BENCH2DEX: A BIMANUAL MANIPULATION BENCHMARK ACROSS DEXTEROUS HANDS WITH SIMULATED FINGERTIP VISUO-TACTILE SUPPORT

Anonymous authors
Paper under double-blind review

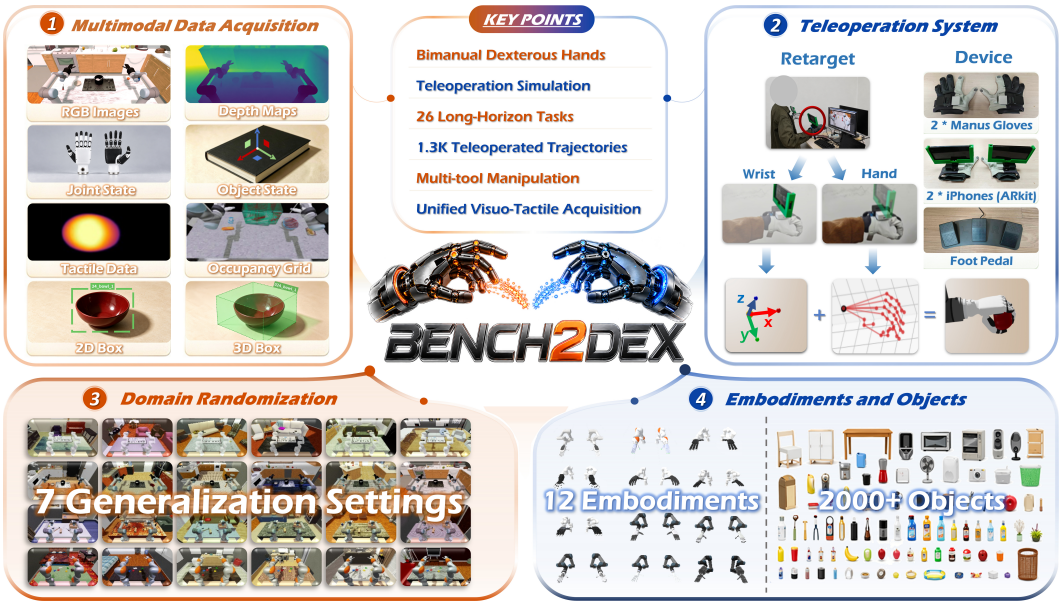


Figure 1: **Overview of Bench2Dex.** Bench2Dex is a simulation benchmark for bimanual dexterous manipulation with 26 tasks, 12 robot embodiments, and about 1.3K human-teleoperated demonstration trajectories, collected across a range of objects and scenes. It supports teleoperated demonstration collection and 8 data modalities: RGB images, depth maps, joint states, object states, a shared visuo-tactile representation, 2D/3D bounding boxes, and occupancy grids. With this multimodal data and controlled domain randomization, Bench2Dex supports evaluation of policy performance, robustness, and generalization on everyday bimanual manipulation tasks.

ABSTRACT

Bimanual dexterous manipulation requires coordinated control of two hands in multi-stage, contact-rich tasks. We introduce **Bench2Dex**, a simulation benchmark for bimanual manipulation supporting 12 dexterous hands. The benchmark includes 26 tasks involving tool use, interaction with articulated objects, and multi-stage coordination, together with approximately 1.3K human-teleoperated demonstrations. It integrates demonstration collection, synchronized multimodal data, and automated metrics for task completion and intermediate progress within a common framework. *Taking a deliberate step beyond vision-only sensing*, we equip every supported hand model with simulated vision-based tactile sensing at the fingertips, encoding local contact geometry as image-like tactile observations. Through a **unified visuo-tactile interface**, we make these observations available alongside external vision and proprioception to support research on tactile representations and multimodal policy learning. We establish reference results using ACT, Diffusion Policy, $\pi_{0.5}$, and GR00T N1.5 with visual and proprioceptive inputs. We assess robustness across four evaluation channels—**None**, **Equivariance-only**, **Invariance-only**, and **Full**—covering an unperturbed baseline and controlled combinations of seven scene perturbation types. We further

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.

conduct an exploratory GR00T N1.5 comparison with and without tactile input. Averaged across the four evaluation channels, the reported stable-success rates are higher with tactile input on 13 of the 26 tasks and lower on the remaining 13, with task-averaged rates of 27.3% with tactile input versus 28.3% without. By combining a bimanual manipulation benchmark across dexterous hands with simulated visuo-tactile support and initial tactile-conditioned policy evaluations, Bench2Dex provides the community with a common starting point for investigating when tactile information is useful and how manipulation policies can effectively incorporate it.

1 INTRODUCTION

Dexterous manipulation is a core capability for embodied agents: it lets robots interact with tools, articulated objects, and everyday environments through direct physical contact [68; 72; 2; 47; 39]. Driven by large-scale teleoperation datasets [35; 66; 85; 17], simulation benchmarks [78; 32; 44; 41; 21; 47; 42; 37], cross-embodiment data efforts [13; 70], and vision-language-action policies [6; 5; 64; 36], robot learning has made steady progress in scale, task coverage, and reproducibility.

However, most manipulation benchmarks and datasets are still built around a limited set of robot platforms, one gripper or hand design, or a narrow set of sensing modalities [78; 32; 44; 41; 21; 47; 35; 66]. Some large datasets now cover multiple embodiments, but their evaluation protocols are not built to compare dexterous manipulation policies across different arm-hand designs and contact surfaces [13; 70; 16].

This gap matters most for visuo-tactile learning. Vision-based tactile sensors output image-like contact signals that depend closely on fingertip shape and sensor layout [27; 14; 38; 69; 59]. As a result, tactile signals recorded on one hand do not transfer directly to another hand, and, to our knowledge, no existing benchmark supports visuo-tactile data collection across multiple dexterous hands under one setting [27; 56; 55; 8]. Due to limited space, we discuss more details of Related Works in Appendix A.

We introduce **Bench2Dex**, a simulation benchmark for bimanual manipulation built on Isaac Lab [51]. Bench2Dex supports 12 dexterous hand models and includes 26 tasks involving tool use, interaction with articulated objects, and multi-stage coordination. It provides approximately 1.3K human-teleoperated demonstrations collected in simulation, together with synchronized multimodal data. Each task has a structured scene specification and executable conditions for task completion and intermediate progress. These components connect demonstration collection, policy learning, and evaluation within a common framework.

Taking a deliberate step beyond vision-only sensing, we equip every supported hand model with simulated vision-based tactile sensing at the fingertips. Through a **unified visuo-tactile interface**, we provide image-like observations of local contact geometry alongside external vision and proprioception, while retaining hand-specific sensing geometry. This supports research on when tactile information is useful and how manipulation policies can effectively incorporate it.

To examine robustness, we organize seven scene perturbation types into four evaluation channels: **None**, **Equivariance-only**, **Invariance-only**, and **Full**. None provides an unperturbed, matched-scene reference. Equivariance-only varies object pose and table height, while Invariance-only introduces variations in tabletop texture, lighting, scene background, camera pose, and distractor objects. Full combines both groups. This organization distinguishes perturbations to task geometry from variations in appearance, viewpoint, and scene context. Reporting results separately by channel, alongside task completion, intermediate progress, and efficiency analysis, allows us to examine how policy performance changes under different types of scene perturbations, rather than reducing robustness to a single aggregate score.

We establish reference results using ACT [85], Diffusion Policy [11], $\pi_{0.5}$ [31], and GR00T N1.5 [50] with visual and proprioceptive inputs. These evaluations provide non-tactile reference performance for the bimanual tasks and robustness protocol. We further conduct an exploratory GR00T N1.5 comparison with and without tactile input. Averaged across the four evaluation channels, the reported stable-success rates are higher with tactile input on 13 of the 26 tasks and lower on the remaining 13. The corresponding task-averaged rates are 27.3% with tactile input and 28.3% without. These mixed outcomes motivate further investigation into tactile representations, multi-

Table 1: Comparison of Bench2Dex with Representative Manipulation Benchmarks.

Benchmark	Task Num	Embodiment	Tool/Device Use	Dexterous Hand	Teleoperation	Vision-Based Tactile	Articulated
LIBERO [41]	130	1	✗	✗	✓	✗	✓
RLBench2 [20]	13	1	✓	✗	✗	✗	✓
RoboCasa [47]	100	1	✓	✗	✓	✗	✓
DROID [35]	86	1	✗	✗	✓	✗	✗
DexMimicGen [33]	9	3	✗	✓	✓	✗	✓
RealMirror [61]	5	1	✓	✓	✓	✗	✓
RoboTwin 2.0 [8]	50	5	✓	✗	✗	✗	✓
RoboMIND 2.0 [26]	739	6	✓	✓	✓	✗	✓
Manipulus [80]	16	1	✓	✗	✗	✗	✗
RoboCasa365 [48]	365	1	✓	✗	✓	✗	✓
RoboDojo [9]	60	3	✓	✗	✓	✗	✓
BiCoord [53]	18	1	✗	✗	✗	✗	✗
DexJoCo [67]	11	2	✓	✓	✓	✗	✓
DexVerse* [76]	19	1	✓	✓	✓	✗	✓
Bench2Dex (Ours)	**26**	**12**	✓	✓	✓	✓	✓

Note. * Statistics based on publicly available data as of the submission date.

modal fusion, and task-specific sensing requirements, rather than a claim of uniform benefit from adding touch. **Our main contributions are as follows:**

- We introduce **Bench2Dex**, a simulation benchmark for bimanual dexterous manipulation including 12 robot models, 26 long-horizon tasks, and 1.3K human-teleoperated demonstrations.
- We develop a ***unified visuo-tactile interface*** that provides image-like observations of local contact geometry across all supported dexterous hands while preserving hand-specific sensing geometry. Together with vision, proprioception, and action signals, it provides synchronized data across eight modalities for tactile and multimodal policy learning.
- We establish an evaluation suite covering task completion, stage-level progress, and success-conditioned efficiency, with seven scene perturbation types organized into four evaluation channels: ***None***, ***Equivariance-only***, ***Invariance-only***, and ***Full***. These channels separate changes in task geometry from variations in appearance, viewpoint, and scene context, and evaluate their combined effects.
- We benchmark ACT, Diffusion Policy, $\pi_{0.5}$, and GR00T N1.5 under matched and perturbed scenes. We further compare GR00T N1.5 with and without tactile input, revealing task-dependent effects of tactile information and motivating further study of when and how manipulation policies can effectively use touch.

2 Bench2Dex Benchmark

Bench2Dex is designed as a full benchmark pipeline rather than a task collection alone. As shown in Figure 1, it connects task construction, human teleoperation, multimodal data collection and executable evaluation into a unified simulation framework. This section describes the benchmark components used to generate data and evaluate policies.

2.1 Online Teleoperation Setup

Bench2Dex uses human teleoperation to collect demonstrations for long-horizon bimanual dexterous manipulation. The teleoperation system is integrated directly into the Isaac Lab simulation loop, so each episode is recorded together with the commanded action, robot state, object state, camera observations, task metadata, and success or metric signals. The recording pipeline is designed to introduce minimal per-step overhead, preserving the responsiveness required for real-time human teleoperation. Teleoperation therefore serves not only as a data collection tool, but also as a practical feasibility check for whether a task can be executed under a given robot embodiment and scene configuration.

The operator controls the robot through a Manus glove and an ARKit wrist-tracking stream. The Manus runtime provides a 25-node hand skeleton through shared memory; the system converts it to 21 MediaPipe-style hand keypoints and uses DexPilot retargeting to solve target joint angles for the active robot hand. The same retargeting interface is configured for twelve dexterous hand embodiments, giving all supported hands a unified teleoperation and data-collection protocol. Arm motion is controlled separately: ARKit provides wrist translation and orientation cues, while a Pinocchio-based closed-loop inverse-kinematics controller maps the desired wrist pose to the corresponding

arm joint targets. The hand and arm targets are then merged into a single absolute joint-position command for the full bimanual robot.

During recording, Bench2Dex stores the action to be executed at the next simulation step, followed by the resulting post-step observations and evaluator state. Before starting or saving a trajectory, the robot is moved to a home configuration, which reduces discontinuities between demonstrations and resets the teleoperation filters and wrist anchors.

2.2 Offline Multimodal Data Acquisition

The lightweight online recording described above captures only the essential motion stream. The remaining modalities—multi-view RGB-D images, object bounding boxes, occupancy labels, and surface-aligned tactile observations—are generated by an offline replay pipeline that reconstructs each episode in simulation and renders the full sensor suite. This decoupling of teleoperation from expensive rendering is a key design choice: the human operator session is kept short, while the replay step can be parallelized, re-run with updated sensor configurations, or selectively applied to a subset of episodes.

Replay pipeline. For each recorded episode, the replay pipeline loads the saved object initial states and robot joint trajectory. Multi-view RGB-D images are rendered from the six calibrated camera viewpoints (chest, overhead, stereo left, stereo right, left wrist, right wrist). Semantic and instance segmentation labels are extracted from the rendered outputs, occupancy grids are voxelized from the scene geometry, and object bounding boxes in both 2D image coordinates and 3D world coordinates are generated from the projected object meshes. The HDF5 writer keeps camera calibration, action metadata, frame validity, and task metadata together with these observations, making replay, evaluation, and cross-embodiment comparison consistent across tasks and robot hands. Tactile observations are generated during the same replay pass using the surface-aligned ray-casting pipeline described in Section 2.3.

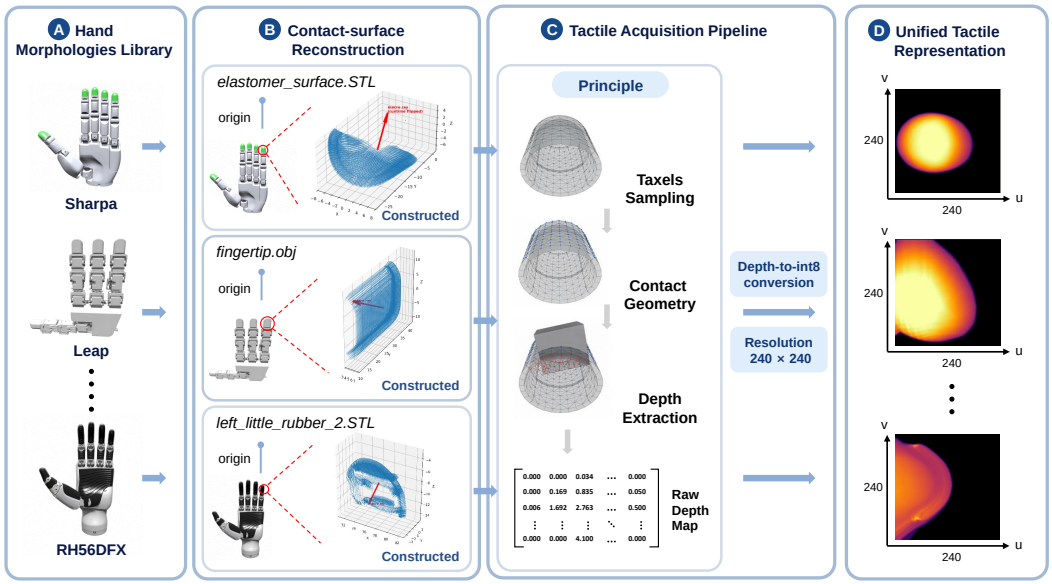


Figure 2: **Unified surface-aligned tactile acquisition for diverse robotic hands.** Contact surfaces are reconstructed for twelve robot-hand morphologies, and a shared surface-aligned ray-casting pipeline maps the heterogeneous hand geometries into a common image-like tactile representation.

2.3 Unified Visuo-Tactile Data Acquisition

Contact is central to bimanual dexterous manipulation. After a robot grasps a tool, pushes an articulated part, or stabilizes an object with the other hand, ***the most informative interaction is often hidden from external cameras.*** Prior systems such as TACTO [69] and Taxim [59] established

image-based tactile simulation, while TacMap [60] introduced a geometry-consistent penetration-depth representation over contact surfaces. Drawing on these paradigms, we develop a unified surface-aligned tactile acquisition layer for Bench2Dex. Our contribution is a cross-embodiment interface that combines reconstructed hand contact surfaces, a shared site registry, consistent encoding, and offline replay to expose heterogeneous robot hands through one tactile data representation.

As illustrated in Figure 2, for each tactile site s, our acquisition layer uses a precomputed local surface map derived from the hand contact mesh. Each grid cell corresponds to a fixed surface point with an associated inward-facing normal. During replay, rays are cast from these surface points along the inward normals against task-object meshes, so the first-hit depth measures how far an object surface has penetrated past the nominal contact surface; the contact-consistency-filtered depth is then encoded as an 8-bit tactile image $\mathbf{T}_{s,t} \in \{0, \dots, 255\}^{H \times W}$ for site s at time t. Bench2Dex also retains the metric ray depth and a binary contact-validity mask, while the compact tactile stream applies a piecewise depth encoding and spatial smoothing. The complete signal semantics and quantization procedure are provided in Appendix D.1. This representation converts sparse geometric contact into an image-like signal that can be processed by visual encoders or fused with RGB-D, proprioception, and object-state observations.

A key design goal is embodiment consistency. Bench2Dex uses one tactile acquisition paradigm for the twelve bimanual robot-hand embodiments included in the benchmark, while preserving the local contact geometry of each hand. To achieve this, we rebuilt the tactile contact-surface meshes for each benchmark hand and converted them into surface point-and-normal assets, provided in Appendix D. A tactile registry then maps each robot embodiment to its tactile site names, surface assets, resolution, normal convention, and site-body attachment rules, avoiding hand-specific branching in the data collection code.

