $\rightarrow$ 20%). The preserved stage-level performance indicates that individual task stages remain reproducible across instances despite lower full-task success. These differences may stem from variations in servo operating conditions and frame assembly that affect physical interactions.

5 Conclusion

In this paper, we presented BREE, a low-cost and standardized real-world benchmark for bimanual robot learning, comprising BREE-Assets and BREE-Data. By standardizing the physical setup and experimental protocols across independently assembled instances, BREE supports reproducible evaluation and cross-instance reuse of policies and demonstrations. Our experiments demonstrate asset reproducibility and direct policy transfer without retraining, while showing that aggregating demonstrations from multiple instances improves policy performance. These findings establish BREE as a practical basis for reproducible bimanual benchmarking and scalable robot learning through distributed data collection.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing, LaTeX reformatting, reference organization, and figure layout planning. The authors reviewed all AI-assisted work and take responsibility for the final text, claims, citations, and artifacts.

REPRODUCIBILITY STATEMENT

BREE is designed to support reproducible real-world evaluation across independently assembled robotic platforms. The platform construction, hardware configuration, calibration procedures, task definitions, and evaluation protocol are described in the main paper and Appendix. BREE-Assets defines the standardized physical assets, construction specifications, and calibration procedures required to build a BREE instance, while BREE-Data provides the demonstrations and evaluation configurations. Training and inference settings for all evaluated policies are provided in the Appendix. The whole BREE stack (including code, 3D printing files, among others) will be released upon acceptance.

REFERENCES

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025. URL `https://proceedings.mlr.press/v305/black25a.html`.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Jack Collins, Mark Robson, Jun Yamada, Mohan Sridharan, Karol Janik, and Ingmar Posner. RAMP: A benchmark for evaluating robotic assembly manipulation and planning. *IEEE Robotics and Automation Letters*, 9(1):9–16, 2024. doi: 10.1109/LRA.2023.3330611.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations (ICLR)*, 2023.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems (RSS)*, 2023.

Alex S. Huang, Jiahui Zhang, Shiqing Tang, and Yu Xiang. VLA-REPLICA: A low-cost, reproducible benchmark for real-world evaluation of vision-language-action models. In *Data-Centric Robotics Workshop at Robotics: Science and Systems (RSS)*, 2026.

Ninad Khargonkar, Sai Haneesh Allu, Yangxiao Lu, Jishnu Jaykumar Padalunkal, Balakrishnan Prabhakaran, and Yu Xiang. SceneReplica: Benchmarking real-world robot manipulation by creating replicable scenes. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8258–8264, 2024. doi: 10.1109/ICRA57147.2024.10610180.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.120.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yAzN4tz7oI.

Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. FMB: A functional manipulation benchmark for generalizable robotic learning. *The International Journal of Robotics Research*, 44(4):592–606, 2025. doi: 10.1177/02783649241276017.

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. doi: 10.1109/LRA.2022.3180108.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun

Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Praveen Selvaraj, Lorenzo Uttini, and Ville Kuosmanen. ArmnetBench v0.1: Parallel real-world evaluation of manipulation policies on a low-cost arm farm. *arXiv preprint arXiv:2607.24481*, 2026.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Brian Yang, Dinesh Jayaraman, Jesse Zhang, and Sergey Levine. REPLAB: A reproducible low-cost arm benchmark for robotic learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8691–8697, 2019. doi: 10.1109/ICRA.2019.8794390.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

A BREE-ASSETS DETAILS

A.1 ROBOT CALIBRATION DETAILS

The BREE Calibration Aid establishes a shared physical reference pose across robot instances. Differences in servo installation may produce different raw encoder readings at this pose. Instance-specific homing offsets compensate for these differences, while shared predefined calibration ranges establish a common joint-coordinate definition. Accurate seating in the fixture is essential, as reference-pose errors can introduce systematic coordinate discrepancies across instances.

