DEXOMNI: A LARGE-SCALE SIMULATION BENCHMARK FOR GENERALIZABLE DEXTEROUS MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dexterous manipulation requires reliable contact establishment, temporal coordination of multi-finger and bimanual actions, as well as maintenance of achieved goals, while also transferring these abilities to unfamiliar objects, interactive relationships, and skill combinations. Herein, we introduce DexOmni, a simulation data platform and benchmark featuring 66 tasks whose feasibility was verified through human demonstrations, 18,000 curated human demonstrations, high-fidelity re-rendering, and visual and goal-preserving prompt randomization. Covering a diverse range of interaction types, task structures, and asset types, core-50 provides demonstrations and execution tests for 50 tasks spanning functional tool use, numerical and spatial grounding, articulation, and bimanual manipulation. Transfer-16 reserves 16 tasks to evaluate asset and relation transfer, skill composition, and deformable manipulation without target-task demonstrations in the benchmark training set. We compare task-specialized diffusion policies, vision-language-action models, and world-action models using success and native-stage diagnostics. Across 40,000 Core-50 episodes, the best of eight multi-task checkpoints reaches 29.1% task-macro success. Success rates are 35–51% for pushing, articulation, and pick-and-place but only 2–3% for tool use and precision fitting; this gap persists after controlling for demonstration count and duration. Failures in contact establishment, temporal coordination, and goal maintenance emerge as the main bottlenecks in multi-finger and bimanual control. Transfer-16 leaves the most headroom; ordered progress separates its checkpoints, and compositions add another failure: policies repeat finished steps or start later ones early. Overall, DexOmni provides a challenging benchmark for systematically evaluating the reliability and generalization of embodied policies for dexterous robotic hands.

1 INTRODUCTION

Dexterous manipulation requires controlling how contacts change during an operation. Moving a mouse must not trigger a premature click; setting two knobs requires both knobs to remain at their target settings simultaneously; and placing a lid requires preserving alignment while releasing it. Evaluating these capabilities requires demonstrations of varied interactions and diagnostics that distinguish initial progress from reliable functional completion. Recent dexterous benchmarks broaden task and embodiment coverage (Wang et al., 2026a; Yao et al., 2026; Yang et al., 2026; Tang et al., 2026), but their human-demonstration sets are small and their tasks rarely include ones reserved for evaluation.

DexOmni provides 18,000 curated simulation demonstrations, approximately 56 hours of interaction with three synchronized camera views. This is 5.7–16.4 times the human trajectory count of representative dexterous benchmarks under the convention in Table 1. High-fidelity re-rendering produces varied observations of recorded behavior while preserving state/action supervision. Visual domain randomization and meaning-preserving prompt randomization broaden appearance and language coverage around task-specific configurations. Figure 1 connects this data pipeline to evaluation and illustrates correspondence with the physical system.