2.4 Datasets and Policy

Dataset. Bench2Dex contains about 1.3K human-teleoperated demonstrations across 26 long-horizon tasks and 12 bimanual dexterous embodiments. Each trajectory is replayed into a unified HDF5 record with synchronized RGB-D observations, joint states and actions, object states, surface-aligned tactile maps, 2D/3D bounding boxes, and occupancy grids.

Policy. We evaluate ACT [85], Diffusion Policy (DP) [11], $\pi_{0.5}$ [31], and GR00T N1.5 [50]. ACT and DP are trained from scratch using multi-view RGB and joint-state observations, whereas $\pi_{0.5}$ and GR00T N1.5 are fine-tuned from pretrained checkpoints. For the pretrained policies, state/action projections are adapted to each embodiment where required, with shape-incompatible or newly added parameters initialized randomly. The model details are in Appendix G.

2.5 Evaluation Metrics

Bench2Dex uses a unified evaluation suite for task completion, stage-level progress, and success-conditioned efficiency. Unless stated otherwise, we use a reach-and-stop protocol: each task provides an executable terminal predicate, and an episode terminates successfully only after that predicate remains true for its configured dwell time (0.5 s by default). Rollouts otherwise terminate at the evaluation budget or when evaluation cannot proceed. The protocol, step budget, dwell time, resolved seed, and sampled generalization parameters are retained for each episode.

The primary completion metric is stable success rate,

$$\mathrm{SR} = \frac{1}{N} \sum_{e=1}^{N} S_e, \tag{1}$$

where N is the number of evaluated episodes in the reported group and $S_e = 1$ only when the terminal predicate stably holds, rather than at a transient frame. To characterize partial progress on long-horizon tasks, the benchmark also provides the *latched stage completion rate* (LSCR),

$$\mathrm{LSCR}_e = \frac{1}{K_e} \sum_{k=1}^{K_e} L_{e,k}, \tag{2}$$

where K_e is the number of stages and $L_{e,k} = 1$ if stage k is reached at any time while its declared dependencies are satisfied by previously latched or concurrently reached stages. Latching prevents

progress from being erased when an early predicate is intentionally reversed by a later action, such as closing door after opening it.

For efficiency analysis, let H_q^{expert} be the expert reference step count for task q, expressed in the same step unit as the policy count, and let $\overline{H}_{q,m}^{\text{success}}$ be the mean number of executed steps to stable success for policy m, computed by pooling all successful episodes from the reported evaluation channels. The step conventions and expert references used for each policy are specified in Section 3.1. We report expert-normalized efficiency

$$\text{Efficiency}_{q,m} = \frac{H_q^{\text{expert}}}{\overline{H}_{q,m}^{\text{success}}}, \tag{3}$$

where a value of one denotes equal expert-reference and policy step counts under the stated counting convention, and higher is better. This success-conditional metric is undefined when no episode succeeds and is always interpreted together with SR. For generalization analysis, episodes are grouped into the `None`, `Equi.`, `Inv.`, and `Full` channels, for which the benchmark reports stable-success counts and SR. Complete metric definitions and reproducibility details are provided in Appendix F.

2.6 Generalization Strategy

Bench2Dex evaluates whether a policy learns task-level manipulation concepts rather than memorizing a fixed simulator instance. Each evaluation keeps the semantic goal, object set, and success conditions unchanged while controlling two groups of scene factors. Scene background, tabletop texture, lighting conditions, distractor objects, and camera pose are treated as **Invariance** factors because they alter task-irrelevant visual conditions without changing the intended task behavior. Object pose and table height are treated as **Equivariance** factors because they alter task-relevant geometry and therefore require corresponding changes in reaching, grasping, and contact trajectories.

The protocol defines four evaluation channels. **No generalization (`None`)** exactly restores a matched anchor episode and serves as the baseline. **Equivariance-only generalization (`Equi.`)** resamples only equivariance factors while preserving the anchor's invariant context, testing geometric adaptation. **Invariance-only generalization (`Inv.`)** resamples only invariance factors while preserving the anchor's task geometry, testing robustness to nuisance variation. **Combined generalization (`Full`)** independently resamples both groups without an anchor, testing whether a policy can simultaneously ignore contextual shifts and adapt to new task geometry.

For each episode e, Bench2Dex samples a generalization configuration $\mathbf{g}_e$ and instantiates the executable scene as

$$\mathcal{S}_e = \text{Build}(\mathcal{T}, \mathbf{g}_e), \tag{4}$$

where $\mathcal{T}$ denotes the task specification and $\mathbf{g}_e$ stores the resolved parameters for scene background, tabletop texture, lighting conditions, distractor objects, object pose, table height, and camera pose. The resolved configuration is saved with the episode metadata, making every channel replayable and auditable. Appendix I specifies the sampling ranges and channel composition.

3 Experiments

3.1 Experimental Setup

Our main comparison evaluates ACT, an image-conditioned 1D U-Net Diffusion Policy variant (DP), $\pi_{0.5}$, and GR00T N1.5 using RGB and proprioceptive observations on all 26 Bench2Dex task–embodiment settings spanning 12 embodiments (Appendix B). Each policy is evaluated with 50 rollouts under four generalization channels, yielding $26 \times 4 \times 4 \times 50 = 20{,}800$ evaluation episodes. Evaluation uses deterministic task-partitioned episode seeds and a task-specific horizon set to $1.5\times$ the recorded expert horizon. Channel semantics follow Section 2.6: `None`, `Equi.`, and `Inv.` are aligned by episode index to matched anchors, whereas `Full` is sampled independently without an anchor. Cross-channel comparisons below therefore use marginal success counts and do not assume a per-instance difficulty ordering.

The primary outcome is reach-and-stop stable success (Section 2.5), and each channel entry in Table 2 is a success count out of 50 rollouts. For a fixed policy and channel, we report the equal-weight task-macro success rate over the 26 settings. Since every setting contributes 50 episodes, this rate

Table 2: **Stable-success counts across generalization channels on all 26 Bench2Dex task–embodiment settings spanning 12 embodiments.** Each task entry is the number of reach-and-stop successes among 50 rollouts. None, Equi., Inv., and Full correspond to the `none`, `equi_only`, `inv_only`, and `inv_equi` scene profiles, respectively. **SR** and **LSCR** are the four-channel mean stable success rate and latched stage completion rate, respectively, for each task–policy pair. **Eff.** is expert-normalized efficiency, computed by directly pooling successful episodes across the four channels. Higher values are better for all three metrics, and "–" denotes an unavailable efficiency value. **Bold** indicates the best **SR**, **LSCR**, and **Eff.** among the four policies. The bottom row reports the equal-weight task-macro SR (%), all-task mean LSCR, and mean efficiency over tasks with at least one success (25 tasks for ACT, 20 tasks for DP, 23 for $\pi_{0.5}$, and 26 for GR00T N1.5).

Task	Embodiment	ACT							DP							$\pi_{0.5}$							GR00T N1.5						
		None	Equi.	Inv.	Full	SR	LSCR	Eff.	None	Equi.	Inv.	Full	SR	LSCR	Eff.	None	Equi.	Inv.	Full	SR	LSCR	Eff.	None	Equi.	Inv.	Full	SR	LSCR	Eff.
Baking Tray Prep	IIWA7+Sharpa	9	2	1	0	6.0%	11.6%	0.924	2	2	0	0	2.0%	7.9%	1.016	22	8	15	10	**27.5%**	**38.4%**	**1.037**	12	7	1	0	10.0%	29.3%	0.929
Canned Food Tray Arrangement		17	2	0	0	9.5%	34.6%	**0.966**	0	0	0	0	0.0%	14.8%	–	15	4	2	3	12.0%	**38.8%**	0.927	17	5	3	0	**12.5%**	37.3%	0.898
Jigsaw Puzzle Assembly		0	0	0	0	0.0%	12.5%	–	0	0	0	0	0.0%	6.3%	–	0	0	0	0	0.0%	8.4%	–	10	0	0	0	**5.0%**	**30.3%**	**0.957**
Gaming Desk Setup	JAKA ZU7+DexHand021	25	10	17	10	31.0%	46.0%	1.013	7	6	4	1	9.0%	30.7%	0.964	16	10	21	20	33.5%	49.3%	**1.052**	41	17	34	27	**59.5%**	**72.2%**	0.992
Medicine Shoebox Pack	Panda+Orca	5	1	5	4	7.5%	21.5%	**1.004**	0	0	0	0	0.0%	9.3%	–	15	2	14	12	**21.5%**	**39.8%**	0.994	18	3	3	2	13.0%	26.2%	0.956
Tool Box Loading		17	10	8	5	20.0%	37.5%	**0.983**	14	6	7	6	16.5%	39.8%	0.898	41	34	35	34	**72.0%**	**86.0%**	0.970	27	15	21	17	40.0%	51.0%	0.940
Sports Ball Cup Sort	Panda+Allegro	4	0	0	0	2.0%	2.5%	0.728	1	0	0	0	0.5%	4.4%	0.725	21	9	2	11	**21.5%**	**33.6%**	**0.769**	15	0	2	1	9.0%	21.5%	0.741
Faucet Cup Water Fill	RM65+Revo2	6	5	2	2	7.5%	35.4%	0.899	3	2	1	2	4.0%	14.0%	0.926	7	6	9	9	15.5%	40.5%	0.891	24	11	18	13	**33.0%**	**55.0%**	**0.959**
Stationery Category Sorting		3	0	0	0	1.5%	**23.1%**	0.916	3	0	1	0	2.0%	18.5%	**1.078**	0	0	0	0	0.0%	2.3%	–	5	1	0	0	**3.0%**	16.9%	1.015
Bimanual Piano Melody	xArm7+Ability	6	0	0	2	4.0%	4.5%	**0.978**	1	0	0	0	0.5%	0.5%	0.854	6	5	2	2	7.5%	16.0%	0.882	30	8	2	6	**23.0%**	**23.5%**	0.931
Wine Glass Plate Balance		15	11	13	13	26.0%	72.0%	**1.089**	3	2	1	1	3.5%	39.8%	0.948	16	14	21	19	**35.0%**	**76.2%**	1.059	19	9	16	13	28.5%	72.5%	1.008
Cleaner Box Loading	xArm7+LEAP	34	20	7	5	33.0%	76.2%	**0.951**	11	3	2	2	9.0%	32.8%	0.795	36	21	29	24	**55.0%**	**88.5%**	0.885	35	18	24	17	47.0%	82.3%	0.801
Shoebox Accessory Pack		19	18	11	12	30.0%	56.3%	**1.025**	6	5	5	3	9.5%	35.2%	0.850	29	21	28	24	**51.0%**	**68.3%**	1.005	32	15	18	16	40.5%	60.3%	0.956
Toilet Lid Cleaner Pour		16	19	20	13	34.0%	50.3%	0.990	7	5	2	3	8.5%	23.5%	0.965	21	22	25	23	**45.5%**	53.3%	**1.058**	32	12	25	21	45.0%	**63.3%**	0.977
Breadbasket Fast-Food Loading	UR5+RH5DG2	11	6	3	1	10.5%	**15.8%**	0.951	5	1	2	1	4.5%	7.3%	0.936	13	3	6	3	**12.5%**	15.6%	**0.959**	15	2	1	0	9.0%	13.5%	0.943
Citrus Plate Loading		43	20	14	26	51.5%	69.9%	0.838	23	7	9	6	22.5%	44.4%	0.819	35	12	34	32	56.5%	67.3%	**0.840**	48	22	24	24	**59.0%**	**73.1%**	0.834
Fridge Wine Interhand Pour		8	4	4	6	11.0%	48.5%	1.053	4	0	1	0	2.5%	41.8%	0.924	6	2	4	5	8.5%	36.1%	**1.095**	14	6	14	8	**21.0%**	**58.6%**	1.047
Fruit Bowl Loading	UR5+RH56DFX	34	15	13	12	37.0%	62.0%	**0.960**	15	6	9	6	18.0%	47.8%	0.927	36	10	12	19	**38.5%**	**71.6%**	0.941	41	8	11	5	32.5%	68.1%	0.914
Screwdriver Box & Hammer		16	5	18	10	24.5%	56.2%	**1.033**	18	5	9	9	20.5%	49.0%	0.913	18	18	14	16	33.0%	57.5%	0.935	35	18	22	21	**48.0%**	**73.7%**	0.956
Trash Disposal		15	6	9	7	18.5%	30.3%	**0.946**	10	8	4	6	14.0%	21.5%	0.904	26	16	22	18	**41.0%**	**53.7%**	0.929	26	10	19	14	34.5%	47.8%	0.882
Fridge Fruit Shelf Sorting	UR5+Shadow	1	0	0	0	0.5%	2.6%	**1.069**	0	0	0	0	0.0%	15.1%	–	7	3	0	2	**6.0%**	**29.6%**	0.898	2	0	0	0	1.0%	19.6%	0.979
Soup Serving		2	0	0	0	1.0%	11.5%	0.883	0	0	0	0	0.0%	3.1%	–	0	0	0	0	0.0%	**14.0%**	–	5	1	0	0	**3.0%**	13.9%	**0.933**
Frypan Stand & Pour	UR5+Schunk	20	8	3	7	19.0%	51.2%	**1.214**	16	9	4	0	14.5%	30.0%	1.192	8	9	7	6	15.0%	33.3%	1.063	39	22	30	18	**54.5%**	**72.3%**	1.177
Microwave Bowl Loading		38	29	29	29	62.5%	74.9%	1.016	17	15	6	4	21.0%	48.6%	0.956	45	32	32	31	70.0%	78.4%	**1.026**	45	27	36	33	**70.5%**	**82.0%**	1.003
Ball Box Loading	UR5+Wuji	7	6	2	2	8.5%	10.4%	**0.993**	0	0	0	0	0.0%	7.1%	–	8	7	9	4	14.0%	19.4%	0.983	31	3	11	1	**23.0%**	**37.1%**	0.937
Condiment Box Loading		12	3	3	3	10.5%	31.8%	**0.929**	2	0	0	0	1.0%	4.2%	0.911	15	7	4	4	**15.0%**	**33.2%**	0.917	13	4	3	1	10.5%	27.7%	0.885
Task mean		29.5%	15.4%	14.0%	13.0%	18.0%	36.5%	**0.974**	12.9%	6.3%	5.2%	3.8%	7.1%	23.0%	0.925	35.5%	21.2%	26.7%	25.5%	27.2%	44.2%	0.961	48.5%	18.8%	26.0%	19.8%	**28.3%**	**47.3%**	0.944

is numerically identical to the pooled success rate over 1,300 episodes. The **SR** column is the equal-weight mean of the four channel-specific success rates and introduces no additional evaluation samples. For each task–policy pair, **Eff.** pools the steps to stable success from all successful episodes across the four channels before taking the mean; it is therefore weighted by the number of successes in each channel rather than by the number of nonempty channels. All four policies use executed physics-step counts and recorded task-level expert references.

3.2 Main Results

GR00T N1.5 achieves the highest stable success rate in matched scenes. Under `None`, it completes 631 of 1,300 rollouts (48.5%), followed by $\pi_{0.5}$ with 462 (35.5%), ACT with 383 (29.5%), and DP with 168 (12.9%) (Table 2). GR00T leads strictly on 17 of the 26 settings and $\pi_{0.5}$ on six, with tied leaders on the remaining three. Thus, GR00T's aggregate advantage extends across most evaluated settings, although even its matched-scene SR remains below 50%.

The leading policy changes under combined scene perturbations. Under `Full`, $\pi_{0.5}$ achieves 331 successes (25.5%), ahead of GR00T with 258 (19.8%), ACT with 169 (13.0%), and DP with 50 (3.8%). Its lead over GR00T is 73 successes, or 5.6 percentage points, with strict task-level leads on 16 settings versus seven for GR00T and three all-zero ties. Across all four channels, however, GR00T has the highest mean SR (28.3%), followed by $\pi_{0.5}$ (27.2%), ACT (18.0%), and DP (7.1%). The aggregate ranking therefore depends on the evaluation distribution: matched-scene performance alone does not establish robustness to the sampled combined shifts.

Stage completion reveals partial progress that stable success alone misses. The mean LSCR is 47.3% for GR00T, 44.2% for $\pi_{0.5}$, 36.5% for ACT, and 23.0% for DP, preserving their ordering by mean SR. These values exceed the corresponding SRs by 19.0, 17.0, 18.5, and 15.9 percentage points. LSCR records whether dependency-valid subgoals are ever reached, whereas SR requires the terminal condition to hold for the prescribed dwell time. On Jigsaw Puzzle Assembly, DP and $\pi_{0.5}$ both have 0.0% SR but attain 6.3% and 8.4% LSCR, respectively; GR00T attains 30.3% LSCR with 5.0% SR. LSCR thus distinguishes intermediate progress even when policies rarely or never complete the task.

Efficiency describes execution after conditioning on success and must be read alongside SR. Mean expert-normalized efficiency is 0.974 for ACT, 0.925 for DP, 0.961 for $\pi_{0.5}$, and 0.944 for GR00T, averaged over 25, 20, 23, and 26 tasks with at least one success, respectively. Each task-

level ratio compares the expert reference with the mean physics-step count pooled over successful rollouts from all four channels. For $\pi_{0.5}$, efficiency is 0.929 on Trash Disposal and 1.063 on Frypan Stand & Pour, indicating more and fewer steps than the respective expert references. Stationery Category Sorting, Jigsaw Puzzle Assembly, and Soup Serving yield no successful $\pi_{0.5}$ rollouts; their efficiencies are undefined and excluded from the mean. Because policies succeed on different tasks and episodes, these averages do not establish a controlled speed ranking; they characterize successful execution relative to expert references, not wall-clock inference throughput.