Homing-Offset Computation. With the arm seated in the fixture, let $P_{j,r}^{\mathrm{read}}$ denote the present encoder position read from joint j on robot instance r, and let $H_{j,r}^{\mathrm{stored}}$ denote the homing offset currently stored in the servo. Since the present position incorporates this offset, the raw encoder position $A_{j,r}$ and the homing offset required for BREE calibration are computed as

$$\begin{aligned} A_{j,r} &= \left(P_{j,r}^{\mathrm{read}} + H_{j,r}^{\mathrm{stored}}\right) \bmod 4096, \\ H_{j,r}^{\mathrm{cal}} &= A_{j,r} - c_j, \end{aligned} \tag{3}$$

where c_j is the shared encoder reference position of joint j at the fixture pose. When the required offset is within the servo's representable range, applying it aligns the calibrated encoder position to c_j, as defined in Eq. 1. Out-of-range offsets are handled as described below.

Table 9: **BREE home-pose configuration.** The targets specify the common initialization configuration in the calibrated LeRobot joint representation.

Joint	Home target
Shoulder pan	0°
Shoulder lift	-90°
Elbow flex	$+90^\circ$
Wrist flex	$+45^\circ$
Wrist roll	-67.52°
Gripper	100 (fully open)

Predefined Joint Ranges. After computing the homing offsets, BREE assigns predefined calibration ranges shared across robot instances. The shoulder pan, shoulder lift, elbow flex, and wrist flex joints use a fixture-relative angular range of $[-90^\circ, 90^\circ]$. For these joints, the fixture reference position is set to $c_j = 2047$ ticks, with predefined calibrated encoder limits of $P_j^{\min} = 1023$ and $P_j^{\max} = 3071$ ticks. Their range midpoint therefore coincides with the fixture reference, which corresponds to 0° under the degree conversion in Eq. 2.

The gripper uses a fixture-relative calibration range of $[-60^\circ, 45^\circ]$, with its reference encoder position set to 2047 ticks. Unlike the revolute joints, its recorded state and action coordinates are linearly mapped to $[0, 100]$. The wrist-roll joint uses a separate asymmetric calibration range, as described below.

Wrist-Roll Calibration. Unlike the other revolute joints, wrist roll uses an asymmetric motion range of $[-270^\circ, 45^\circ]$ relative to the fixture pose, corresponding to a total rotation of 315°. To represent this range as a continuous calibrated encoder interval, BREE assigns

$$P_5^{\min} = 0, \qquad P_5^{\max} = 3583, \qquad c_5 = 3071 \quad \text{ticks}. \tag{4}$$

The fixture reference is thus positioned toward the upper end of the calibrated range rather than at its numerical midpoint. This configuration is designed to avoid a wrap-around discontinuity within the effective calibrated motion range.

Since LeRobot's degree conversion uses the midpoint of the calibrated range, the wrist-roll fixture pose corresponds to approximately 112.48°, rather than 0°, in the recorded joint-angle representation. This reference coordinate is shared across BREE instances, preserving a consistent joint-coordinate definition despite the asymmetric calibration range.

Offset Validation and Mechanical Adjustment. The script checks the computed homing offsets against the representable range of each servo. For joints other than wrist roll, an out-of-range offset is clipped to the nearest representable value, and the resulting residual alignment error is reported.

For wrist roll, an out-of-range offset instead terminates calibration before any new calibration parameters are written or saved. The offset is not clipped, as clipping would introduce a systematic reference-alignment error. The script reports the minimum encoder displacement and corresponding angular adjustment required to bring the offset within the representable range.

The robot must then be powered off, the wrist-roll horn or spline mechanically repositioned, and calibration repeated with the arm fully seated in the fixture. The adjustment direction is determined by whether the raw encoder reading must increase or decrease. A feasible spline adjustment should place the resulting offset within the representable range, preferably with sufficient margin from its limits.