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

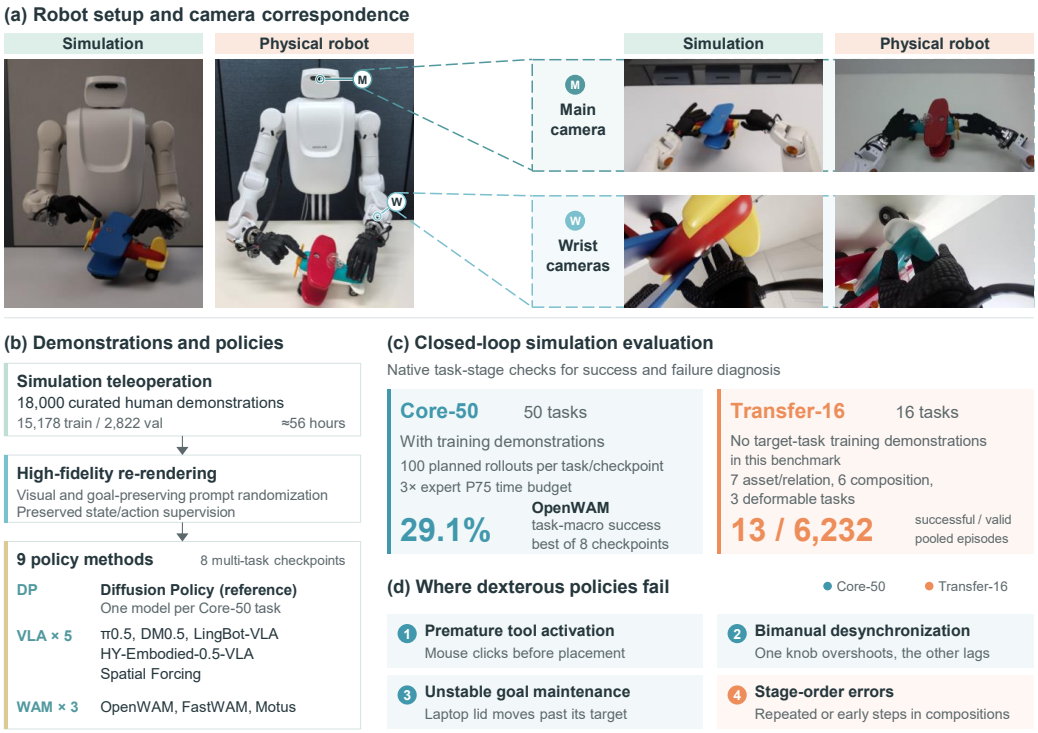


Figure 1: DexOmni overview. (a) Simulated and real views of the same toy-aircraft task from the setup camera (not a policy input), wrist cameras (W), and main camera (M); real images show appearance only, and real-robot results are shown in videos only. (b) Data and policies. (c) Evaluation suites. (d) Failure modes analyzed in Section 5.5.

The benchmark comprises two complementary suites: **Core-50**, 50 tasks with demonstrations for policy learning and fresh-rollout evaluation, and **Transfer-16**, 16 tasks reserved for generalization evaluation. Core-50 tasks combine functional and numerical understanding, articulation, spatial relations, precise contacts, and bimanual execution (Figure 2). Transfer-16 tests new asset relations and skill combinations, including three deformable interactions without deformable-task demonstrations in this dataset. The simulated embodiment follows a physical platform, and we include real-robot rollout videos.

Our Core-50 study compares eight multi-task checkpoints and a single-task diffusion-policy reference. The best multi-task score is 29.1%. Tool use and precision fits are nearly unsolved, and demonstration count and duration do not explain this ordering; two VLA checkpoints barely condition on which object the instruction names. Failures concentrate where closing the grasp fires a button, where both hands must hold their targets together, and where reached goals must be kept. These findings motivate evaluating what happens after acquisition, alongside aggregate success.

Our contributions are:

1. A large human-demonstration dataset, re-rendered with visual and prompt randomization.
2. Core-50 and Transfer-16: 50 demonstration-backed tasks and 16 evaluation-only asset, skill-composition, and deformable tasks, scored by native stage checkers.
3. A comparison of DP, five VLA, and three WAM policies with a confound-controlled skill analysis and a language-conditioning probe, locating failures at three hand transitions and, in compositions, at step order and completion.

Table 1: Dexterous learning resources (DMG: DexMimicGen; NR: not reported). Human and generated trajectories are counted separately; catalogs and evaluation scopes differ; DexOmni shaded.

	DexJoCo	DexVerse	Bench2Dex	DMG	SPD	**DexOmni**
Task catalog	11	100	26	9	6 scenes	**50 + 16**
Human demos	1,100	3,180	∼1,300	60	∼2,000	**18,000**
Generated demos	—	—	—	21,000	—	—
Language variation	Digits / words	NR	NR	NR	No input	**32 / 8 per goal**[a]
Test-only tasks	NR	NR	NR	NR	NR	**16 tasks**
Transfer setting	Visual; language	Visual; embodiment	Visual / geometric	Initial states; objects	Real-task finetuning	**Assets; skills; soft objects**
Policy coverage	ACT, DP, VLA	DP, 3D DP, VLA	ACT, DP, VLA	BC-RNN, DP	Diffusion transformer	**DP; 5 VLA; 3 WAM**
Extra diagnostics	Failures; grounding	Category SR	Stages; time; safety; grasp	Generation yield	Stage progress	**Transitions; retention; contact**
Real evaluation	NR	NR	NR	Real demos via sim	Real finetuning	Videos only[b]

[a] 32 train / 8 held-out paraphrases for each of 91 task–goal pairs; separate from Transfer-16. [b] Recorded real-robot rollouts. Extra views do not add human demonstrations. Sources, scope, and broader comparisons: Appendix B.13.