3.3 Generalization Patterns

All policies degrade under combined perturbations, with strongly policy-specific sensitivity. From `None` to `Full`, SR decreases by 16.5, 9.1, 10.1, and 28.7 percentage points for ACT, DP, $\pi_{0.5}$, and GR00T, respectively. The corresponding ratios of aggregate `Full` to `None` SR are 44.1%, 29.8%, 71.6%, and 40.9%. All differences and ratios use success counts before rounding. These ratios compare channel-level aggregates, not the fraction of matched successful episodes that remain successful after perturbation. DP's smaller absolute decrease therefore does not imply greater robustness: its matched-scene SR is only 12.9%, and its ratio is the lowest. In contrast, $\pi_{0.5}$ has both the highest `Full` SR and the highest ratio. Across the 104 task–policy pairs, 90 have lower `Full` success counts, ten are tied, and four have higher counts than `None`. All ten ties correspond to zero successes in both channels, so they offer limited insight into relative robustness.

The isolated perturbation channels reveal different sensitivity profiles across policies. `Equi.` changes task geometry and requires trajectory adaptation, whereas `Inv.` tests robustness to nuisance variation. Relative to `Equi.`, `Inv.` SR is lower by 1.4 and 1.2 percentage points for ACT and DP, but higher by 5.5 and 7.2 points for $\pi_{0.5}$ and GR00T. The latter two policies also achieve higher `Inv.` than `Equi.` success counts on 15 and 16 of the 26 settings, respectively. Thus, geometric adaptation is the more challenging isolated condition for $\pi_{0.5}$ and GR00T under the tested perturbation ranges, whereas ACT and DP show smaller differences in the opposite direction. Separating the two channels exposes these policy-specific sensitivities that a single aggregate robustness score would obscure.

Combined perturbations do not produce a uniform ordering of channel difficulty. For ACT and DP, aggregate SR decreases from `None` through `Equi.` and `Inv.` to `Full`. For $\pi_{0.5}$ and GR00T, `None` remains highest and `Inv.` is second, but `Full` exceeds `Equi.` by 4.3 and 1.1 percentage points, respectively. These differences characterize performance under the sampled channel distributions. As specified in Section 3.1, `Full` draws an independent full-scene sample without a matched anchor; it is therefore not a paired extension of either isolated condition. The observed ordering does not establish that additional perturbations improve performance or that the two factor groups compensate for one another.

3.4 Task-Level Heterogeneity

Stable success varies across tasks under combined perturbations. To examine the task variation behind aggregate SR, we focus on `Full`, which combines both perturbation groups (Table 2). All four policies record nonzero success on 12 of the 26 settings, all record zero success on three, and the remaining 11 show mixed outcomes. Microwave Bowl Loading has the largest total across policies, with 97 successes from 200 rollouts: 29, 4, 31, and 33 for ACT, DP, $\pi_{0.5}$, and GR00T, respectively. By comparison, Stationery Category Sorting, Jigsaw Puzzle Assembly, and Soup Serving yield no stable successes within the 50 rollouts per policy. Even within UR5+RH5DG2, the across-policy total is 88 for Citrus Plate Loading but five for Breadbasket Fast-Food Loading.

Policy rankings reverse across tasks on the same embodiment. On UR5+RH56DFX, $\pi_{0.5}$ records 19 `Full` successes on Fruit Bowl Loading versus five for GR00T, whereas GR00T leads on Screwdriver Box & Hammer with 21 successes versus 16 for $\pi_{0.5}$. These comparisons describe the evaluated task–embodiment settings under the reported training configurations. As task and embodiment configurations are not fully orthogonalized, we refrain from definitive cross-embodiment generalization claims.

3.5 Tactile Policy Comparison

Table 3 compares the non-tactile GR00T N1.5 baseline in Table 2 with its tactile-conditioned variant. We encode the tactile depth maps with a shared CNN and fuse the resulting tokens into GR00T's action features through residual cross-attention after the diffusion transformer (Appendix H). The

Table 3: **GR00T N1.5 success rates with and without tactile input.** Each entry is the mean stable-success rate across four generalization channels, with 50 rollouts per channel and 200 rollouts per task."w/o" and "w/" denote without and with tactile input, respectively. Δ is the success rate with tactile input minus that without tactile input, in percentage points (pp). Tasks with improved success rates are shown on the left, and tasks with decreased success rates are shown on the right. Positive changes are shown in bold red and negative changes in bold green. The shaded final row reports the equal-weight average across all 26 tasks.

Improved with tactile				**Declined with tactile**			
Task	**w/o**	**w/**	Δ	**Task**	**w/o**	**w/**	Δ
Fruit Bowl Loading	32.5%	41.5%	**+9.0**	Wine Glass Plate Balance	28.5%	28.0%	**−0.5**
Cleaner Box Loading	47.0%	55.5%	**+8.5**	Fridge Fruit Shelf Sorting	1.0%	0.5%	**−0.5**
Bimanual Piano Melody	23.0%	30.0%	**+7.0**	Toilet Lid Cleaner Pour	45.0%	44.0%	**−1.0**
Tool Box Loading	40.0%	46.5%	**+6.5**	Jigsaw Puzzle Assembly	5.0%	4.0%	**−1.0**
Baking Tray Prep	10.0%	16.5%	**+6.5**	Shoebox Accessory Pack	40.5%	39.0%	**−1.5**
Medicine Shoebox Pack	13.0%	19.5%	**+6.5**	Soup Serving	3.0%	1.5%	**−1.5**
Citrus Plate Loading	59.0%	61.5%	**+2.5**	Gaming Desk Setup	59.5%	58.0%	**−1.5**
Ball Box Loading	23.0%	25.5%	**+2.5**	Screwdriver Box & Hammer	48.0%	41.0%	**−7.0**
Sports Ball Cup Sort	9.0%	11.0%	**+2.0**	Faucet Cup Water Fill	33.0%	26.0%	**−7.0**
Canned Food Tray Arrangement	12.5%	14.0%	**+1.5**	Frypan Stand & Pour	54.5%	42.0%	**−12.5**
Breadbasket Fast-Food Loading	9.0%	10.5%	**+1.5**	Microwave Bowl Loading	70.5%	57.0%	**−13.5**
Condiment Box Loading	10.5%	11.5%	**+1.0**	Fridge Wine Interhand Pour	21.0%	5.5%	**−15.5**
Stationery Category Sorting	3.0%	3.5%	**+0.5**	Trash Disposal	34.5%	15.5%	**−19.0**
Overall average (all 26 tasks)					28.3%	27.3%	**−1.0**

tactile variant improves mean SR on 13 of the 26 tasks and reduces it on the other 13. The largest gains occur on Fruit Bowl Loading (+9.0 percentage points), Cleaner Box Loading (+8.5), and Bimanual Piano Melody (+7.0), whereas Trash Disposal and Fridge Wine Interhand Pour decline by 19.0 and 15.5 points, respectively. These mixed task-level outcomes may reflect differences in contact sensitivity across task types, the still-underexplored integration of unified simulated tactile inputs into base policies, and the influence of tactile data quality. Our benchmark provides a platform and preliminary insights for further research on tactile-conditioned manipulation.

4 Conclusion

We introduced **Bench2Dex**, a simulation benchmark for bimanual dexterous manipulation spanning 12 dexterous hand models, 26 long-horizon tasks, and approximately 1.3K human-teleoperated demonstrations. Bench2Dex integrates teleoperation, synchronized multimodal data collection, a **unified visuo-tactile interface**, and executable evaluation of task completion, stage-level progress, and success-conditioned efficiency within a common framework. Evaluations of ACT, Diffusion Policy, $\pi_{0.5}$, and GR00T N1.5 using visual and proprioceptive observations show lower aggregate success under combined scene perturbations than under matched-scene conditions, with performance varying across tasks and policies. We further compare GR00T N1.5 with and without tactile input. Tactile input improves performance on half of the evaluated tasks but does not increase the overall task-averaged success rate, showing that its effect is task dependent rather than uniformly beneficial. Together, these results provide reference points for future studies of bimanual manipulation, robustness, and tactile-conditioned policy learning. Bench2Dex provides a common starting point for investigating when tactile information is useful and how manipulation policies can effectively incorporate it across diverse dexterous-hand settings.

Limitations and Outlook. Dexterous-hand and vision-based tactile sensor designs continue to evolve, and the simulated hand and tactile models in Bench2Dex have not been calibrated against corresponding physical hardware. Our current focus is to provide a unified simulation framework for studying bimanual visuo-tactile manipulation under explicit sensing and modeling assumptions, rather than to reproduce the physical response of a specific hand–sensor system. Future work could incorporate more accurate hand and tactile models, hardware-specific calibration, and real-world validation to better understand which findings transfer from simulation to physical manipulation.

AI USE STATEMENT

We used generative AI tools (GPT) for language polishing and grammar checking during manuscript preparation. Generative AI was not used to generate research ideas, conduct experiments, analyze results, or make scientific conclusions. All AI-assisted edits were reviewed and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, results, and artifacts.

ETHICS STATEMENT

This work focuses on simulated robotic manipulation and does not involve human-subject studies or the collection of personal or sensitive data. Human operators were involved only in teleoperating simulated robots for demonstration collection. We do not identify any significant ethical concerns arising from this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the benchmark, tasks, data collection procedure, visuo-tactile interface, evaluation metrics, perturbation settings, and experimental protocols in the main paper and appendix. To support reproducibility and future research, we will publicly release the benchmark environments, task definitions, teleoperated demonstrations, evaluation code, and related resources upon completion of the review process.

REFERENCES

[1] AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, Jianlan Luo, Ping Luo, Yao Mu, Yuehan Niu, Yixuan Pan, Jiangmiao Pang, Yu Qiao, Guanghui Ren, Cheng Ruan, Jiaqi Shan, Yongjian Shen, Chengshi Shi, Mingkang Shi, Modi Shi, Chonghao Sima, Jianheng Song, Huijie Wang, Wenhao Wang, Dafeng Wei, Chengen Xie, Guo Xu, Junchi Yan, Cunbiao Yang, Lei Yang, Shukai Yang, Maoqing Yao, Jia Zeng, Chi Zhang, Qinglin Zhang, Bin Zhao, Chengyue Zhao, Jiaqi Zhao, and Jianchao Zhu. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems, 2025. URL https://arxiv.org/abs/2503.06669.

[2] Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects, 2023. URL https://arxiv.org/abs/2305.05706.

[3] Raunaq Bhirangi, Venkatesh Pattabiraman, Enes Erciyes, Yifeng Cao, Tess Hellebrekers, and Lerrel Pinto. Anyskin: Plug-and-play skin sensing for robotic touch, 2024. URL https://arxiv.org/abs/2409.08276.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

[7] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. Dexycb: A benchmark for capturing hand grasping of objects, 2021. URL https://arxiv.org/abs/2104.04631.

[8] 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, 2025. URL https://arxiv.org/abs/2506.18088.

[9] Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, Weijie Wan, Baijun Chen, Songling Liu, Haowen Yan, Honghao Su, Zhiyang Dou, Kaixuan Wang, Dandan Zhang, Yunze Liu, Yan Qin, Qiwei Liang, Qiwei Wu, Zijian Lin, Wenwei Lin, Yuran Wang, Minghua He, Tianshu Wu, Ruihai Wu, Jingquan Zhou, Kai-Chong Lei, Haibao Yu, Yuanfeng Ji, Weiyang Jin, Guanyu Lin, Xiaofan Li, Qi Xiong, Renjing Xu, Zhongyu Li, Wenhao Chai, Enze Xie, Ziwei Wang, Yao Mu, Hao Dong, Wojciech Matusik, Mingyu Ding, Wenbo Ding, Ping Luo, and Masayoshi Tomizuka. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies, 2026. URL https://arxiv.org/abs/2607.04434.

[10] Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuang Jiang, Stephen Marcus McAleer, Yiran Geng, Hao Dong, Zongqing Lu, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning, 2022. URL https://arxiv.org/abs/2206.08686.

[11] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

[12] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots, 2024. URL https://arxiv.org/abs/2402.10329.

[13] Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer

Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto 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, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

[14] Siyuan Dong, Wenzhen Yuan, and Edward H. Adelson. Improved gelsight tactile sensor for measuring geometry and slip. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 137–144. IEEE, September 2017. doi: 10.1109/iros.2017.8202149. URL http://dx.doi.org/10.1109/IROS.2017.8202149.

[15] Siyuan Dong, Devesh K. Jha, Diego Romeres, Sangwoon Kim, Daniel Nikovski, and Alberto Rodriguez. Tactile-rl for insertion: Generalization to objects of unknown geometry, 2021. URL https://arxiv.org/abs/2104.01167.

[16] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot, 2023. URL https://arxiv.org/abs/2307.00595.

[17] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024. URL https://arxiv.org/abs/2401.02117.

[18] Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy, 2025. URL https://arxiv.org/abs/2410.01345.

[19] Abraham George, Selam Gano, Pranav Katragadda, and Amir Barati Farimani. Vital pretraining: Visuo-tactile pretraining for tactile and non-tactile manipulation policies, 2024. URL https://arxiv.org/abs/2403.11898.

[20] Markus Grotz, Mohit Shridhar, Tamim Asfour, and Dieter Fox. Peract2: Benchmarking and learning for robotic bimanual manipulation tasks, 2024. URL https://arxiv.org/abs/2407.00278.

[21] 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, 2023. URL https://arxiv.org/abs/2302.04659.

[22] Peng Hao, Chaofan Zhang, Dingzhe Li, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Tla: Tactile-language-action model for contact-rich manipulation, 2025. URL https://arxiv.org/abs/2503.08548.

[23] Jingbo He, Michael Färber, and Roberto Calandra. RCT: A robot-collected touch-vision-language dataset for tactile generalization, 2026. URL https://arxiv.org/abs/2606.31694.

[24] Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing, 2024. URL https://arxiv.org/abs/2410.24090.

[25] Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video, 2026. URL https://arxiv.org/abs/2505.11709.

[26] Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, Chenyang Gu, Zhuoyang Liu, Nuowei Han, Xiangju Mi, Yaoxu Lv, Yankai Fu, Gaole Dai, Langzhe Gu, Tao Li, Yuheng Zhang, Yixue Zhang, Xinhua Wang, Shichao Fan, Meng Li, Zhen Zhao, Ning Liu, Zhiyuan Xu, Pei Ren, Junjie Ji, Haonan Liu, Kuan Cheng, Shanghang Zhang, and Jian Tang. Robomind 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence, 2026. URL https://arxiv.org/abs/2512.24653.

[27] Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3d-vitac: Learning fine-grained manipulation with visuo-tactile sensing, 2025. URL https://arxiv.org/abs/2410.24091.

[28] Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-vla: Unlocking vision-language-action model's physical knowledge for tactile generalization, 2025. URL https://arxiv.org/abs/2507.09160.

[29] Yuzhe Huang, Pei Lin, Wanlin Li, Daohan Li, Jiajun Li, Jiaming Jiang, Chenxi Xiao, and Ziyuan Jiao. Taf-vla: Tactile-force alignment in vision-language-action models for force-aware manipulation, 2026. URL https://arxiv.org/abs/2601.20321.

[30] Yuzhe Huang, Jiaping Wu, Jiaming Jiang, Hezhe Lin, Aikebaier Aierken, Yunlong Wang, Kun Cheng, Wanlin Li, Chenxi Xiao, Ziyuan Jiao, and Yuanxin Zhong. Ht-bench: Benchmarking and learning dexterous full-hand tactile representations with egocentric vision, 2026. URL https://arxiv.org/abs/2606.19161.

[31] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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, 2025. URL https://arxiv.org/abs/2504.16054.

[32] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment, 2019. URL https://arxiv.org/abs/1909.12271.

[33] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning, 2025. URL https://arxiv.org/abs/2410.24185.

[34] Bowen Jing, Mingxin Wang, Ruiyang Hao, Chenchen Ge, Hanwen Shen, Junjie He, Yang Cui, Yiming Hou, Weitao Zhou, Jiawei Wang, Minglei Li, Dandan Zhang, Ding Zhao, Houde Liu, Xiaofan Li, Si Liu, Ping Luo, and Haibao Yu. Softvtbench: A safety-aware visuo-tactile benchmark for physically constrained robotic manipulation of deformable objects (early version), 2026. URL https://arxiv.org/abs/2607.04234.

[35] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2025. URL https://arxiv.org/abs/2403.12945.

[36] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

[37] Yejin Kim, Wilbert Pumacay, Omar Rayyan, Max Argus, Winson Han, Eli VanderBilt, Jordi Salvador, Abhay Deshpande, Rose Hendrix, Snehal Jauhri, Shuo Liu, Nur Muhammad Mahi Shafiullah, Maya Guru, Ainaz Eftekhar, Karen Farley, Donovan Clay, Jiafei Duan, Arjun Guru, Piper Wolters, Alvaro Herrasti, Ying-Chun Lee, Georgia Chalvatzaki, Yuchen Cui, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmospaces: A large-scale open ecosystem for robot navigation and manipulation, 2026. URL https://arxiv.org/abs/2602.11337.

[38] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, Dinesh Jayaraman, and Roberto Calandra. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5 (3):3838–3845, July 2020. ISSN 2377-3774. doi: 10.1109/lra.2020.2977257. URL http://dx.doi.org/10.1109/LRA.2020.2977257.