Home-Pose Specification. After calibration, BREE uses a predefined home pose as the common initialization configuration for both robot arms. The home pose is specified in the calibrated joint-coordinate system and is distinct from both the fixture reference pose and a servo-defined mechanical zero. It is defined by a fixed set of joint targets shared across BREE robot instances, as summarized in Table 9.

For the shoulder pan, shoulder lift, elbow flex, and wrist flex joints, the fixture reference coincides with 0° in the calibrated joint representation, and the home targets therefore directly specify their

displacement from the fixture pose. Wrist roll requires separate treatment because its asymmetric calibrated range places the fixture reference at approximately $112.48°$. The desired home configuration corresponds to a $-180°$ displacement from this fixture reference. In encoder coordinates,

$$P_5^{\text{home}} = c_5 - 180° \frac{4095}{360°} \approx 1023.5 \quad \text{ticks}, \tag{5}$$

which corresponds to approximately $-67.52°$ under the LeRobot degree conversion in Eq. 2. The same calibrated home targets are used for both arms, while instance-specific homing offsets account for differences in servo installation.

Home-Pose Setup Verification. As an additional post-calibration setup check, BREE evaluates two directly measurable geometric quantities at the predefined home pose. Let h_r^{home} denote the vertical distance from the platform reference plane to the highest point of robot instance r at the home pose, and let α_r^{home} denote the inclination angle of its upper arm relative to the platform reference plane. The nominal home-pose values are

$$h^{\text{ref}} = 20.5\,\text{cm}, \qquad \alpha^{\text{ref}} = 0°. \tag{6}$$

The corresponding deviations are defined as

$$e_{h,r}^{\text{home}} = \left| h_r^{\text{home}} - h^{\text{ref}} \right|, \qquad e_{\alpha,r}^{\text{home}} = \left| \alpha_r^{\text{home}} - \alpha^{\text{ref}} \right|. \tag{7}$$

For BREE reproduction, the home-pose setup check is satisfied when

$$e_{h,r}^{\text{home}} \leq 3\,\text{mm}, \qquad e_{\alpha,r}^{\text{home}} \leq 5°. \tag{8}$$

Both conditions must be satisfied. These quantities provide a practical geometric check of the reproduced home configuration and are not intended to characterize end-effector geometric accuracy over the full robot workspace.

A.2 Camera Calibration Details

BREE standardizes camera observations through camera pose alignment and, for the wrist cameras, fixed imaging parameters. The same camera configuration is used when collecting demonstrations and evaluating policies.

Camera Selection. During camera selection, we observed geometric distortion near the image boundaries in some candidate cameras, as well as differences in imaging characteristics across units of the same nominal model. We therefore considered edge distortion and consistency across units when selecting the cameras and lenses used in BREE.

Camera Pose Alignment. Camera mounting locations provide nominal viewpoints, while reference images are used to reduce residual pose differences after installation. Robot calibration is completed before camera alignment. Both arms are then commanded to the predefined BREE home pose in Table 9, providing a common robot configuration for camera alignment across platforms.

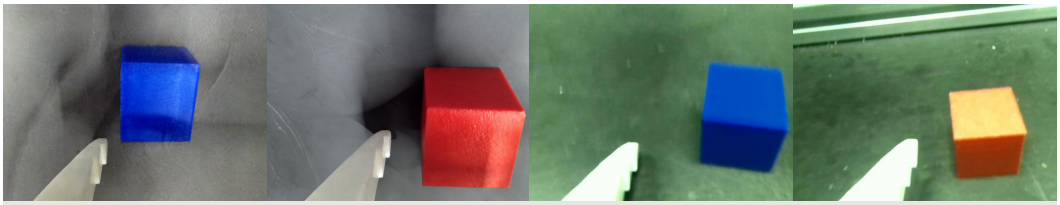
For the global camera, the camera pose is adjusted until the structural frame and robot configuration match the provided reference image. For each wrist camera, the gripper is used as the visual reference, and the camera pose is adjusted until the gripper matches its corresponding reference image. The resulting camera poses are fixed throughout data collection and evaluation.