2 Related Work

DexJoCo, DexVerse, Bench2Dex, and LabDex cover functional and bimanual manipulation with diverse embodiments (Wang et al., 2026a; Yao et al., 2026; Yang et al., 2026; Tang et al., 2026). DexMimicGen expands human seeds through automated trajectory generation, whereas SPD studies simulation pretraining and real-data adaptation (Jiang et al., 2025; Kamat et al., 2026). Table 1 compares data scale, randomization, policy coverage, and evaluation protocols. DexOmni combines these dimensions with three WAM checkpoints and a separate dexterous task-transfer suite.

LIBERO, CALVIN, DexArt, RoboDojo, and RoboTwin motivate transfer, capability evaluation, and randomization (Liu et al., 2023; Mees et al., 2022; Bao et al., 2023; Chen et al., 2026; 2025). Earlier dexterous suites learn mainly by reinforcement or from small demonstration sets (Rajeswaran et al., 2017; Chen et al., 2022; Liu et al., 2025); large simulated and real manipulation corpora are dominated by grippers (Gu et al., 2023; Nasiriany et al., 2024; Li et al., 2022; Open X-Embodiment Collaboration, 2024; Khazatsky et al., 2024). Dexterous teleoperation and motion capture (Qin et al., 2023; Iyer et al., 2024; Wang et al., 2024) collect human data outside a benchmark; DexOmni records it in the evaluation simulator. Dexterous sim-to-real learning precedes DexOmni (OpenAI et al., 2018; Handa et al., 2023); Appendix B.13 details counting conventions and real-data adaptation.

3 The DexOmni Data Platform

3.1 Task Diversity and Human Demonstrations

DexOmni couples goal understanding to physical interaction. *Functional tool use* includes hammering, drilling, and shutter activation; *number understanding* grounds timer settings and left/right knob angles; *articulation* constrains opening, sliding, and unfolding. *Spatial and comparative goals* include placing tableware and selecting the largest fitting ball. Precision fits, regrasping, handovers, and ordered sequences add contact and coordination requirements (Figure 2). Task designs adapted from DexJoCo and DexVerse are credited in Appendix A.1 (Wang et al., 2026a; Yao et al., 2026).

Human teleoperation in Isaac Sim records synchronized images, measured states, commanded actions, language, and source metadata (NVIDIA Corporation, 2026). At least ten operators collected