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

[40] Jongbin Lim, Taeyun Ha, Mingi Choi, Jisoo Kim, Byungjun Kim, Subin Jeon, and Hanbyul Joo. HRDexDB: A paired human-robot dataset for cross-embodiment dexterous grasping, 2026. URL https://arxiv.org/abs/2604.14944.

[41] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

[42] Xuanyi Liu, Haofeng Wang, Ruiqi Li, Danni Yu, Rui Wan, Ruixu Zhang, Siyu Tao, Xue Yang, Shaofeng Zhang, Zicheng Zhang, Jiaqi Zhang, and Siwei Ma. Triworldbench: A tri-view consistency perspective on embodied world models, 2026. URL https://arxiv.org/abs/2609.26314.

[43] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations, 2023. URL https://arxiv.org/abs/2310.17596.

[44] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks, 2022. URL https://arxiv.org/abs/2112.03227.

[45] Elle Miller, Jayaram Reddy, Ayush Deshmukh, Trevor McInroe, David Abel, Oisin Mac Aodha, and Sethu Vijayakumar. roto 2.0: The robot tactile olympiad, 2026. URL https://arxiv.org/abs/2605.21429.

[46] Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins, 2025. URL https://arxiv.org/abs/2504.13059.

[47] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024. URL https://arxiv.org/abs/2406.02523.

[48] Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots, 2026. URL https://arxiv.org/abs/2603.04356.

[49] Suting Ni, Hanbing Zhang, Zhenyu Wei, Guo Chen, Chixuan Zhang, Ye Shi, and Jingya Wang. Tactidex: A real-world tactile-guided benchmark for human-like dexterous manipulation, 2026. URL https://arxiv.org/abs/2607.09190.

[50] NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

[51] NVIDIA, :, Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, Mihir Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, Hammad Mazhar, Masoud Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander Poddubny, Nathan Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, Ruchik Thaker, Matthew Trepte, Karl Van Wyk, Fangzhou Yu, Alex Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, Ashwin Varghese Kuruttukulam, Michael A. Lin, Ajay Mandlekar, Karsten Patzwaldt, John Welsh, Huihua Zhao, Fatima Anes, Jean-Francois Lafleche, Nicolas Moënne-Loccoz, Soowan Park, Rob Stepinski, Dirk Van Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, Pablo de Heras Ciechomski, Gilles Daviet, Mohammad Mohajerani, Julia von Muralt, Viktor Reutskyy, Michael Sauter,

Simon Schirm, Eric L. Shi, Pierre Terdiman, Kenny Vilella, Tobias Widmer, Gordon Yeoman, Tiffany Chen, Sergey Grizan, Cathy Li, Lotus Li, Connor Smith, Rafael Wiltz, Kostas Alexis, Yan Chang, David Chu, Linxi "Jim" Fan, Farbod Farshidian, Ankur Handa, Spencer Huang, Marco Hutter, Yashraj Narang, Soha Pouya, Shiwei Sheng, Yuke Zhu, Miles Macklin, Adam Moravanszky, Philipp Reist, Yunrong Guo, David Hoeller, and Gavriel State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning, 2025. URL https://arxiv.org/abs/2511.04831.

[52] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation, 2019. URL https://arxiv.org/abs/1808.00177.

[53] Xingyu Peng, Chen Gao, Liankai Jin, Annan Li, and Si Liu. Bicoord: A bimanual manipulation benchmark towards long-horizon spatial-temporal coordination, 2026. URL https://arxiv.org/abs/2604.05831.

[54] Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation, 2024. URL https://arxiv.org/abs/2402.08191.

[55] Yuzhe Qin, Hao Su, and Xiaolong Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation, 2023. URL https://arxiv.org/abs/2204.12490.

[56] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system, 2024. URL https://arxiv.org/abs/2307.04577.

[57] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, 2018. URL https://arxiv.org/abs/1709.10087.

[58] Carmelo Sferrazza, Younggyo Seo, Hao Liu, Youngwoon Lee, and Pieter Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning, 2023. URL https://arxiv.org/abs/2311.00924.

[59] Zilin Si and Wenzhen Yuan. Taxim: An example-based simulation model for gelsight tactile sensors, 2021. URL https://arxiv.org/abs/2109.04027.

[60] Lei Su, Zhijie Peng, Renyuan Ren, Shengping Mao, Juan Du, Kaifeng Zhang, and Xuezhou Zhu. Tacmap: Bridging the tactile sim-to-real gap via geometry-consistent penetration depth map, 2026. URL https://arxiv.org/abs/2602.21625.

[61] Cong Tai, Zhaoyu Zheng, Haixu Long, Hansheng Wu, Haodong Xiang, Zhengbin Long, Jun Xiong, Rong Shi, Shizhuang Zhang, Gang Qiu, He Wang, Ruifeng Li, Jun Huang, Bin Chang, Shuai Feng, and Tao Shen. Realmirror: A comprehensive, open-source vision-language-action platform for embodied ai, 2025. URL https://arxiv.org/abs/2509.14687.

[62] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai, 2025. URL https://arxiv.org/abs/2410.00425.

[63] ALOHA 2 Team, Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, Wayne Gramlich, Torr Hage, Alexander Herzog, Jonathan Hoech, Thinh Nguyen, Ian Storz, Baruch Tabanpour, Leila Takayama, Jonathan Tompson, Ayzaan Wahid, Ted Wahrburg, Sichun Xu, Sergey Yaroshenko, Kevin Zakka, and Tony Z. Zhao. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation, 2024. URL https://arxiv.org/abs/2405.02292.

[64] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

[65] Stephen Tian, Frederik Ebert, Dinesh Jayaraman, Mayur Mudigonda, Chelsea Finn, Roberto Calandra, and Sergey Levine. Manipulation by feel: Touch-based control with deep predictive models, 2019. URL https://arxiv.org/abs/1903.04128.

[66] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024. URL https://arxiv.org/abs/2308.12952.

[67] Hanwen Wang, Weizhi Zhao, Xiangyu Wang, Siyuan Huang, He Lin, Boyuan Zheng, Rongtao Xu, Gang Wang, Yao Mu, He Wang, Lue Fan, Hongsheng Li, Zhaoxiang Zhang, and Tieniu Tan. Dexjoco: A benchmark and toolkit for task-oriented dexterous manipulation on mujoco, 2026. URL https://arxiv.org/abs/2605.16257.

[68] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation, 2023. URL https://arxiv.org/abs/2210.02697.

[69] Shaoxiong Wang, Mike Lambeta, Po-Wei Chou, and Roberto Calandra. Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):3930–3937, April 2022. ISSN 2377-3774. doi: 10.1109/lra.2022.3146945. URL http://dx.doi.org/10.1109/LRA.2022.3146945.

[70] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, Shichao Fan, Xinhua Wang, Fei Liao, Zhen Zhao, Guangyu Li, Zhao Jin, Lecheng Wang, Jilei Mao, Ning Liu, Pei Ren, Qiang Zhang, Yaoxu Lyu, Mengzhen Liu, He Jingyang, Yulin Luo, Zeyu Gao, Chenxuan Li, Chenyang Gu, Yankai Fu, Di Wu, Xingyu Wang, Sixiang Chen, Zhenyu Wang, Pengju An, Siyuan Qian, Shanghang Zhang, and Jian Tang. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems XXI*, RSS2025. Robotics: Science and Systems Foundation, 2025. doi: 10.15607/rss.2025.xxi.152. URL http://dx.doi.org/10.15607/RSS.2025.XXI.152.

[71] Yansong Wu, Zongxie Chen, Fan Wu, Lingyun Chen, Liding Zhang, Zhenshan Bing, Abdalla Swikir, Sami Haddadin, and Alois Knoll. Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation, 2025. URL https://arxiv.org/abs/2409.11047.

[72] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, Tengyu Liu, Li Yi, and He Wang. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy, 2023. URL https://arxiv.org/abs/2303.00938.

[73] Zhengtong Xu and Yu She. Letac-mpc: Learning model predictive control for tactile-reactive grasping, 2024. URL https://arxiv.org/abs/2403.04934.

[74] Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation, 2025. URL https://arxiv.org/abs/2503.02881.

[75] Zhenjie Yang, Xingyu Jiao, Guopeng Zhong, Shuzhe Yang, Shi Che, Chao Wu, Chenyu Jiang, Dongjie Zhang, Yideng Zhang, Zheng Zhang, Muyun Jiang, Haisheng Su, Shuang Jin, Donghang Zhang, Chao Yang, Li Chen, Hongyang Li, Zuxuan Wu, Yu-Gang Jiang, Xiaosong Jia, and Junchi Yan. Handedit: A unified benchmark for egocentric human-to-robot dexterous hand image editing, 2026. URL https://arxiv.org/abs/2608.12122.

[76] Yunchao Yao, Zhuxiu Xu, Tianqi Zhang, Zixian Liu, Sikai Li, Zhenyu Wei, Feng Chen, Dihong Huang, Kechang Wan, Chenyang Ma, Shuqi Zhao, Shenghua Gao, Masayoshi Tomizuka, Yi Ma, and Mingyu Ding. Dexverse: A modular benchmark for multi-task, multi-embodiment dexterous manipulation, 2026. URL https://arxiv.org/abs/2607.08751.

[77] Jianglong Ye, Keyi Wang, Chengjing Yuan, Ruihan Yang, Yiquan Li, Jiyue Zhu, Yuzhe Qin, Xueyan Zou, and Xiaolong Wang. Dex1b: Learning with 1b demonstrations for dexterous manipulation, 2025. URL https://arxiv.org/abs/2506.17198.

[78] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2021. URL https://arxiv.org/abs/1910.10897.

[79] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017. ISSN 1424-8220. doi: 10.3390/s17122762. URL https://www.mdpi.com/1424-8220/17/12/2762.

[80] Jonathan Zamora, Daniel Seita, and Yue Wang. Mujoco manipulus: A robot learning benchmark for generalizable tool manipulation, 2025. URL https://openreview.net/forum?id=b9Ne5lHJ8Y.

[81] Yuan Zeng, Yujia Shi, Tiao Tan, Xingting Li, Yaqi Qin, Zongqing Lu, Wenming Yang, Jing-Hao Xue, and Qingmin Liao. EgoTactile: Learning grasp pressure for everyday objects from egocentric video, 2026. URL https://arxiv.org/abs/2606.09243.

[82] Jialiang Zhang, Haoran Liu, Danshi Li, Xinqiang Yu, Haoran Geng, Yufei Ding, Jiayi Chen, and He Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes, 2024. URL https://arxiv.org/abs/2410.23004.

[83] Kaidi Zhang, Heng Zhang, Zhengtong Xu, Zhiyuan Zhang, Md Rakibul Islam Prince, Xiang Li, Xiaojing Han, Yuhao Zhou, Arash Ajoudani, and Yu She. Tacvla: Contact-aware tactile fusion for robust vision-language-action manipulation, 2026. URL https://arxiv.org/abs/2603.12665.

[84] Shuqi Zhao, Xinghao Zhu, Yuxin Chen, Chenran Li, Lichen Xie, Xiang Zhang, Mingyu Ding, and Masayoshi Tomizuka. Dexh2r: Task-oriented dexterous manipulation from human to robots, 2026. URL https://arxiv.org/abs/2411.04428.

[85] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[86] Zhaxizhuoma, Kehui Liu, Chuyue Guan, Zhongjie Jia, Ziniu Wu, Xin Liu, Tianyu Wang, Shuai Liang, Pengan Chen, Pingrui Zhang, Haoming Song, Delin Qu, Dong Wang, Zhigang Wang, Nieqing Cao, Yan Ding, Bin Zhao, and Xuelong Li. Fastumi: A scalable and hardware-independent universal manipulation interface with dataset, 2025. URL https://arxiv.org/abs/2409.19499.

[87] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning, 2025. URL https://arxiv.org/abs/2009.12293.

[88] Zijie Zhu, Weiren Cai, Yizhou Wang, Zhenjie Yang, Yide Liu, Jiahao Chen, and Guanqi He. Mint: A unified model for world-space camera and hand motion estimation from scalable egocentric pipeline supervision. *arXiv preprint arXiv:2609.04958*, 2026.

A Related Work

A.1 Manipulation Benchmarks and Datasets

Simulation benchmarks such as Meta-World [78], RLBench [32], robosuite [87], CALVIN [44], LIBERO [41], and ManiSkill3 [62] provide standardized evaluation for multi-task manipulation, while MimicGen [43] and RoboTwin [46] reduce demonstration cost through data generation. Large-scale real-world datasets—BridgeData V2 [66], Open X-Embodiment [13], DROID [35], RoboMIND 2.0 [26], and AgiBot World [1]—have fueled generalist policies such as Octo [64], OpenVLA [36], and π_0 [4]. Recent robustness benchmarks—RoboCasa [47], RoboCasa365 [48], GemBench [18], THE COLOSSEUM [54], RoboTwin 2.0 [8], RLBench2 [20], and MuJoCo Manipulus [80]—test controlled distribution shifts. This line of work relies on parallel-jaw grippers, so it provides no multi-finger dexterous hand or vision-based tactile support, and its evaluation protocols do not compare dexterous hand morphologies or visuo-tactile sensing configurations.

A.2 Dexterous and Bimanual Manipulation Benchmarks

A separate line of benchmarks targets dexterous or bimanual manipulation specifically, but each covers only part of the properties in Table 1. Bi-DexHands [10] focuses on reinforcement learning with dual Shadow Hands but provides no visual observations, and BiCoord [53] studies bimanual coordination without dexterous hands. DexMimicGen [33] generates demonstrations for 3 dexterous embodiments across 9 tasks, but does not include tool use or tactile sensing. RealMirror [61] evaluates a single dexterous embodiment across five manipulation tasks, including air-fryer drawer operation and food loading, without tactile sensing. DexJoCo [67] supports 2 embodiments and 11 tasks with tool use, but omits tactile data. DexVerse [76] covers 19 teleoperated tool-use tasks, derived from its publicly released demonstration subset with one dexterous embodiment design, but lacks vision-based tactile sensing. None of these benchmarks combines multiple dexterous embodiments with tactile sensing.

A.3 Dexterous Data Collection and Teleoperation

Foundational work on Adroit [57], OpenAI in-hand manipulation [52], and DexYCB [7] established dexterous hands as challenging platforms for robot learning. Recent efforts scale dexterous data through synthetic grasps [82; 77], generative demonstrations [33], hand-motion reconstruction from egocentric videos [88] and human-to-robot transfer [84; 25; 75]. Teleoperation systems such as Mobile ALOHA [17], ALOHA 2 [63], UMI [12], and FastUMI [86] advance bimanual data collection but use parallel-jaw grippers, while AnyTeleop [56] and From One Hand to Multiple [55] address cross-hand retargeting at the algorithmic level. These methods improve how dexterous demonstrations are generated or collected, but each is evaluated on its own custom setup rather than a shared benchmark spanning multiple embodiments.

A.4 Tactile Sensing and Benchmarking for Manipulation

Vision-based tactile sensors, such as GelSight [79; 14], DIGIT [38], TACTO [69], and Taxim [59], convert local contact deformation into image-like tactile observations, enabling robots to reason about contact states that are difficult to infer from external vision alone. Recent studies have used tactile feedback for insertion, grasp adjustment, and contact-rich manipulation [65; 15; 73], while newer work further explores reusable tactile skins [3], self-supervised touch representations [24], visuo-tactile pretraining and policy learning [58; 19; 27], and tactile-conditioned diffusion or vision-language-action policies [74; 71; 22; 28; 83; 29].

Complementary benchmark efforts have begun to standardize tactile evaluation at different levels. EgoTactile pairs egocentric video with full-hand pressure supervision, RCT evaluates contact-sequence-aware generalization across materials and sensors, HT-Bench targets full-hand tactile representation learning over 226 tasks, and HRDexDB aligns human and robot grasp sequences for cross-embodiment study [81; 23; 30; 40]. At the closed-loop policy level, roto 2.0 evaluates tactile-only reinforcement learning across four dexterous morphologies, TactiDex measures physically grounded contact in real-world dexterous manipulation, and SoftVTBench introduces goal- and safety-aware evaluation for visuo-tactile deformable-object manipulation [45; 49; 34]. Among the benchmarks in Table 1, RoboMIND 2.0 [26] includes dexterous-hand demonstrations as well as tactile force measurements collected with parallel-jaw grippers on an AgileX platform. Its reported

tactile observations encode forces and directions rather than vision-based tactile images. These efforts address complementary slices of tactile learning, but do not jointly target a common tactile representation spanning heterogeneous bimanual dexterous embodiments and long-horizon tasks.

Bench2Dex instead focuses on a unified cross-embodiment tactile representation by reconstructing tactile contact surfaces from different hand meshes and converting contact depth into a common surface-aligned tactile-map format, building on geometry-consistent penetration-depth encoding [60]. This enables consistent tactile data acquisition and evaluation across diverse dexterous embodiments.

As summarized in Table 1, no existing benchmark jointly supports diverse bimanual dexterous embodiments, teleoperated demonstration collection, vision-based tactile sensing, tool use, and articulated-object interaction. Bench2Dex fills this gap with a single simulation pipeline that unifies multiple bimanual dexterous embodiments under one teleoperation interface, records synchronized multimodal observations including tactile data, and evaluates policies under controlled distribution shifts.

B Full Task Catalog

Table 4 lists the tasks. For each task we report the task identifier and name, a short English description, the success condition used by the evaluator, the robot embodiment, and a scene snapshot.