Wrist-Camera Imaging Parameters. Under their default imaging settings, the left and right wrist cameras can produce noticeably different color appearances, as illustrated in Fig. 5. We apply the shared settings in Table 10 to both cameras during data collection and evaluation. Automatic white balance is disabled and the color temperature is fixed at $4600\,\text{K}$ to reduce scene-dependent color shifts as the robot and objects move. Gamma is reduced from 300 to 130 to reduce the washed-out appearance observed with the default settings. Brightness, contrast, saturation, and sharpness are set to 7, 48, 53, and 40, respectively. Dynamic exposure frame-rate adjustment is disabled to reduce the risk that longer exposure times lower the acquisition rate below the target of 30 FPS. This setting does not by itself guarantee the achieved frame rate. The global camera retains its default imaging settings.

Table 10: **Imaging parameters used for the BREE wrist cameras.**

Parameter	Default	BREE
Brightness	0	7
Contrast	40	48
Saturation	64	53
Automatic white balance	On	Off
White-balance temperature	4600 K	4600 K
Gamma	300	130
Sharpness	50	40
Dynamic exposure frame-rate	On*	Off

*The default behavior can depend on the camera state.



Before Adjustment After Adjustment

Figure 5: **Wrist-camera observations before and after imaging-parameter adjustment.** The left and right wrist cameras show noticeable color differences under their default settings. Applying the shared configuration in Table 10 reduces these differences and improves visual consistency.

Disabling automatic white balance and fixing the color temperature at 4600 K reduces color shifts between the two wrist-camera observations and prevents the color response from adapting to changes in scene content. The lower gamma setting reduces the washed-out appearance observed under the default configuration. Brightness, contrast, saturation, and sharpness are fixed to the values in Table 10 to provide a shared imaging configuration across wrist cameras.

Dynamic exposure frame-rate adjustment is disabled because increased exposure time can reduce the effective acquisition rate. This setting helps maintain the 30 FPS acquisition rate used for BREE demonstrations. These parameters remain fixed during both data collection and evaluation.

A.3 BREE Reproduction Checklist

After assembling a BREE platform, we recommend verifying the items in Table 11 before data collection or evaluation. The checklist covers the structural, calibration, sensing, lighting, task-asset, and data-collection configurations required to reproduce the BREE setup.

Table 11: **BREE reproduction checklist.** The checklist provides the main items for verifying the BREE setup before data collection and evaluation.

Category	Check Item	Check
Structural Frame	Left and right robot bases are positioned **29 cm from the corresponding side frames**, with a **32 cm distance between the two bases**.	☐
	The global camera mount is positioned **50 cm from each side frame**.	☐
	The outer edges of the overhead lighting units are positioned **15 cm from the corresponding side frames** and **11 cm from the upper frame**.	☐
	The workspace and task area conform to the predefined BREE structural layout.	☐
	All frame connections are securely fastened without visible looseness, deformation, or misalignment.	☐
Robot Calibration	Both robot arms are placed in the shared reference pose using the provided **physical calibration fixture**.	☐