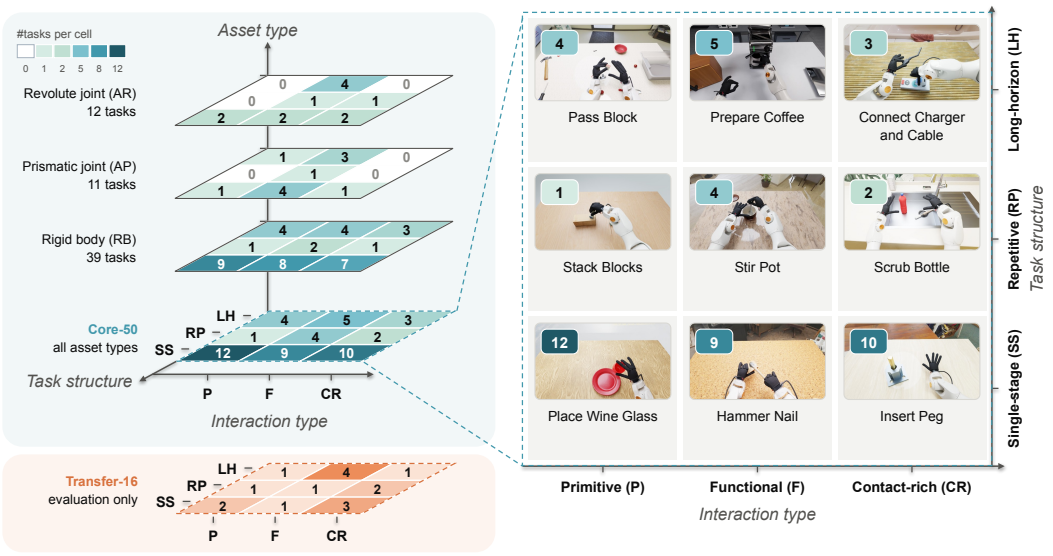


Figure 2: Core-50 tasks span primitive, functional, and contact-rich interactions; single-stage, repetitive, and long-horizon structure; and rigid/articulated assets. Scenes illustrate these combinations. Asset labels overlap. Deformable interactions are reserved for Transfer-16.

the data in 1,050 sessions. The curated package contains 18,000 trajectories (15,178 training; 2,822 validation), approximately 56 hours at 30 Hz, with one main and two wrist views; a Core-50 task has 46–463 training trajectories (median 268), and the median trajectory lasts 8.7 s (262 steps). Splits are made per trajectory or session, and re-rendered derivatives inherit them. Quotas of 500, 300, or 100 trajectories per task group, favoring shorter eligible demonstrations, define this curated distribution; collection, selection, and source-to-catalog mappings are detailed in Appendix B. The platform uses ROKAE AR5-08 arms and first-generation Wuji Hands with 20 actively controlled joints per hand (ROKAE Robotics, 2026; Wuji Technology, 2026); the simulated kinematics, asset scale, cameras, state/action frames, and timing follow the physical system (Appendix B.5).

3.2 High-Fidelity Re-rendering and Randomization

Recorded scene and robot states support physically based materials, textured surfaces, room context, and controlled lighting while preserving the source state/action sequence and temporal alignment. Visual domain randomization varies materials, lighting, backgrounds, and bounded camera extrinsics around source-specific configurations. Appearance is sampled per trajectory and remains temporally consistent, including for video-conditioned policies. Rendering settings, sampling ranges, and which labels each factor must preserve appear in Appendix B.3.

Prompt randomization preserves each episode's object, numerical goal, direction, order, and constraints. Each of the 91 active canonical-goal pairs has 32 training and eight held-out paraphrases. Changing wording is distinct from changing the executable goal. Banks and sampling revisions are frozen per run; actual conditioning remains model-specific, so structured-goal DP does not automatically use language augmentation. Appendix B.4 gives the sampler and held-out bank.

4 Benchmark Design and Evaluation

4.1 Core-50 and Transfer-16

Held-out trajectories measure action prediction; fresh closed-loop trials on Core-50 measure execution of demonstration-backed tasks. Transfer-16 (six Comb, seven Gen, three Flex) provide no demonstrations for benchmark training or task-specific finetuning; their source skills, assets, and held-out changes are identified in the task map. *Asset/relation transfer* changes objects or their relationships; *skill composition* connects trained operations, such as opening a laptop and clicking a

mouse; and *deformable manipulation* tests sponge compression, towel folding, and pillow packing without deformable-task demonstrations. Appendix B.9 gives the design and mappings. Every Core-50 and Transfer-16 task is verified before release, including in-person checks by the data-collection operators (Appendix B.2). Goal-preserving held-out-paraphrase and visual-shift protocols are specified in Appendix B.10, and Section 5.2 probes whether policies read the named object. Physical deployment follows a matched direct-transfer protocol (Appendix B.11); recorded real-robot rollouts are supplementary videos.

4.2 Execution, Stopping, and Diagnostics

Actions execute at 30 Hz with a physical-time limit of three times the expert P75 demonstration duration. Non-Flex Transfer-16 tasks use three times the sum of predeclared supporting-task P75 durations as a proxy; Flex tasks retain explicit native-duration exceptions. The native environment checks success, irreversible failure, and timeout after every action, including within predicted chunks. Success includes any required release or dwell; incomplete episodes score zero. Lack of progress alone does not stop a rollout. Infrastructure-invalid episodes are excluded from policy scoring. Exact quantiles, budgets, and boundary rules are in Appendices B.6 and B.7.