Table 4: Full Bench2Dex task catalog: identifier, description, success condition, robot embodiment, and scene snapshot.

Task	Description	Success Condition	Embodiment	Scene
Wine Glass Plate Balance	Carry three filled wine glasses to the plates of three diners without spilling or knocking over any objects.	All three wine glasses are kept upright (within 30° of vertical), each placed within 0.10 m of one of the target positions, and fully at rest.	xArm7+ Ability	
Fruit Bowl Loading	Move the bowl to the center of the table and place two apples and a banana into it.	The bowl is moved into the central zone of the table and kept upright, and two apples and one banana are placed inside it, all at rest.	UR5+ RH56DFX	
Citrus Plate Loading	Place two lemons and two oranges onto the plate.	Two lemons and two oranges are placed on the plate, with each object's center within 0.15 m of the plate interior.	UR5+ RH5DG2	
Frypan Stand & Pour	Place the frypan onto the display stand, then pour the soy sauce and olive oil into the frypan one by one.	The frypan is placed on the stand; the soy sauce and olive oil are each tilted at least 50° with their mouth over the frypan to pour, then both bottles are returned upright and at rest on the table while the bread remains in the pan.	UR5+Schunk	
Cleaner Box Loading	Use both hands to lift the cleaner upright into the wooden box, then place the soap into the box.	The wooden box is upright, the cleaner is inside the box and upright (not tilted), and the soap is inside the box.	xArm7+ LEAP	
Screwdriver Box & Hammer	Place the right screwdriver into the box and move the box left, place the left screwdriver into the box, then strike the wooden block once with the hammer and place the hammer into the box.	Both screwdrivers are inside the upright box, the hammer strikes the wooden block once (swing near the block with a rigid-body response), and the hammer is then placed inside the box.	UR5+ RH56DFX	

Continued on next page

Table 4: Full Bench2Dex task catalog (continued).

Task	Description	Success Condition	Embodiment	Scene
Condiment Box Loading	Place the monosodium glutamate, soy sauce, and vinegar into the wooden box.	The box is upright; the MSG, soy sauce, and vinegar are all inside the box, with the soy sauce and vinegar kept upright, and all objects at rest.	UR5+Wuji	
Tool Box Loading	Use the right hand to place the drill and flat screwdriver and the left hand to place the wrench and Phillips screwdriver into the wooden box.	All four tools (drill, flat screwdriver, adjustable wrench, Phillips screwdriver) are placed inside the upright wooden box.	Panda+Orca	
Stationery Category Sorting	Use the right hand to sort the marker and battery into the pen cup and plastic box, and the left hand to sort the gray pen and glue into the respective containers.	The pen and marker are placed upright in the pen cup and the glue and battery are placed in the plastic box, with both containers upright and all objects at rest.	RM65+Revo2	
Canned Food Tray Arrange-ment	Use the left hand to place the master chef can and MSG and the right hand to place the milk box and potted meat can onto the tray.	The master chef can, MSG, milk box, and potted meat can are all placed on the tray and kept upright, with the tray upright and all objects at rest.	IIWA7+ Sharpa	
Ball Box Loading	Place the mini soccer ball, tennis ball, golf ball, and ping-pong ball into the box.	The box is upright and all four balls are placed inside it, each within 0.22 m of the box interior.	UR5+Wuji	
Baking Tray Prep	Use the left hand to place the brush and spatula and the right hand to place the small pudding box and large gelatin box onto the tray.	The brush, spatula, pudding box, and gelatin box are all placed on the tray, with the tray kept upright and all objects at rest.	IIWA7+ Sharpa	
Fridge Wine Interhand Pour	Open the refrigerator, take out the wine bottle, hand it to the right hand in the air to pour a glass of wine, return it to the left hand to put back into the fridge, and close the door.	The fridge door is opened, the wine bottle is lifted out, tilted at least 50° toward the glass to pour, then returned upright inside the fridge, and the door is closed.	UR5+ RH5DG2	
Trash Disposal	Open the trash can lid, throw the crumpled paper into the trash can, use the left hand to throw the bottle and banana peel into the trash can, then close the lid.	The trash can is upright and the crumpled paper, bottle, and banana peel are all placed inside it, at rest.	UR5+ RH56DFX	
Fridge Fruit Shelf Sorting	Move the banana to the left side of the desk, open both fridge doors, place the lemon on the upper shelf and the banana on the lower shelf, then close the door.	The banana and lemon are placed inside the fridge, both fridge doors are closed, and both objects are at rest.	UR5+Shadow	
Microwave Bowl Loading	Open the microwave door, bring the bowl to the microwave, place the baguette into the bowl, place the bowl into the microwave, and close the door.	The bowl is inside the microwave cavity and upright, the baguette is placed in the bowl, and the microwave door is closed.	UR5+Schunk	
Toilet Lid Cleaner Pour	Open the toilet lid, pour the cleaner into the toilet, and close the toilet lid.	The toilet lid is opened, the cleaner is lifted and tilted at least 50° toward the toilet bowl to pour while the lid stays open, and the lid is then closed.	xArm7+ LEAP	

Continued on next page

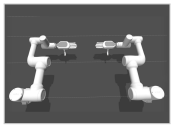
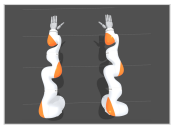
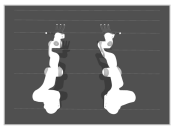
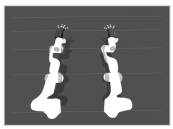
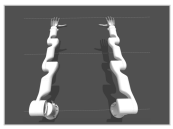
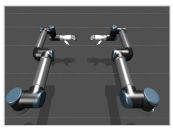
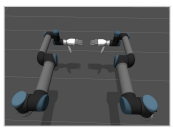
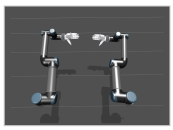
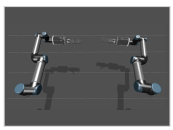
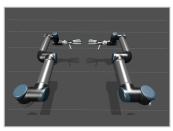
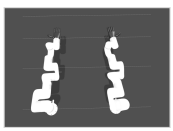
Table 4: Full Bench2Dex task catalog (continued).

Task	Description	Success Condition	Embodiment	Scene
Breadbasket Fast-Food Loading	Place the baguette and bread into the bread basket, move the basket to the center of the table, then place the hamburger and french fries into the basket.	The bread basket is upright and the french fries, hamburger, bread, and baguette are all placed inside it.	UR5+ RH5DG2	
Medicine Shoebox Pack	Move the shoe box to the center of the table, then place the pill bottle, toothpaste, and hydrating oil into the box.	The shoe box is upright and the pill bottle, toothpaste, and hydrating oil are all placed inside it.	Panda+Orca	
Shoebox Accessory Pack	Place the seal into the shoe box and move the box to the center of the table, then place the shoe and pet collar into the box.	The shoe box is upright and the seal, shoe, and pet collar are all placed inside it, at rest.	xArm7+ LEAP	
Sports Ball Cup Sort	Place the tennis ball and baseball into the large cup and the racquetball and golf ball into the small cup.	Both cups are upright; the tennis ball and baseball are inside the large cup and the racquetball and golf ball are inside the small cup, all at rest.	Panda+ Allegro	
Faucet Cup Water Fill	Place the spoon into the mug, place the mug under the faucet, open the faucet to fill the mug and then close it, and place the mug onto the tray.	The spoon is placed in the upright mug; while the mug remains under the faucet opening, the faucet is opened and then closed again; and the mug is placed at rest on the tray.	RM65+Revo2	
Jigsaw Puzzle Assembly	Use the right and left hands in turn to assemble the green, red, blue, and yellow puzzle pieces onto the fixed white puzzle piece in the center of the table, forming a rectangle.	All four colored puzzle pieces are placed at their target positions around the fixed white piece (within 0.02 m in xy and 0.01 m in height of the reference), and all pieces are at rest.	IIWA7+ Sharpa	
Soup Serving	Hold the bowl beside the pot with the left hand, ladle two scoops of soup from the pot into the bowl with the right hand, carry the bowl to the front-right area, and return the ladle.	The pot is on the stove, the bowl is held near the pot, the ladle dips into the pot and is brought over the bowl twice, the bowl is delivered to the front-right serving zone, and the ladle is returned.	UR5+Shadow	
Bimanual Piano Melody	Both hands play the piano key sequence C-C-G-G-A-A-G together, the left hand playing the bass notes and the right hand the treble notes.	The left hand plays the bass melody and the right hand plays the treble melody C-C-G-G-A-A-G, with each note key pressed in sequence.	xArm7+ Ability	
Gaming Desk Setup	Straighten the monitor screen forward with both hands, press the ESC key, move the mouse back to the left side of the keyboard, and click the left mouse button.	The monitor is straightened forward, the ESC key is pressed once, the mouse is returned to the left of the keyboard, and the left mouse button is clicked.	JAKA ZU7+ DexHand021	

C Supported Bimanual Robot Embodiments

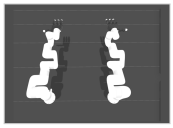
This section summarizes the 12 bimanual robot embodiments supported by Bench2Dex. Each embodiment combines a robot arm and a dexterous hand, enabling evaluation across different arm kinematics, hand morphologies, palm geometries, and finger configurations.

Table 5: Supported bimanual robot embodiments in Bench2Dex.

Embodiment	Render	Description
JAKA ZU7+DexHand021		A metallic-gray collaborative arm with blue circular joint caps, paired with a silver dexterous hand with dense exposed mechanisms, a ribbed palm surface, and slender articulated fingers. The transparent wrist housing and mechanical finger structure make it distinctive.
IIWA7+Sharpa		A white KUKA-style arm with bright orange accent panels, paired with a smooth anthropomorphic hand. The clean enclosed hand silhouette contrasts with the strong white-orange arm styling.
Panda+Allegro		A light Panda-style arm paired with a compact dark modular hand. The boxy palm, rectangular segmented fingers, and bright fingertip caps create a sharp contrast with the soft industrial arm.
Panda+Orca		A light Panda-style arm attached to a dark hand with a narrow palm and strongly spread fingers. Long separated digits with bright caps create an open fan-like silhouette.
RM65+Revo2		A smooth white arm paired with a compact anthropomorphic hand with a rounded palm shell, slim fingers, and a clean enclosed structure. The embodiment appears polished and tightly integrated.
UR5+RH56DFX		A metallic-gray industrial arm with rounded cylindrical links and blue joint caps, ending in a white-and-silver dexterous hand with a sculpted palm shell, dark finger pads, and a large side thumb.
UR5+RH5DG2		A metallic-gray UR5-style arm with blue circular joint caps, attached to a white-and-gray hand with a rounded palm shell, dark cylindrical finger coverings, and a thick side thumb close to the palm plane.
UR5+Schunk		A metallic-gray arm with blue round joint covers, paired with a robust white industrial hand. A broad palm, thick segmented fingers, gray fingertip pads, and heavy thumb create a sturdy precision-oriented look.
UR5+Shadow		A metallic-gray arm with blue circular joints, attached to a dark anthropomorphic hand with slim multi-joint fingers and bright fingertip caps. The hand is lightweight and human-like relative to the heavier arm.
UR5+Wuji		A metallic-gray arm with blue circular joint caps, ending in a slim dexterous hand with long narrow fingers and a thin side thumb. The hand appears lighter and more elongated than other UR5-based embodiments.
xArm7+Ability		A white arm with smooth enclosed links, paired with a compact white hand with a rounded palm shell, dark finger coverings, and a thick side thumb. The embodiment is soft-contoured and highly enclosed.

Continued on next page

Table 5: Supported bimanual robot embodiments in Bench2Dex (continued).

Embodiment	Render	Description
xArm7+LEAP		A white arm with smooth rounded links, attached to a compact low-profile robotic hand with a blocky palm and modular fingers. Stacked box-like segments and bright caps give it a clean geometric appearance.

D UNIFIED TACTILE SURFACE RECONSTRUCTION

Bench2Dex reconstructs tactile contact surfaces for the twelve bimanual hand embodiments included in the benchmark. The representation builds on image-based tactile simulation [69; 59] and the geometry-consistent penetration-depth formulation of TacMap [60], while extending the surface-map abstraction to heterogeneous hand morphologies. For each hand, we identify the physical regions that should generate tactile evidence—elastomer pads, rubber pads, fingertip shells, force-sensor covers, or dedicated touch links—and rebuild or clean them in Blender. The cleaned meshes are then converted into surface point-and-normal assets so that the tactile signal is defined on the intended contact surface rather than on arbitrary full-link geometry. The reconstruction follows a common convention across embodiments. Each surface mesh is expressed in the tactile sensor attach-link frame, or converted from the original URDF visual or collision geometry using the corresponding geometry origin, rotation, and mesh scale. The mesh is rasterized into a regular surface-aligned tactile grid, producing point and inward-facing normal maps at the native resolution used by the tactile registry, so that runtime rays are cast from the nominal contact surface toward the finger interior. The generated points are stored in millimeters and converted back to meters during sensing, keeping offline asset generation and runtime ray-casting consistent.

Different hands expose different mesh sharing patterns: some use a shared non-thumb surface for all non-thumb fingers, while others require independent assets per finger because the pad geometry varies. Table 6 summarizes the resulting grouping policy. The current 12-hand setup uses a four-finger-shared grouping for nine hands and a per-finger independent grouping for three hands. Each hand places one tactile site on every finger, so the ten five-finger embodiments carry ten sites each, while the four-finger Allegro and LEAP embodiments carry eight; palm sites are excluded. In the released data, each site is sampled at the native surface-map resolution of 240×240 (subsampling step 1) and captured once per simulation step during replay.

Table 6: Tactile surface-map grouping policy for the twelve Bench2Dex hands. "4F" denotes a shared non-thumb surface group.

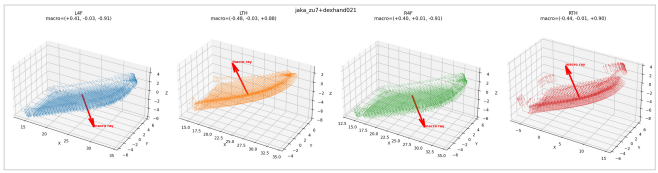
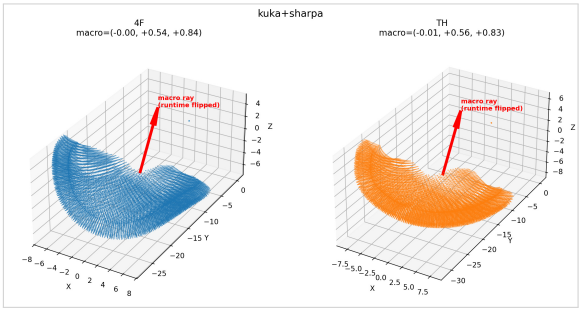
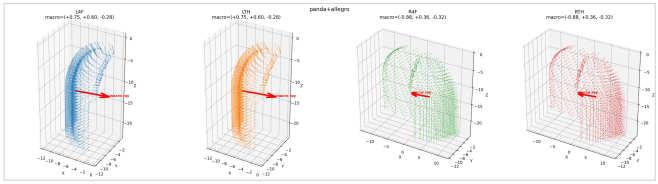
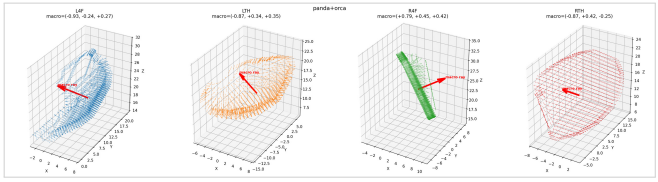
Grouping policy	Hands	Surface-map asset groups
Legacy side-shared thumb/4F	Sharpa	Two assets are shared across both hands: TH for thumbs and 4F for all non-thumb fingers. This preserves the original Sharpa TacMap asset convention.
Side-specific thumb/4F shared	RH5DG2, Shadow, Schunk, Wuji, Allegro, Orca, LEAP, DexHand021	Four assets are used per embodiment: RTH, R4F, LTH, and L4F. The non-thumb group uses one representative surface per side. Allegro and LEAP are four-finger hands, so 4F covers index, middle, and ring.
Per-finger independent	RH56DFX, Ability, Revo2	Each finger has its own surface-map group on each side, such as RTH, RIDX, RMID, RRING, RLIT and the corresponding left-hand groups. This is used when non-thumb pad meshes are not safely interchangeable.

Table 8: Tactile contact-surface mesh checklist used for Blender reconstruction. The mesh column lists only source filenames rather than full dataset paths.