Continued on next page

Table 11 continued

Category	Check Item	Check
	Both arms are correctly seated in the calibration fixture at the designated contact points.	☐
	The calibration files for both robot arms are correctly saved and loaded.	☐
	Joint ranges are set to **shoulder pan** $[-90^\circ, 90^\circ]$**, shoulder lift** $[-90^\circ, 90^\circ]$**, elbow flex** $[-90^\circ, 90^\circ]$**, wrist flex** $[-90^\circ, 90^\circ]$**, wrist roll** $[-270^\circ, 45^\circ]$**, and gripper** $[-60^\circ, 45^\circ]$.	☐
	After calibration, both arms reproduce the shared reference pose when seated in the physical calibration fixture.	☐
	Both arms reach the predefined BREE home pose specified in Table 9.	☐
	At the BREE home pose, both arms satisfy the geometric setup tolerances specified in Eq. 8.	☐
Global Camera	The global camera is a **DCXW100** with a **2.1 mm lens** and a **100**$^\circ$ field of view.	☐
	The camera is mounted using the designated fixed bracket at the predefined position and orientation.	☐
	The camera is fully seated and securely fixed without unintended translation or rotation.	☐
	The resolution is set to 640×480.	☐
	The frame rate is set to **30 FPS**.	☐
	Automatic exposure is enabled.	☐
	Automatic white balance is enabled.	☐
	The global camera field of view matches the provided **reference observation image**.	☐
Wrist Cameras	The left and right wrist cameras are **DCXW100** units with **2.8 mm lenses** and a **90**$^\circ$ field of view.	☐
	Both cameras are mounted using the designated fixed brackets at the predefined poses.	☐
	Both cameras are fully seated and securely fixed without unintended translation or rotation.	☐
	The resolution is set to 640×480.	☐
	The frame rate is set to **30 FPS**.	☐
	Automatic exposure is enabled for both cameras.	☐
	Automatic white balance is disabled for both cameras.	☐
	With both arms in the predefined BREE home pose, the camera fields of view match the provided **reference observations**.	☐
Lighting	**Two overhead lighting units** are installed.	☐
	The lighting color temperature is set to **6000 K**.	☐
	The lighting brightness is set to the **maximum level**.	☐
	The surrounding ambient-lighting condition matches the predefined BREE setup.	☐
	Camera observations under the standard lighting condition show no obvious overexposure, underexposure, or severe shadows.	☐
Task Assets	Task objects and fixtures are fabricated using the **provided 3D-printing files**.	☐
	Bambu Lab PLA Basic is used as the fabrication material.	☐
	Task-object colors match the BREE specifications.	☐
	All objects and fixtures required for the eight tasks are complete and free of visible damage.	☐
	Fixed task fixtures are installed at the predefined locations and aligned with the corresponding mounting features.	☐
	Task objects and fixtures are initialized according to the provided **task-specific reference configurations and images**.	☐
Data Collection	Data from **one global camera and two wrist cameras** are recorded.	☐
	All three camera streams are recorded at **30 FPS**.	☐
	Robot states and actions are recorded at **30 Hz**.	☐
	Robot states, actions, and camera observations are **temporally synchronized**.	☐
	Before each episode, both robot arms are reset to **the predefined BREE home pose** in Table 9.	☐

Continued on next page

Table 11 continued

Category	Check Item	Check
	Before each episode, all task objects are reset to the predefined initial configuration.	☐
	Demonstration data are stored in the specified **LeRobot format**.	☐
	Dataset metadata, FPS, camera keys, robot-state keys, and action keys conform to the **BREE data specification**.	☐

B BREE-DATA DETAILS

B.1 TASK DEFINITIONS AND DEMONSTRATION DATA

BREE contains eight bimanual manipulation tasks. We collect 50 demonstrations for each task and store all demonstrations in the LeRobot format. Each demonstration contains synchronized robot states, actions, and observations from one global camera and two wrist cameras. Table 12 summarizes the task definitions and evaluation stages.

B.2 EVALUATION PROTOCOL

Evaluation configurations. We use predefined evaluation configurations to ensure consistent evaluation across policies and BREE instances. Initial object poses and task configurations are specified by task-specific reference images. In the main benchmark, each policy is evaluated on five configurations using four inference seeds (0, 1, 2, and 3), yielding 20 evaluation pairs per task. Each pair consists of one configuration and one seed. The same configurations and seeds are used across policies and instances. The hardware and lighting experiments in Sec. 4.2 use the evaluation counts reported in their corresponding tables.