The primary score is task-macro success within the $3\times$ limit; the limit is shared by all models and leaves room for slow but progressing policies. Success within $1\times$ P75, a prefix of the same rollouts, is a secondary score that rewards expert-like speed. Diagnostics distinguish acquisition from functional activation, stage transitions, and retention of achieved goals. Ordered progress credits the longest completed prefix of stages and does not replace native success. Every aggregate specifies its task/trial population and uncertainty resampling unit (Appendix B.8).

5 Experiments and Findings

Experiments address Core-50 execution, task generalization, and the contribution of demonstration scale, and then locate where episodes fail.

5.1 Models and Experimental Setup

We train eight multi-task checkpoints on the Core-50 demonstrations. The five vision-language-action (VLA) models are $\pi_{0.5}$ (Physical Intelligence et al., 2025), DM0.5 of the DM0 family (Yu et al., 2026), LingBot-VLA (Wu et al., 2026), HY-Embodied-0.5-VLA (Zhang et al., 2026), and Spatial Forcing (Li et al., 2025), which aligns the visual features of $\pi_{0.5}$ with a pretrained 3D foundation model during training. The three world-action models (WAMs), OpenWAM (Wang et al., 2026b), FastWAM (Yuan et al., 2026), and Motus (Bi et al., 2025), combine action prediction with modeling of future observations. Each checkpoint executes its integration's default action chunk per query (Table 2, Steps/plan). The selected checkpoints are $\pi_{0.5}$ at 120,000 updates, DM0.5 at 86,000, OpenWAM at 70,000, Motus and FastWAM at 100,000, Spatial Forcing at 90,000, HY-Embodied at 125,000, and LingBot-VLA at 151,000; they are fixed, not chosen by benchmark success. The single-task reference, Diffusion Policy (DP) (Chi et al., 2023), trains one 30,000-update model per Core-50 task. Without a predeclared routing protocol it is not run on Transfer-16; this is not a measured zero-shot failure. Pretraining data and compute differ across models, so the comparison does not isolate architecture.

5.2 How Well Do Current Policies Perform on Core-50?

Table 2 reports all 50 Core-50 tasks for the eight multi-task checkpoints (per task in Appendix C). For analysis we group the tasks by the hand skill they demand (Appendix D; Figure 3a).

Policies move objects and mechanisms; tools and precision fits remain unsolved. OpenWAM has the highest success, 29.1%, and picking the best checkpoint per task would give 35.4%. Success is ordered by what the task asks of the fingers beyond closing them. Pushing or flicking without a grasp (two tasks, 51% pooled over checkpoints), moving a hinge, drawer, lever, or lid (43%), and grasp–transport–release including handovers and object selection (35%) are the relatively better-performing groups in the suite; all nine tasks with mean success above 40% are of these kinds.

Table 2: Core-50 task-macro success (%), 100 episodes per task and checkpoint; 95% intervals in Table 11. Steps/plan: control steps (30 Hz) per policy query. Tasks ≥50%: number of the 50 tasks with at least 50% success. P/F/CR: primitive (17), functional (18), contact-rich (15) tasks. @1×: success within the expert p75 duration. Early: episodes ended by an irrecoverable failure. Stall: time-outs with no new checker state after 1× (Section 5.5). Shaded: best (lowest for Early, Stall).

Model	Steps/plan	SR	Tasks ≥50%	P	F	CR	@1×	Early	Stall
OpenWAM	16	**29.1**	**11**	40.8	**23.8**	**22.1**	8.3	**10.9**	**65**
FastWAM	16	26.4	9	**42.7**	20.2	15.3	10.2	13.0	72
$\pi_{0.5}$	30	24.9	10	40.4	14.6	19.7	**14.6**	14.1	78
DM0.5	30	21.4	9	37.7	10.0	16.7	10.9	13.9	80
LingBot-VLA	30	19.7	8	26.9	14.3	17.9	12.0	12.7	83
HY-Embodied	30	18.9	8	27.1	13.3	16.4	11.0	14.4	81
Motus	16	18.1	5	30.1	11.5	12.3	5.4	15.6	70
Spatial Forcing	16	12.4	3	17.9	8.5	10.9	5.6	16.5	75
DP (single-task)†	—	9.7	3	reference; 9.9 on 11 matched tasks					

†One 30k-step model per task; 38 of 50 task evaluations use an earlier goal sampler. Matched: ten retrained single-arm tasks and Open Latched Door ($\pi_{0.5}$: 29.1). HY-Embodied: 99 valid episodes on Place Wine Glass.