Hand	Grouping	Source links for reconstruction	Mesh source used in Blender	Notes
DexHand021	Side-specific thumb/4F	Thumb: r_f_link1_4 and l_f_link1_4. 4F representative: r_f_link2_4 and l_f_link2_4.	r_f_link1_4.STL; l_f_link1_4.STL; r_f_link2_4.STL; l_f_link2_4.STL.	The source is the fourth link of each finger. The current 4F group uses finger 2 as the representative non-thumb surface.
Sharpa	Legacy side-shared thumb/4F	Thumb representative: right_thumb_elastomer. 4F representative: right_index_elastomer.	thumb_elastomer_surface.STL; elastomer_surface.STL.	Preserves the original TH/4F naming. The legacy assets store outward-facing normals, which are flipped at runtime so that all hands share the inward-facing normal convention.
Allegro	Side-specific thumb/4F	Thumb: multi_link_15.0_tip and link_15.0_tip. 4F representative: multi_link_3.0_tip and link_3.0_tip.	link_tip.obj.	All fingertips reuse the same tip mesh. The Blender export keeps the fingertip offset convention, with side-specific sensor normals.
Orca	Side-specific thumb/4F	Thumb: multi_right_thumb_dp and left_thumb_dp. 4F representative: multi_right_index_ip and left_index_ip.	right_collision_thumb_dp_skin_mesh.stl; left_collision_thumb_dp_skin_mesh.stl; right_collision_index_ip_skin_mesh.stl; left_collision_index_ip_skin_mesh.stl.	Uses collision skin geometry rather than visual body geometry. Left and right origins/rotations differ, so side-specific meshes are required.
Revo2	Per-finger independent	Right and left thumb_touch_link, index_touch_link, middle_touch_link, ring_touch_link, and pinky_touch_link.	right_thumb_touch_link.STL; right_index_touch_link.STL; right_middle_touch_link.STL; right_ring_touch_link.STL; right_pinky_touch_link.STL; left-hand equivalents.	Each touch link is a tactile surface. The non-thumb meshes differ enough that a shared 4F map would lose edge fidelity.
RH56DFX	Per-finger independent	Right and left thumb_rubber_3, index_rubber_2, middle_rubber_2, ring_rubber_2, and little_rubber_2.	right_thumb_rubber_3.STL; right_index_rubber_2.STL; right_middle_rubber_2.STL; right_ring_rubber_2.STL; right_little_rubber_2.STL; left-hand equivalents.	Rubber pad dimensions differ across fingers, especially the little finger, so each finger is reconstructed separately.
RH5DG2	Side-specific thumb/4F	Thumb: right_thumb_force_sensor and left_thumb_force_sensor. 4F representative: right_index_force_sensor and left_index_force_sensor.	right_thumb_force_sensor.STL; left_thumb_force_sensor.STL; right_index_force_sensor.STL; left_index_force_sensor.STL.	The force-sensor meshes are used as the contact pads. The non-thumb force-sensor geometry is treated as shareable within each side.
Schunk	Side-specific thumb/4F	Thumb: right_hand_c and left_hand_c. 4F representative: right_hand_t and left_hand_t.	d13.obj; d13_left.obj; finger_tip.obj.	Site-body overrides move the tactile sites from distal bodies to fingertip bodies. Non-thumb fingers reuse the same fingertip mesh.
Shadow	Side-specific thumb/4F	Thumb: thdistal and l_thdistal. 4F representative: ffdistal and l_ffdistal.	th_distal_pst.obj; f_distal_pst.obj.	The Shadow URDF applies a 0.001 mesh scale. The non-thumb distal mesh is shared across the four non-thumb fingers.
Wuji	Side-specific thumb/4F	Thumb: right_finger1_tip_link and left_finger1_tip_link. 4F representative: right_finger2_tip_link and left_finger2_tip_link.	right_finger1_tip_link.STL; left_finger1_tip_link.STL; right_finger2_tip_link.STL; left_finger2_tip_link.STL.	Tactile sites attach to the true fingertip links. Finger 2 represents the non-thumb group in the 4F-shared version.
Ability	Per-finger independent	Right and left thumb_L2, index_L2, middle_L2, ring_L2, and pinky_L2.	thumb_F2_right.STL (right thumb); thumb_F2_left.STL (left thumb); idx_F2_Lg.STL (index, middle, ring, and left pinky); idx_F2.STL (right pinky).	The right-hand pinky source mesh is smaller than the shared non-thumb mesh, so Ability uses per-finger surface-map groups.
LEAP	Side-specific thumb/4F	Thumb: thumb_fingertip. 4F representative: fingertip.	thumb_fingertip.obj; fingertip.obj.	LEAP is a four-finger hand. The non-thumb fingertip mesh is shared, but its URDF visual origin must be preserved unless the Blender export is already in the attach-link frame.

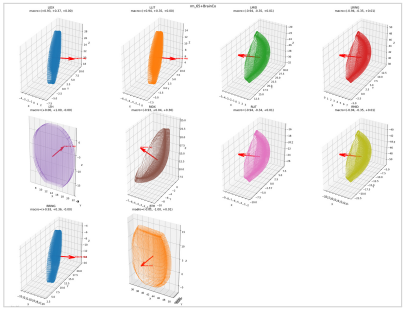
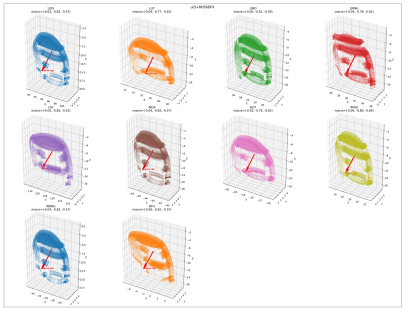
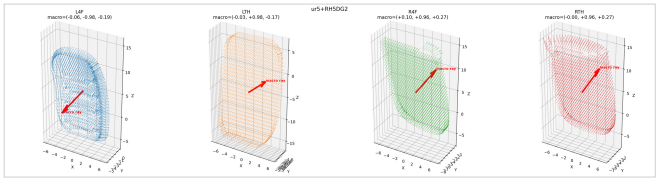
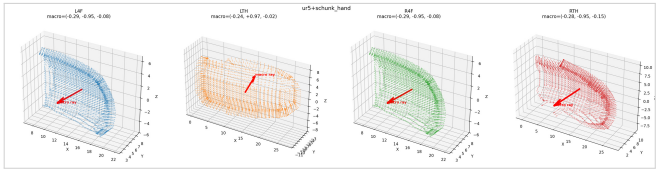
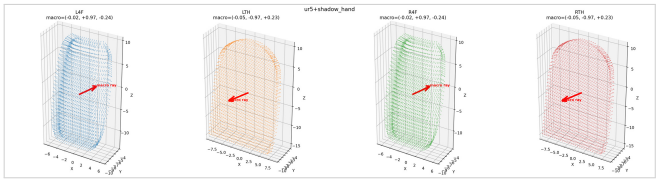
Table 8 lists the source meshes used for Blender reconstruction, and Table 9 visualizes the resulting unified tactile surfaces for all twelve hands.

Table 9: Reconstructed tactile contact surfaces for the twelve Bench2Dex hands.

Hand	Reconstructed tactile surface
DexHand021	
Sharpa	
Allegro	
Orca	

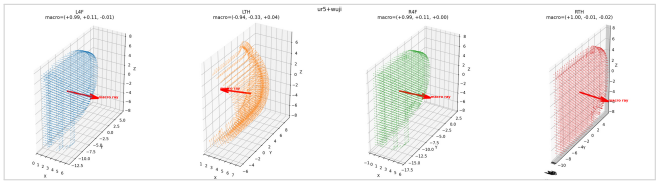
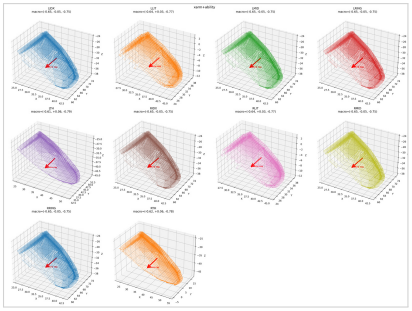
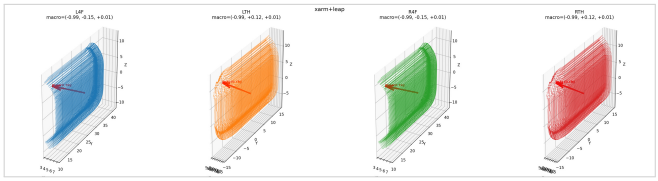
Continued on next page

Table 9: Reconstructed tactile contact surfaces for the twelve Bench2Dex hands (continued).

Hand	Reconstructed tactile surface
Revo2	
RH56DFX	
RH5DG2	
Schunk	
Shadow	

Continued on next page

Table 9: Reconstructed tactile contact surfaces for the twelve Bench2Dex hands (continued).

Hand	Reconstructed tactile surface
Wuji	
Ability	
LEAP	

D.1 Signal Semantics and Depth Quantization

The raw value before quantization is the ray-cast distance along the inward-facing local surface normal: each ray is launched from a nominal surface point toward the finger interior, so the first-hit distance measures the depth to which an object surface has penetrated past the nominal contact surface, rather than the gap to an approaching but non-contacting object. Let $d_{s,t}(u, v)$ denote this metric distance, in meters, at tactile pixel (u, v) for site s and frame t. The contact-consistency check is a two-pass test: after the first hit, the ray origin is advanced to just short of the hit point and a reverse ray is cast; the sample is kept only when the reverse ray also intersects the target mesh, which holds once the object surface has crossed the ray's nominal surface origin. Rays are cast against the meshes of the dynamic task objects and, for articulated task objects, against their mesh-bearing rigid links, within a maximum cast range of 15 mm; static scene geometry such as the tabletop is not among the raycast targets and produces no tactile response. Missed rays (no target hit within this range) and samples that fail the check are assigned zero distance and are additionally recorded by a binary contact-validity mask. For valid samples, the distance is first converted to millimeters,

$$z_{s,t}(u, v) = 10^3 d_{s,t}(u, v), \tag{5}$$

and then mapped to an 8-bit integer through the piecewise quantizer used by the released TacMap implementation [60],

$$\mathcal{Q}(z) = \begin{cases} z/0.005, & z < 0.5, \\ (z - 0.5)/0.03 + 100, & z \geq 0.5, \end{cases} \tag{6}$$

followed by clipping and casting,

$$\widetilde{\mathbf{T}}_{s,t}(u,v) = \mathrm{uint8}(\mathrm{clip}(\mathcal{Q}(z_{s,t}(u,v)), 0, 255)) \,. \tag{7}$$

The quantized map is then spatially smoothed with a Gaussian kernel evaluated in floating point. The filtered response is rounded and cast back to an 8-bit integer before being stored in the tactile stream,

$$\mathbf{T}_{s,t} = \mathrm{uint8}\Big(\mathrm{round}\Big(\mathcal{G}_\sigma * \widetilde{\mathbf{T}}_{s,t}\Big)\Big) \,, \tag{8}$$

where $\mathcal{G}_\sigma$ denotes Gaussian filtering and the current implementation uses $\sigma = 1.5$; the kernel size is $\max(1, \lfloor 9/\mathrm{step} \rfloor)$ rounded up to an odd integer, which gives 9×9 at the released subsampling step of 1. The first branch allocates finer precision to small contact depths, using 0.005 mm per integer level for $z < 0.5$ mm, while the second branch uses a coarser 0.03 mm per level for larger geometric penetration-depth values. Values of 5.15 mm and above saturate at 255. Thus, each tactile-map pixel preserves high sensitivity near initial contact while remaining compact as a single 8-bit value in the HDF5 tactile stream.

E HDF5 Schema Details

Table 10: Full per-episode HDF5 organization used by Bench2Dex when all eight modalities are collected. T denotes the number of recorded frames, J the robot joint dimension, $H \times W$ the camera resolution, and $H_t \times W_t$ the surface-aligned tactile resolution (240×240 in the released data).

HDF5 node	Stored content
episode_{id}.hdf5	One synchronized trajectory episode. The parent directory also stores an `episode_manifest.json` index.
`meta/`	Task, scene, robot key, modality list, camera definitions, collection config, instruction, success flag, schema version, and dataset metadata.
`time/`	`frame_index`, `timestamp_ns`, and `sim_step` for temporal alignment across all modalities.
`episode/`	Episode flags: `is_first`, `is_last`, `done`, and `success`.
`action/`	Commanded robot action, action validity, action source, joint/action names, control mode, and action type.
`frame_valid, frame_errors`	Per-frame validity flags and error messages for filtering corrupted or partially captured frames.
cameras/{camera_id}/	Per-camera visual observations and calibration metadata.
`rgb`	RGB images, shape $[T, H, W, 3]$ when stored as dense images. *Modality: RGB images*.
`depth_m, depth_semantics`	Metric depth maps and depth convention, shape $[T, H, W]$. *Modality: depth maps*.
`intrinsic, extrinsic_world_from_cam`	Camera intrinsics and per-frame world-from-camera extrinsics.
`camera_model, fisheye calibration`	Camera model, clipping range, and optional fisheye matrix, distortion coefficients, and valid mask.
robot/	Robot proprioception and tactile observations.
`joint_names, qpos, qvel, qeffort`	Joint names and per-frame joint position, velocity, and effort, shape $[T, J]$. *Modality: robot states*.
`tactile/meta`	Surface-aligned tactile metadata, including the tactile site list and sensor settings.
`tactile/tacmap/{site_name}`	Quantized surface-aligned tactile images, shape $[T, H_t, W_t]$, uint8. *Modality: tactile signals*.
`tactile/distance_along_normal_m`	Group of per-site raw metric ray-depth datasets (float32, meters), one dataset per tactile site, each of shape $[T, H_t, W_t]$.
`tactile/contact_mask`	Group of per-site binary contact-validity datasets (bool), one dataset per tactile site, each of shape $[T, H_t, W_t]$; true where the sample passed the contact-consistency check.
objects/{obj_id}/	Per-object physical state traces.
`pose_world`	Object pose in world frame, shape $[T, 7]$ for position and quaternion. *Modality: object states*.
`lin_vel_world, ang_vel_world`	Object linear and angular velocity traces, shape $[T, 3]$.
`joint_names, qpos, qvel, qeffort`	Optional articulated-object joint state fields.
`labels/box2d/{camera_id}/{obj_id}/`	Image-space boxes `xyxy` and visibility flags, shapes $[T, 4]$ and $[T]$. *Modality: 2D boxes*.
`labels/box3d/{obj_id}/`	Oriented 3D boxes: `center_world`, `size_lwh`, and `quat_world`. *Modality: 3D boxes*.
`labels/occupancy_tsdf/`	Occupancy grid state, grid shape, bounds, voxel size, method, semantics, and frame-valid flags. *Modality: occupancy*.
`metrics/episode`	Episode-level task progress, success, and efficiency metrics when available.
`metrics/timeseries`	Per-frame or per-step metric traces aligned with the episode timeline.

Table 11: Eight data modalities recorded by Bench2Dex.

Modality	Description
RGB images	Multi-camera visual observations for policy learning and video analysis.
Depth maps	Geometric observations for 3D perception and reconstruction.
Robot states	Joint positions, velocities, efforts, and proprioceptive state information.
Object states	Object poses, velocities, and task-relevant physical states.
Tactile signals	Unified surface-aligned tactile images defined on robot hand contact surfaces.
2D boxes	Image-space object annotations for visual detection supervision.
3D boxes	Object-level 3D bounding boxes for geometric supervision.
Occupancy	TSDF or mesh-based occupancy labels for 3D scene understanding.

F Evaluation Metrics

Bench2Dex uses a unified metric suite for task completion, stage-level progress, and success-conditioned efficiency. Each evaluated task defines executable terminal conditions and stage-level subgoals. Let q, m, and p index the task, policy, and evaluation channel, respectively, and let N denote the number of recorded episodes in a (q, m, p) cell unless stated otherwise. Cell indices are omitted from aggregate metric symbols below when they are clear from context. Episode-level quantities are first summarized within each cell, and benchmark-level episode metrics are equal-weight means across tasks. For metrics defined as means over all N episodes, this task-macro mean equals the corresponding pooled episode mean because every reported cell contains 50 episodes. The task–policy SR and LSCR entries in the main table are equal-weight means of their four channel-specific values and, because all four cells have the same size, are also pooled means over the corresponding episodes. Expert-normalized efficiency is the exception: its task–policy denominator directly pools successful episodes across all four channels. Unless noted otherwise, the benchmark uses the reach-and-stop protocol.

F.1 Primary Metrics

Metric set. The core metric set is

$$\mathcal{M}_{\text{core}} = \{\text{SR},\ \text{LSCR},\ \text{Efficiency}\}. \tag{9}$$

SR is the primary completion outcome; LSCR characterizes stage-level progress, while efficiency characterizes success-conditional execution speed.

Reach-and-stop success. In the reported evaluations, the evaluator is updated once per executed physics step. Let $C_{e,j} \in \{0, 1\}$ be the terminal predicate at evaluator update j, $\delta t_{e,j}$ the elapsed simulation time since the preceding update, n_e the number of executed updates, and Δ_e the configured dwell time. The accumulated dwell time is

$$h_{e,0} = 0, \qquad h_{e,j} = C_{e,j}\big(h_{e,j-1} + \delta t_{e,j}\big). \tag{10}$$

Define $j_e^{\text{stable}} = \min\{j \le n_e : h_{e,j} \ge \Delta_e\}$ when this set is nonempty and $S_e = \mathbf{1}[j_e^{\text{stable}} \text{ exists}]$. The default dwell time is 0.5 s. Under reach-and-stop, an episode terminates at stable success; otherwise it ends at the evaluation horizon or when evaluation cannot proceed. A recorded episode that terminates because of a policy or runtime evaluation error remains in the denominator and is counted as unsuccessful. The stable success rate is

$$\text{SR} = \frac{1}{N}\sum_{e=1}^{N} S_e. \tag{11}$$

This protocol avoids counting transient contacts or unstable placements as successful episodes.

Latched stage progress. Binary success is insufficient for long-horizon manipulation because a policy may complete early stages but fail later. Each task specifies a set of stages $\mathcal{G}_e = \{g_{e,k}\}_{k=1}^{K_e}$ with optional dependency edges. Let $L_{e,k}$ be one if the predicate of stage k is true at some update $j \le n_e$ and all dependencies were latched earlier or become valid in the same-step fixed-point

closure over the task's acyclic dependency graph; otherwise let it be zero. Once set, $L_{e,k}$ remains one. The primary progress metric is

$$\mathrm{LSCR}_e = \frac{1}{K_e} \sum_{k=1}^{K_e} L_{e,k}. \tag{12}$$

The benchmark defines aggregate LSCR as the mean of LSCR_e over all evaluated episodes. It measures ever-reached, dependency-valid milestones rather than final-state satisfaction.