Execution budgets and termination. Table 13 specifies the maximum duration of each attempt. Timing begins when policy execution is initiated, before the first action is requested, and includes inference waiting time and action execution. Model loading and scene initialization are excluded. Each rollout terminates when the full-task success criteria are satisfied, the time limit is reached, or a safety stop or unrecoverable failure occurs. Elapsed time is the authoritative execution limit. The step counts in Table 13 give the corresponding nominal budgets at 30 Hz and do not denote model-query counts. The same task-specific limits apply to all policies and instances.

Autonomous Recovery. Each evaluation pair corresponds to a single rollout with the task-specific execution budget specified in Table 13. Within this budget, the policy may autonomously recover from intermediate failures. For example, a missed grasp followed by a successful regrasp is permitted and does not by itself constitute a failed rollout. Recovery does not reset the scene, policy state, action buffer, or execution budget. Success is determined by whether the task completion criteria are satisfied at rollout termination. Human assistance is prohibited except to terminate execution for safety. Such intervention constitutes a failed rollout when necessitated by policy behavior.

Full-task and stage-wise success. Full-task success requires all task-specific requirements to be satisfied within a single rollout and its execution budget, with the required final outcome preserved at termination. Intermediate failures do not preclude success if the policy autonomously recovers and completes the task. The full-task success rate is computed over all valid rollouts, with each evaluation pair contributing one rollout.

Stage-wise scoring credits outcomes that remain valid at termination. Transient completion that is subsequently undone by a failure is not credited. An intermediate stage may also be credited when its outcome is legitimately superseded by a subsequent task stage. For example, correctly closing a drawer does not invalidate an earlier successful drawer-opening stage, and returning a loaded plate does not invalidate the preceding plate-picking stage. Ordered trajectory stages are assessed from the execution sequence, since the final position alone does not establish successful traversal.

Each rollout contributes one full-task score and one score for each predefined stage. For a task with K stages and M valid rollouts, we compute

$$\mathrm{SR}_{\mathrm{full}} = \frac{1}{M}\sum_{i=1}^{M} y_i, \qquad \mathrm{SR}_{\mathrm{stage}} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{M}\sum_{i=1}^{M} s_{i,k}\right), \tag{9}$$

where y_i indicates full-task success in rollout i and $s_{i,k}$ indicates success of stage k in the same rollout under the above criteria. Rates are reported as percentages.

Table 12: **BREE task definitions and evaluation stages.**

Task	Task Name	Description	Stages
Synchronized Coordination			
01	Pick-and-Place	Both arms simultaneously pick up the red blocks and place them into the white plate.	1.1 Pick and Place Blocks
02	Switch Pushing	Both arms simultaneously push the switches forward.	2.1 Push Switches
Asymmetric Coordination			
03	Trajectory Pushing	The left arm grips the handle of the red base while the right arm pushes the green cylindrical slider along the constrained trajectory.	3.1 First Segment 3.2 Second Segment 3.3 Third Segment and Return
04	Plate Loading	The right arm picks up the plate, the left arm places the block on it, and the right arm returns the loaded plate to its original position.	4.1 Pick Plate 4.2 Place Block and Return Plate
05	Cup Hanging	The left arm stabilizes the cup holder while the right arm picks up the cup and hangs it on the holder.	5.1 Pick Cup 5.2 Hang Cup
Sequential Coordination			
06	Block Stacking	The right arm places the blue block at the designated position, followed by the left arm stacking the red block on top.	6.1 Pick and Place Blue Block 6.2 Pick Red Block 6.3 Complete Stacking
07	Drawer Placement	The left arm opens the drawer while the right arm places the green block inside, followed by closing the drawer.	7.1 Open Drawer 7.2 Pick and Place Block 7.3 Close Drawer
08	Cylinder Insertion	The right arm picks up the red cylinder, followed by the left arm picking up the blue cylinder and placing it onto the red cylinder.	8.1 Pick Red Cylinder 8.2 Pick Blue Cylinder 8.3 Assemble and Release

Table 13: **Maximum execution budget per rollout.** Task identifiers follow Table 12. Nominal control-step budgets are computed as $N_{\max} = 30T_{\max}$.