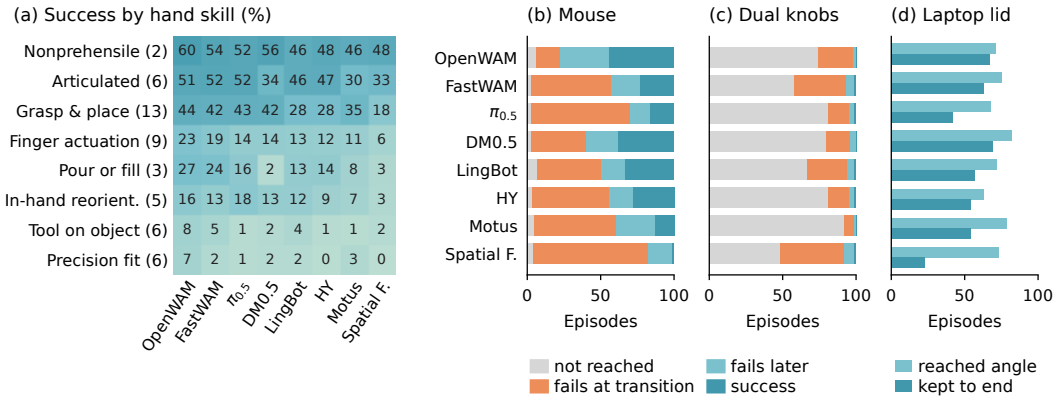


Figure 3: Core-50 diagnostics, eight multi-task checkpoints. (a) Task-macro success by hand-skill group (number of tasks in parentheses). (b)–(d) Episodes at the three transitions of Section 5.5; grey: transition never reached, orange: failed at it, light blue: failed after it. (b) Mouse moved 8 cm onto its pad before any click. (c) Both knobs within 5° of target at once (grey: at most one knob). (d) Laptop display ever at the requested angle, and still there at the end.

Operating a trigger, button, knob, slider, or shutter with the fingers drops to 14%, pouring to 13%, and regrasping or reorienting an object in or between the hands to 11%. Two groups are essentially unsolved: acting on a third object through a held tool (137 of 4,800 episodes, 2.9%; hammer, spoon, set square, and spatula together 9 of 3,200) and precision fits such as peg insertion, stacking, or face alignment (107 of 4,800, 2.2%). The best checkpoint on any of these twelve tasks reaches 21% (OpenWAM on Stir Pot); on ten of them none exceeds 14%. Data volume and duration do not explain the ordering: precision fits have more demonstrations per task (median 413) than grasp–transport–release (392), and with log demonstration count and log expert duration as covariates the tool, precision, in-hand, and finger-actuation groups stay 20–25 points below grasp–transport–release, with task-bootstrap intervals excluding zero (Appendix D).

Checkpoints differ in using a grasped object, not in grasping it. On grasp–transport–release tasks four checkpoints are within two points (OpenWAM 44, $\pi_{0.5}$ 43, FastWAM 42, DM0.5 42), and the OpenWAM–$\pi_{0.5}$ gap on primitive tasks is $+0.5$ points (paired task bootstrap 95% interval -6.9 to $+7.6$). The same gap on functional tasks is $+9.2$ ($+4.7$ to $+13.9$): Pour Seasonings 60 versus 34%, Move and Click Mouse 44 versus 16, Press Soap Pump 33 versus 9, Stir Pot 21 versus 4. Section 5.5 locates this gap at specific transitions: OpenWAM clicks before the mouse is ready in