Expert-normalized efficiency. For the task–policy efficiency reported in Table 2, let $\mathcal{E}^{\mathrm{succ}}_{q,m}$ contain all successful episodes pooled across the None, Equi., Inv., and Full channels, and let H^{stable}_e be the executed steps to stable success under the counting convention reported for that policy. We first compute

$$\overline{H}^{\mathrm{success}}_{q,m} = \frac{1}{|\mathcal{E}^{\mathrm{succ}}_{q,m}|} \sum_{e \in \mathcal{E}^{\mathrm{succ}}_{q,m}} H^{\mathrm{stable}}_e, \tag{13}$$

$$\mathrm{Efficiency}_{q,m} = \frac{H^{\mathrm{expert}}_q}{\overline{H}^{\mathrm{success}}_{q,m}}, \tag{14}$$

where H^{expert}_q is the task-level expert reference step count in the same step unit as the policy count. Direct pooling makes $\overline{H}^{\mathrm{success}}_{q,m}$ equivalent to weighting each channel's mean steps to success by that channel's number of successful episodes; it is not an equal-weight mean of nonempty channel means. A value of one denotes equal expert-reference and policy step counts under the stated convention, values above one indicate fewer steps than the expert reference, and higher is better. Efficiency is undefined when no episode succeeds and is therefore interpreted together with SR and the number of successful episodes. The table's task-macro efficiency averages only tasks for which efficiency is defined and reports that denominator explicitly. Expert-normalized efficiency is not recomputed by averaging channel-level efficiency ratios and is comparable only under matched counting conventions, expert-reference construction, simulation timing, initial-state distribution, terminal condition, dwell time, and evaluation horizon.

F.2 Reproducibility Metadata

Each evaluation summary is accompanied by the protocol required to interpret and reproduce its scores: policy identity, base seed and seed-derivation rule, number of episodes, termination rule, maximum evaluation horizon, and dwell time. For randomized evaluation, the resolved episode seed and sampled generalization parameters are retained for each rollout. Together, these quantities support reproducible audit and reconstruction under a matched simulator, backend, task, and perturbation configuration.

G Policy Training Configurations

This section discloses the training configurations of the four policies evaluated in Table 2: ACT, DP, $\pi_{0.5}$, and GR00T N1.5. ACT and DP are trained from scratch on Bench2Dex demonstrations; $\pi_{0.5}$ and GR00T N1.5 are fine-tuned from their respective public checkpoints. For $\pi_{0.5}$ and GR00T N1.5, whose pretrained action heads assume lower-dimensional action spaces than the bimanual-dexterous embodiments require, we retain the pretrained weights and randomly initialize the additional output dimensions to match the target action space. Unless stated otherwise, each policy is trained on a single task–embodiment setting and consumes the synchronized multi-view RGB and proprioceptive observations recorded in the unified HDF5 episodes. The default hyperparameters reported below are those of the released training launchers (`policy/<name>/train.sh`); per-task overrides, when applied, are logged with the corresponding checkpoint.

ACT. We train ACT from scratch using its conditional variational autoencoder (cVAE) formulation. The encoder and decoder are Transformer networks with a hidden dimension of 512 and a feed-forward dimension of 3,200. Inputs are four camera views (both wrist and both stereo cameras) encoded by a shared ResNet-18 backbone together with the normalized proprioceptive state, whose dimension is sized to each embodiment's active degrees of freedom. The policy predicts an action

chunk of length $T_{\text{pred}} = 30$ from a single observation ($T_o = 1$) and is supervised with an ℓ_1 reconstruction loss regularized by a KL term of weight 10. We optimize with AdamW at a learning rate of 1×10^{-5} and a batch size of 32 for up to 6,000 epochs on a single RTX 4090 GPU. At inference the policy re-plans at every step and fuses overlapping chunk predictions by temporal ensembling.

Diffusion Policy (DP). We train DP from scratch using the image-conditioned 1D U-Net variant of Diffusion Policy. Multi-view RGB observations are resized to 216×288 and encoded independently by ResNet-18 visual backbones, whose features are fused into the global conditioning vector of the denoiser. The denoiser is a 1D conditional U-Net with feature widths $(256, 512, 1024)$, a 128-dimensional diffusion-step embedding, and kernel size 5. At each control cycle the policy conditions on the most recent $T_o = 3$ observations, denoises a $T_{\text{pred}} = 8$-step action trajectory, and executes the first $T_{\text{exec}} = 6$ actions before re-planning from fresh observations, yielding receding-horizon closed-loop control. Training adopts the DDPM (ϵ-prediction) objective over 100 diffusion timesteps under a squared-cosine noise schedule, and 100 denoising steps are taken at inference. We optimize with AdamW at a peak learning rate of 1×10^{-4}, a global batch size of 512, 500-step linear warmup followed by cosine decay, and an EMA of the policy weights with decay 0.9999. Each task-specific policy is trained for up to 300 epochs on eight H100 GPUs.

$\pi_{0.5}$. We fully fine-tune the pretrained $\pi_{0.5}$ vision–language–action model using the openpi implementation, initializing from the public `pi05_base` checkpoint. The released `pi05_base_dex2bench_full` configuration trains the non-LoRA PaliGemma-2B vision–language backbone and Gemma-300M action expert jointly. The state and action dimensions are selected from the active degrees of freedom of each embodiment. Compatible pretrained parameters are retained, while state/action projection layers whose shapes differ from the pretrained checkpoint remain randomly initialized. The policy conditions on four RGB views (both stereo and both wrist cameras), normalized proprioception, and the task instruction. It predicts an action chunk of $T_{\text{pred}} = 20$ steps with a maximum token length of 280; all $T_{\text{exec}} = 20$ actions are executed before the policy re-observes and re-plans. We optimize all trainable parameters with AdamW using a cosine learning-rate schedule with 100 warmup steps, a peak learning rate of 1×10^{-4}, and decay to 1×10^{-6} over 2,000 steps. The released launcher uses a global batch size of 256, 32 data-loading workers, `fsdp_devices`$= 1$, and a total of 2,000 training steps. EMA is disabled.

GR00T N1.5. We fine-tune the pretrained GR00T N1.5 generalist policy on the demonstrations of each evaluation task, without LoRA adapters. The Eagle vision tower, language model, and its MLP connector are kept frozen; the trainable parameters are the action head together with the backbone's output projector, which maps the frozen Eagle features into the action-head dimension.The policy receives four temporally synchronized RGB camera views, the embodiment proprioceptive state, and the task instruction as a per-task language prompt. To accommodate the heterogeneous bimanual-dexterous embodiments, we pad the state vector to a common 64-dimensional representation and extend the pretrained action head from its native maximum dimensionality to 64 dimensions; action channels beyond the pretrained head are newly initialized and optimized jointly with the pretrained parameters, while padded action dimensions are masked from the action objective, preserving a single policy interface across embodiments. GR00T N1.5 uses a flow-matching action head that, from a single observation ($T_o = 1$), predicts a $T_{\text{pred}} = 16$-step action chunk; at inference the chunk is generated with 4 flow-matching denoising steps and executed in full ($T_{\text{exec}} = 16$) before the policy re-observes and re-plans. We optimize with AdamW at a peak learning rate of 1×10^{-4} with 5% linear warmup followed by cosine decay and a global batch size of 64, training in bfloat16 mixed precision for 20,000 steps on a single H100 GPU.

H TACTILE INTEGRATION INTO GR00T N1.5

This section describes the tactile-conditioned GR00T N1.5 variant evaluated in Section 3.5. It retains the RGB, proprioceptive, and language inputs and adds a tactile branch to the action head. All 26 evaluated task models use the same fusion architecture described below, with the number of tactile sites determined by the embodiment.

Tactile input and normalization. At each policy observation, we read the surface-aligned metric depth maps recorded in the HDF5 field `distance_along_normal_m`, synchronized with the

RGB and proprioceptive inputs. These are the raw floating-point contact-depth measurements described in Appendix D.1; the policy does not consume the quantized 8-bit tactile maps. For site s, we normalize the depth map as

$$X_s = \frac{\mathrm{clip}(d_s, 0, d_{\max})}{d_{\max}}, \qquad d_{\max} = 0.015\,\mathrm{m}. \tag{15}$$

Non-contact and invalid ray samples have zero depth. Each site supplies one 240×240 channel, yielding an input tensor of shape $B \times N \times 1 \times 240 \times 240$, where B is the batch size and $N \in \{8, 10\}$ is the total number of tactile sites across both hands. Site ordering is fixed by the model's saved tactile schema. The signal represents geometric contact depth, with no additional force or shear channels.

Shared spatial token encoder. We apply the same CNN independently to every tactile site. Its three convolutional layers have output widths $(32, 64, 128)$, kernel sizes $(5, 3, 3)$, strides $(2, 2, 2)$, and padding $(2, 1, 1)$, each followed by ReLU. Adaptive average pooling produces a 2×2 feature grid per site. A linear projection maps each 128-dimensional grid feature to a 1536-dimensional token, and learned site and spatial-position embeddings identify its source:

$$z_{s,p} = W f_{s,p} + b + e^{\mathrm{site}}_s + e^{\mathrm{spatial}}_p, \qquad p \in \{1, 2, 3, 4\}. \tag{16}$$

Here $f_{s,p}$ is the pooled CNN feature at grid position p. Concatenating the tokens over sites and grid positions gives $Z \in \mathbb{R}^{4N \times 1536}$ per observation: 32 tokens for eight sites or 40 tokens for ten sites. This preserves coarse spatial information within each contact map while sharing the local feature extractor across sites.

Residual cross-attention after the diffusion transformer. The tactile tokens condition the action head directly. Let $H \in \mathbb{R}^{16 \times 1024}$ denote the action-position features output by GR00T's diffusion transformer (DiT), which has already incorporated the vision-language context, proprioception, and noisy action chunk. A single eight-head cross-attention layer uses these action features as queries and the tactile tokens as keys and values:

$$\widetilde{H} = H + g\ \mathrm{MHA}(\mathrm{LN}_q(H), \mathrm{LN}_k(Z), Z)\,. \tag{17}$$

The attention layer projects the 1536-dimensional keys and values to its 1024-dimensional attention space. Separate layer normalizations are applied to queries and keys; values use the unnormalized tactile tokens. The embodiment-conditioned action decoder then maps $\widetilde{H}$ to the predicted flow velocity. This `post_dit` fusion leaves tactile tokens outside the vision-language backbone and allows each action position to attend to the current contact pattern. We initialize the attention output projection's weights and bias to zero, so the added residual initially contributes zero.

Tactile dropout and training objective. During training, we independently drop the entire tactile modality for each sample with probability 0.3. For a dropped sample, all tactile maps are zeroed and the residual gate is set to $g = 0$; otherwise $g = 1$. Gating the residual also suppresses the learned site and spatial embeddings when tactile input is dropped. At evaluation, dropout is disabled and $g = 1$. The model is trained with the standard masked flow-matching objective. For a demonstration action chunk A, Gaussian noise ϵ, and sampled flow time t, the noisy input is $A_t = (1 - t)\epsilon + tA$ and the target velocity is $A - \epsilon$:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\frac{\left\| M \odot \left(\widehat{V}_\theta - (A - \epsilon)\right)\right\|_F^2}{\sum_{i,j} M_{ij}}\right]. \tag{18}$$

Here $\widehat{V}_\theta$ is the predicted velocity and M is the binary action-validity mask. State and action vectors use the common padded 64-dimensional interface; padded action dimensions are excluded from the loss. The tactile branch receives supervision through this action objective, without an auxiliary tactile reconstruction or contact-prediction loss.

Optimization and demonstration preprocessing. We initialize from pretrained GR00T N1.5 and train a separate model for each task. In this tactile variant, the Eagle vision and language backbone is frozen, while the action-head projectors, DiT, action decoder, and added tactile encoder and cross-attention modules are trainable. No LoRA adapters are used. The recorded training runs use AdamW

with a peak learning rate of 10^{-4}, weight decay 10^{-5}, 5% linear warmup followed by cosine decay, batch size 64, bfloat16 mixed precision, and 20,000 optimization steps, with random seed 42. Demonstrations are truncated before the annotated return-to-home frame when a valid homing annotation is available, so the terminal homing motion is excluded from the action targets. The backbone freezing setting is the same as for the non-tactile GR00T baseline in Appendix G; the two differ only in the additional tactile modality.

Inference. The policy uses one current multimodal observation to generate a 16-step action chunk with four flow integration steps. Tactile maps are encoded once per policy query, and the resulting tokens are reused in the residual cross-attention at every integration step. Deployment uses the saved site ordering, map resolution, and depth normalization to match training.

I Generalization Configuration

Scene background. Scene background randomization changes the visual surroundings while leaving the tabletop task unchanged. The benchmark includes 60 iTHOR/USD indoor scenes, partitioned into 50 seen and 10 unseen scenes. Each scene has a calibrated yaw and translation offset, and, when this factor is resampled, each episode samples exactly one scene from its designated partition; the nominal background is not substituted. The geometry of the sampled background scene is used only for visual rendering and is excluded from collision and contact simulation. This factor therefore evaluates object grounding under unseen scene backgrounds without introducing room-level physical interactions.

Tabletop texture. Tabletop texture randomization changes the visual material of the support surface while preserving its geometry and contact behavior. The texture collection contains 11,824 materials across carpet, fabric, flooring, leather, metal, rust, stone, and wood, with 9,436 seen and 2,388 unseen materials assigned by a deterministic 80/20 split. When this factor is resampled, each episode receives a sampled texture, preventing a fixed tabletop texture from becoming a localization shortcut.

Lighting conditions. Lighting randomization modifies the lights embedded in the sampled USD room. The distant light samples intensity in $[1000, 1500]$, each color channel in $[0.4, 1.0]$, pitch in $[-30^\circ, -5^\circ]$, yaw in $[-90^\circ, 90^\circ]$, and angular size in $[0.5^\circ, 1.0^\circ]$. The dome light samples intensity in $[150, 350]$, color between $[0.8, 0.8, 0.6]$ and $[1, 1, 1]$, and exposure in $[-0.5, 0.5]$. These ranges are shared across splits and probe sensitivity to illumination, shadows, contrast, and exposure.

Object pose. Object pose randomization changes task-relevant initial poses within valid bounds. For explicitly positioned objects, x and y are independently perturbed by up to ± 2 cm and yaw by up to $\pm 10^\circ$. If a sampled pose violates collision or workspace constraints, the perturbation magnitude is progressively reduced; the nominal pose is retained only when no valid perturbed pose is found. Objects without a fixed pose are instead resampled within task-defined zones subject to collision constraints. Task-specific geometric constraints may fix an object's initial pose or restrict its perturbation range.

Camera pose. Camera pose randomization perturbs external and wrist-mounted camera viewpoints. World-mounted cameras sample translation and look-at-target offsets of ± 5 mm per axis, distance offsets of ± 3 cm, and roll–pitch–yaw offsets of $\pm 1^\circ$ per axis. Wrist cameras sample link-frame translation offsets of ± 5 mm per axis and rotation offsets of $\pm 1^\circ$ per axis. Stereo cameras share one perturbation sample to preserve their relative pose. The seen and unseen camera profiles use the same numerical ranges, so this factor evaluates calibration and viewpoint perturbations rather than a disjoint camera-hardware split.

Distractor objects. Distractor object randomization adds one to three dynamic distractors per episode. The distractor set contains 68 assets, partitioned into 52 seen and 16 unseen assets; each selected object is scaled by a factor in $[0.8, 1.2]$. Objects that duplicate a task-relevant object or are semantically confusable with one are excluded before sampling. Placement satisfies collision constraints, a 5 cm table-edge margin, and a central exclusion region $x \in [-0.25, 0.25]$ m and $y \in [-0.20, 0.35]$ m. This procedure evaluates grounding under occlusion, ambiguity, crowding, and incidental contact without deliberately blocking the primary workspace.

Table 12: Composition and perturbation strengths of the four Bench2Dex evaluation channels. Entries prefixed by *From anchor* replay the resolved parameters of the matched anchor episode. Task-level object overrides may narrow or disable pose perturbations.

Channel	Scene construction	Invariance factors: scene background, tabletop texture, lighting conditions, distractor objects, camera pose	Equivariance factors: object pose, table height	Evaluation purpose
`none`	Matched anchor; all factors are replayed exactly.	*From anchor*: scene background, tabletop texture, lighting conditions, distractor objects, and camera pose.	*From anchor*: object poses and table height.	Establishes the matched baseline without resampling.
`equi_only`	Matched anchor; only equivariance factors are resampled.	*From anchor*: scene background, tabletop texture, lighting conditions, distractor objects, and camera pose.	*Resampled*: explicit object x–y: ± 2 cm; yaw: $\pm 10^\circ$; objects without fixed pose: task zones; table: ± 5 cm.	Tests adaptation to task-relevant geometry while controlling visual context.
`inv_only`	Matched anchor; only invariance factors are resampled from the unseen split.	*Resampled from unseen split*: scene background: 10 iTHOR scenes; tabletop texture: 2,388 materials; distant/dome lighting: ranges above; distractor objects: 1–3 objects from 16 assets at 0.8–$1.2\times$ scale; world-mounted camera translation/look-at-target offsets: ± 5 mm, distance: ± 3 cm, RPY: $\pm 1^\circ$; wrist-mounted camera translation: ± 5 mm, RPY: $\pm 1^\circ$.	*From anchor*: object poses and table height.	Tests nuisance robustness under matched task geometry.
`inv_equi`	Independent full-scene sample; no anchor is used.	*Resampled from unseen split*: the same perturbations to scene background, tabletop texture, lighting conditions, distractor objects, and camera pose as `inv_only`.	*Resampled*: the same perturbations to object pose and table height as `equi_only`.	Tests simultaneous perceptual invariance and geometric adaptation.