Task	Task name	Time limit (s)	Nominal steps at 30 Hz
01	Pick-and-Place	30	900
02	Switch Pushing	30	900
03	Trajectory Pushing	70	2100
04	Plate Loading	60	1800
05	Cup Hanging	40	1200
06	Block Stacking	70	2100
07	Drawer Placement	70	2100
08	Cylinder Insertion	40	1200

Invalid trials and replacement. Only attempts invalidated by verified errors unrelated to policy capability are excluded. Eligible cases include an initial scene that clearly deviates from the reference configuration, an incorrect reference image or task instruction, a camera that fails to open, a frozen camera stream, or an incorrect camera identifier. USB, servo-communication, or server-connection failures before action execution begins, as well as model or checkpoint loading failures, also qualify.

Other eligible cases include accidental operator termination, a failed reset that leaves the initial robot or scene configuration outside the prescribed requirements, and corrupted records or incomplete video caused by an independent recording failure.

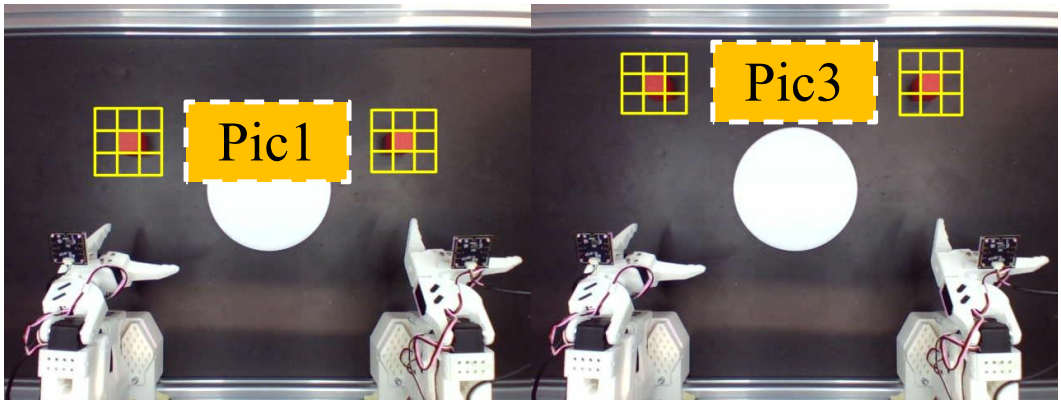


Figure 6: **Evaluation positions for the cross-instance experiments.** Predefined evaluation positions within the Pic1 and Pic3 regions.

An invalid rollout is excluded and repeated using the same evaluation configuration and inference seed. Its replacement contributes one valid rollout to the evaluation. Exclusion reasons and replacement counts are recorded. Missing records do not justify discarding an otherwise established policy-induced failure.

Failures retained in evaluation. Policy-induced failures are retained and scored as failed rollouts. These include dropping an object or moving it outside the workspace, collisions, servo overload, and safety stops caused by policy actions. Prolonged inactivity, repeated ineffective trajectories, incorrect actions, and entry into an unrecoverable state are also counted as failures.

Failure to complete the task within the time limit despite functioning inference, camera occlusion caused by policy behavior, and emergency operator intervention required because of excessive or unsafe policy motion are retained as failures. A valid failed rollout is not reclassified as invalid to permit an additional replacement.

Cross-instance evaluation regions. For the cross-instance experiments in Sec. 4.3, we additionally define two spatial regions, Pic1 and Pic3, for Pick-and-Place. As shown in Fig. 6, the same predefined evaluation positions are used within each region across BREE instances. This keeps the evaluation configurations fixed when changing the training-data source or evaluation instance.