Table height. Table height randomization samples a vertical offset in $[-0.05, 0.05]$ m around each task's nominal tabletop height (default 0.75 m). The robot mount remains at its nominal height, while on-table object placement, evaluator height references, and dependent collision checks follow the shifted surface. The resulting change in robot-to-table geometry tests adaptation of reaching, grasping, and contact heights.

J COLLECTED TACTILE OBSERVATIONS

The following figures present representative tactile observations collected across 12 robot–hand embodiments. Each image is shown individually and without cropping. Each figure visualizes the tactile maps of one replayed frame, with one heatmap per tactile site; brighter pixels encode larger quantized penetration depth (values 0–255, Appendix D.1).

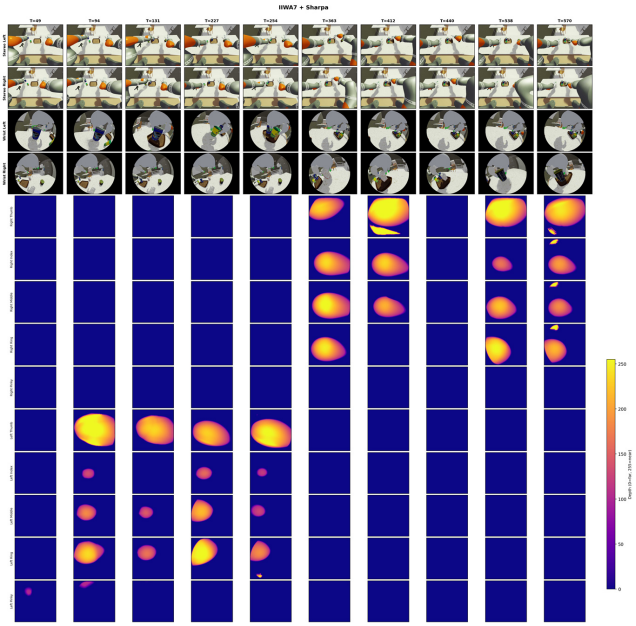


Figure 3: Visualization of tactile data collected with the IIWA7+Sharpa embodiment.

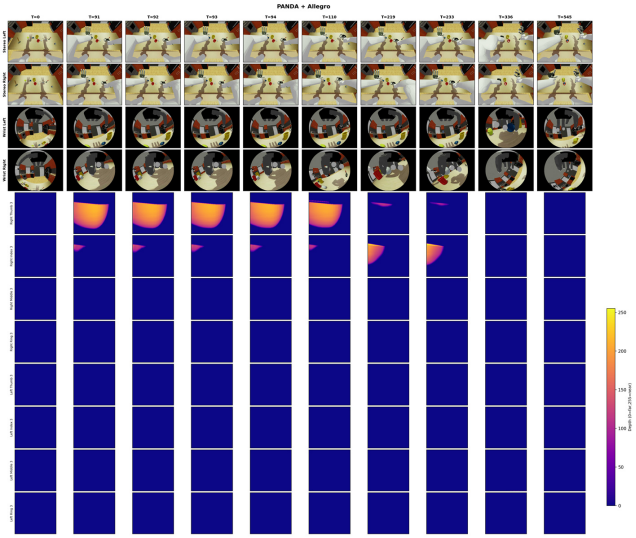


Figure 4: Visualization of tactile data collected with the Panda+Allegro embodiment.

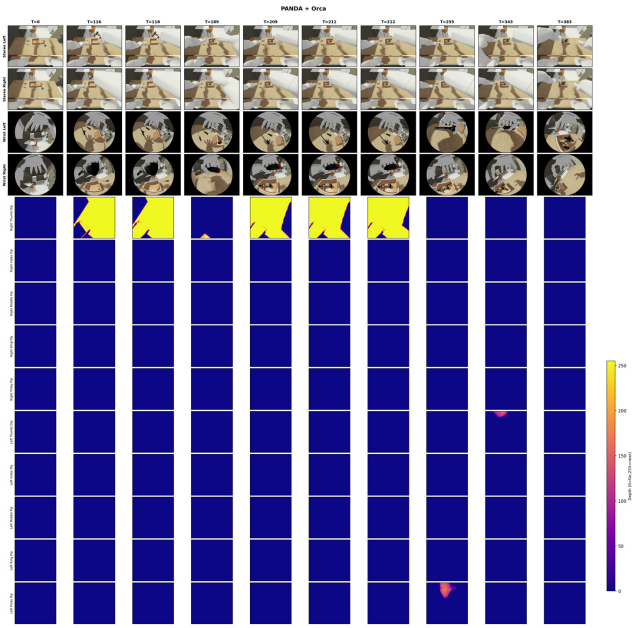


Figure 5: Visualization of tactile data collected with the Panda+Orca embodiment.

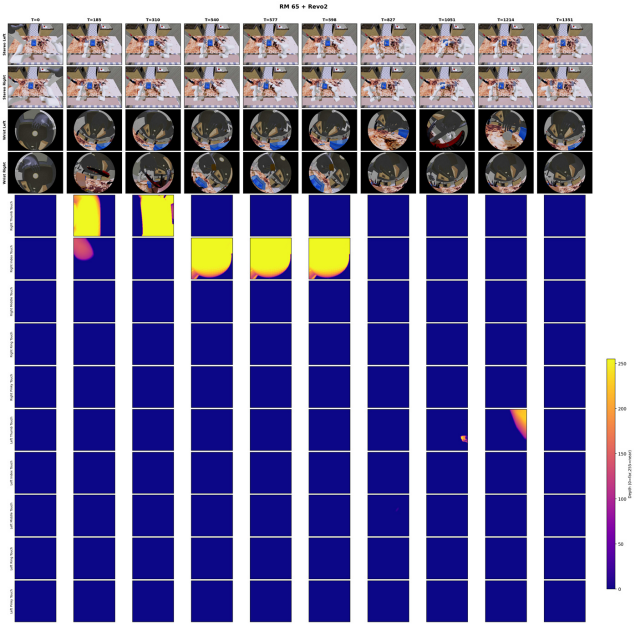


Figure 6: Visualization of tactile data collected with the RM65+Revo2 embodiment.

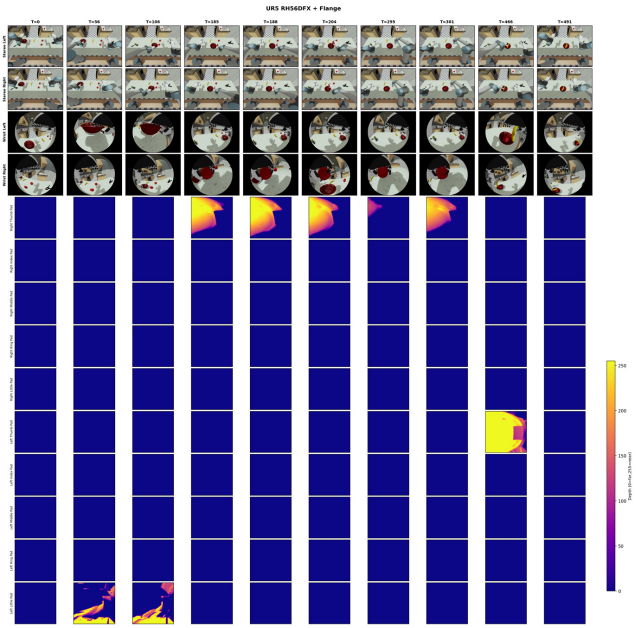


Figure 7: Visualization of tactile data collected with the UR5+RH56DFX embodiment.

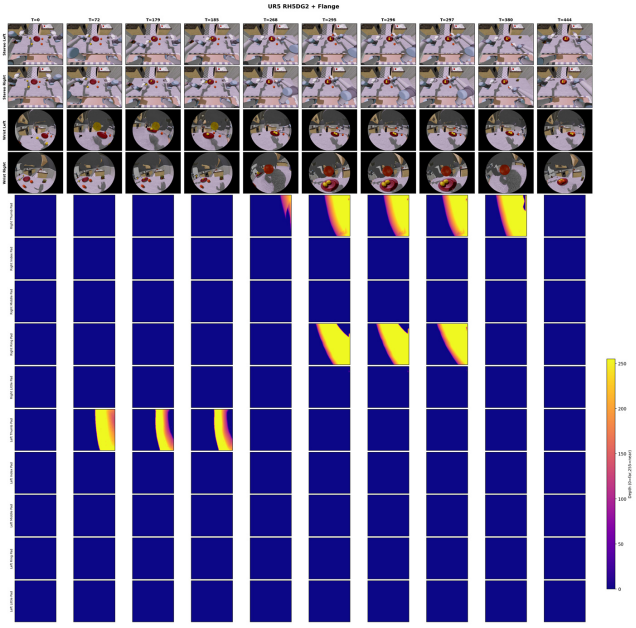


Figure 8: Visualization of tactile data collected with the UR5+RH5DG2 embodiment.

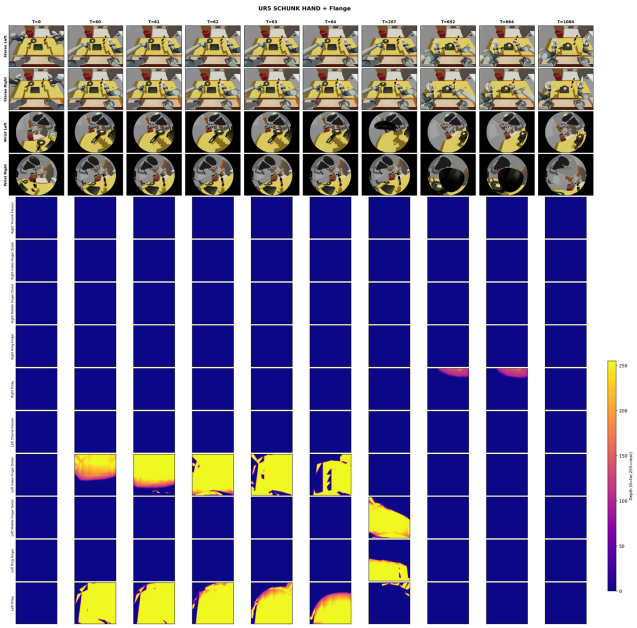


Figure 9: Visualization of tactile data collected with the UR5+Schunk embodiment.

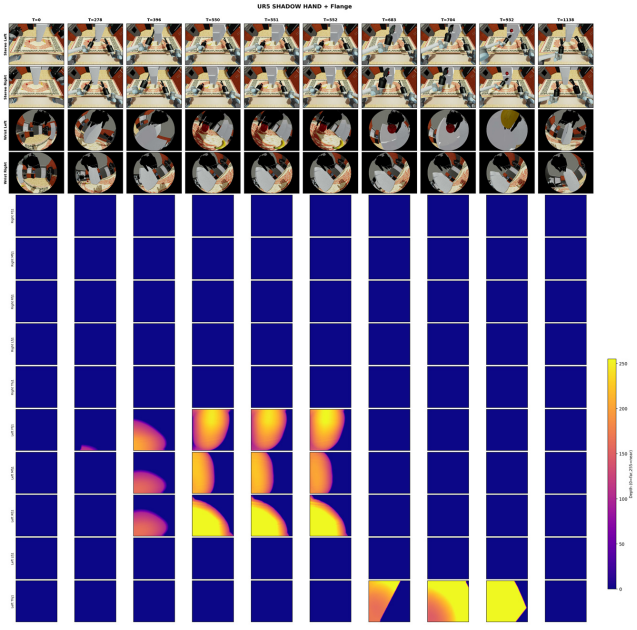


Figure 10: Visualization of tactile data collected with the UR5+Shadow embodiment.

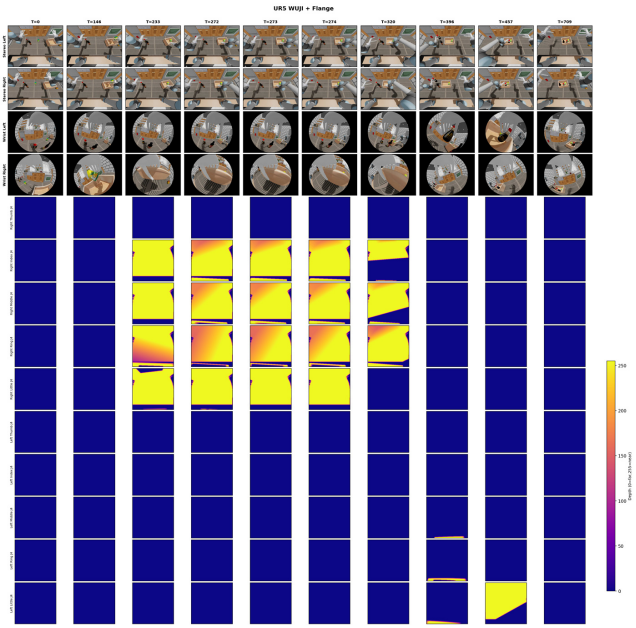


Figure 11: Visualization of tactile data collected with the UR5+Wuji embodiment.

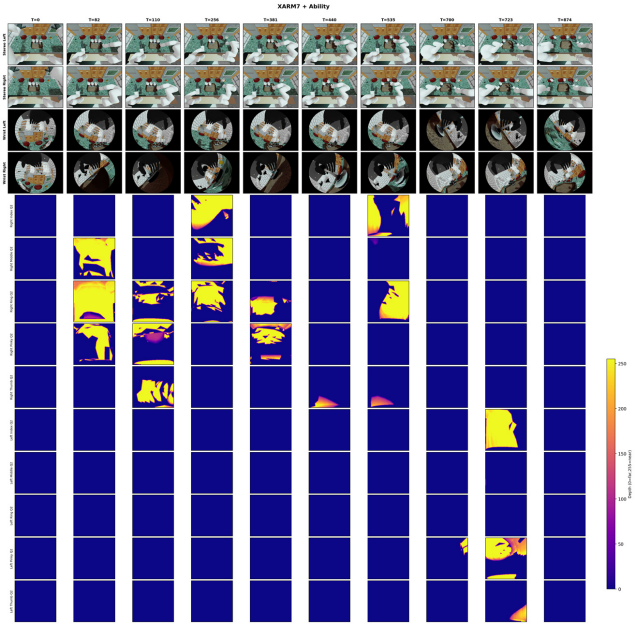


Figure 12: Visualization of tactile data collected with the xArm7+Ability embodiment.

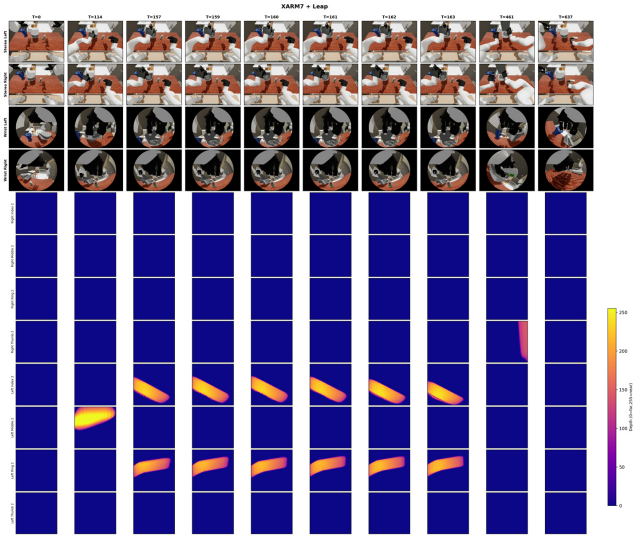


Figure 13: Visualization of tactile data collected with the xArm7+LEAP embodiment.

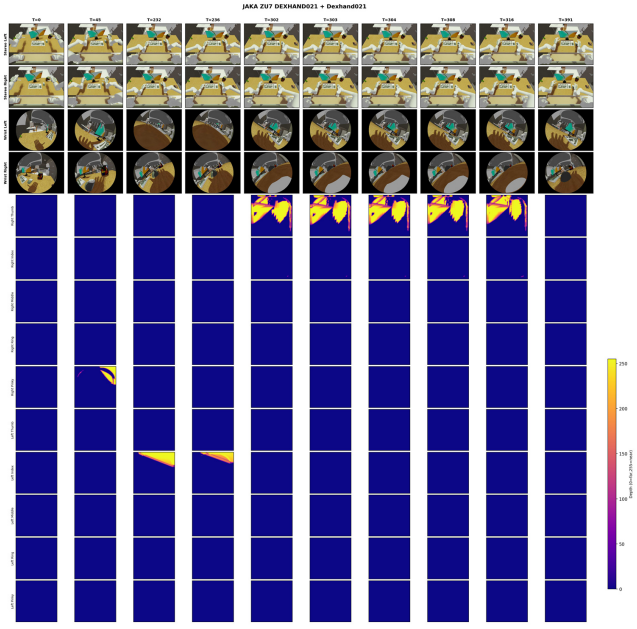


Figure 14: Visualization of tactile data collected with the JAKA ZU7+DexHand021 embodiment.