C Policy Implementation Details

We provide the training and inference settings for the five policies evaluated on BREE. Each policy is trained separately for each task using the corresponding demonstrations. Table 14 summarizes the main implementation settings.

Inference Hardware. All evaluations on BREE Instance A and BREE Instance B are conducted using the same inference computer equipped with an NVIDIA GeForce RTX 4060 Ti with 16 GB of GPU memory. Direct cross-instance transfer evaluations use the same policy checkpoint without retraining or fine-tuning. These evaluations therefore assess policy reuse across independently assembled BREE instances while keeping the inference hardware fixed.

ACT. We train ACT from scratch for each task using the implementation and default policy configuration provided by LeRobot. Each task is trained independently for 20k optimization steps using its corresponding 50 demonstrations.

SmolVLA. We initialize SmolVLA from the official `lerobot/smolvla_robotwin` base model and fine-tune a separate policy for each task using the default SmolVLA training configuration provided by LeRobot. Each policy is fine-tuned using the 50 demonstrations from the corresponding BREE task.

RDT. We initialize RDT from the official `RDT-170M` pretrained model and fine-tune a separate policy for each task. We follow the official RDT training configuration and use an action horizon of 64.

X-VLA. We initialize X-VLA from the official pretrained checkpoint and fine-tune a separate policy for each task. We follow the official training configuration for task-specific adaptation using the corresponding 50 demonstrations.

π0.5. We initialize π0.5 from the `pi05_base` checkpoint and fine-tune a separate policy for each task. We apply LoRA to the PaliGemma 2B backbone and the Gemma 300M action expert. The action dimension is set to 32 with an action horizon of 50. Task language instructions are provided during training.

Table 14: **Implementation settings of the evaluated policies.**

Policy	Pretrain	Steps	Batch	LR	Horizon	Actions/Query
ACT	From scratch	20k	8	1×10^{-5}	100	100
SmolVLA	`smolvla_robotwin`	20k	12	1×10^{-4}	50	50
RDT	`RDT-170M`	20k	8	1×10^{-4}	64	64
X-VLA	`xvla-base`	20k	2	1×10^{-4}	30	30
π0.5	`pi05_base`	20k	8	default	50	50

C.1 Benchmark Results

Table 15 reports the stage-wise success rates of the evaluated policies across the eight BREE tasks.

Table 15: **Stage-wise evaluation results on BREE.** All entries report success rates in percent.

Task	Task Name	Stage	ACT	RDT	SmolVLA	X-VLA	$\pi_{0.5}$
Synchronized Coordination							
01	Pick-and-Place	1.1 Pick and Place Blocks	60	40	60	70	55
02	Switch Pushing	2.1 Push Switches	70	45	80	70	75
Asymmetric Coordination							
03	Trajectory Pushing	3.1 First Segment	95	70	80	90	85
		3.2 Second Segment	55	50	70	85	85
		3.3 Third Segment and Return	15	40	65	50	85
04	Plate Loading	4.1 Pick Plate	100	85	100	95	100
		4.2 Place Block and Return Plate	30	50	95	30	85
05	Cup Hanging	5.1 Pick Cup	65	30	95	55	100
		5.2 Hang Cup	5	0	45	10	50
Sequential Coordination							
06	Block Stacking	6.1 Pick and Place Blue Block	65	40	90	90	65
		6.2 Pick Red Block	25	0	40	50	55
		6.3 Complete Stacking	5	0	20	5	30
07	Drawer Placement	7.1 Open Drawer	100	65	80	75	100
		7.2 Pick and Place Block	20	25	50	40	65
		7.3 Close Drawer	20	25	45	35	65
08	Cylinder Insertion	8.1 Pick Red Cylinder	95	90	100	95	100
		8.2 Pick Blue Cylinder	90	75	100	40	100
		8.3 Assemble and Release	15	40	95	10	100