16 of 100 episodes and $\pi_{0.5}$ in 67, OpenWAM keeps the laptop display at the requested angle in 67 and $\pi_{0.5}$ in 42, and OpenWAM completes the handover to the receiving hand in 41 and $\pi_{0.5}$ in 20. Two VLA checkpoints place fixed objects as well as the leaders but collapse when the instruction selects the object: HY-Embodied and LingBot-VLA score 55 and 61% on Place Cup on Plate ($\pi_{0.5}$ 62) but at most 22% on Place Ball in Bowl, Place Fruit in Bucket, and Place Largest Fitting Ball, where $\pi_{0.5}$ scores 35–75%. They barely read the object name: in an offline probe (Appendix D), naming the other ball moves the 1-s palm endpoint of $\pi_{0.5}$ and DM0.5 by twice the resampling change (15 and 17 versus 7 and 8 cm) but that of HY-Embodied and LingBot-VLA by no more (7–9 cm), and with the swapped name they still head for the original ball in 57 and 71% of first frames (DM0.5 10%). Motus is a strong picker (76% on Place Ball in Bowl) but near zero wherever a mechanism must be held open or a load carried (Open Micropipette Box 8%, Open Drawers 4%, Lift Carton 8%). Spatial Forcing is weakest in every group and, unlike all other checkpoints, worse on single-arm (3.8%) than on bimanual tasks (14.5%).

World-action models finish later but finish more. The three WAM checkpoints complete their successes at a median of 1.13–1.27 p75 durations versus 0.88–0.99 for the four 30-step VLA checkpoints, and more of their time-outs are still making progress after $1\times$ (14.5–18.7% of episodes versus 10.1–11.8%). Counting only completions within $1\times$ (Table 2, @$1\times$), $\pi_{0.5}$ leads with 14.6% and OpenWAM ranks sixth with 8.3%; 12 of the 28 checkpoint pairs change order between the two limits (Appendix D). The WAMs execute 16 control steps per query, but Spatial Forcing, a VLA that also executes 16, is not slow (median 1.09), so the query cadence alone does not explain the difference; @$1\times$ is therefore the secondary score (Section 4.2).

Reliability and feasibility. The 95% interval over tasks and episodes (22.8–36.0% for OpenWAM, Table 11) mainly reflects the task mix (episodes alone: 28.0–30.1%). Paired task resampling orders $\pi_{0.5}$ above DM0.5 in 99% of draws but HY-Embodied above Motus in only 61% (Appendix D). Leaving out any one task swaps at most HY-Embodied and Motus and moves no score by more than 1.4 points. Every task is completed by verified human demonstrations, and some checkpoint solves 47 of the 50 (Appendix B.2).

Single-task reference. DP averages 9.7% over the 50 tasks, but 38 of its task evaluations use an earlier goal sampler; on the 11 tasks evaluated under the current protocol it reaches 9.9%, against 22.7–34.5% for seven multi-task checkpoints and 9.3% for Spatial Forcing. This does not separate multi-task data sharing from pretraining or model scale.

5.3 Do Learned Skills Generalize Beyond the Training Task Set?

Table 3 groups Transfer-16 results by asset/relation transfer, skill composition, and deformable manipulation, with ordered progress crediting the longest completed prefix of stages. Transfer-16 is much harder than Core-50, and we rank checkpoints by ordered progress, which also credits partial completion. OpenWAM's 12.9% exceeds HY-Embodied, LingBot-VLA, and Spatial Forcing by 6.0, 3.7, and 2.6 points (paired intervals exclude zero); the other four are within 1.7 points. Every Transfer-16 task is verified in person by the data-collection operators (Appendix B.2), so these scores measure the policies, not the tasks; Section 5.5 locates where they fail.

5.4 Does Demonstration Scale Help?

We finetune $\pi_{0.5}$ separately on *Place Ball in Bowl* (A03) and *Squeeze Scissors* (D05) on nested subsets of 10, 50, 100, 200, and all 438 and 439 training trajectories, each run from the same pretrained initialization for 20,000 updates at batch size 32, with one training seed and the same 100 evaluation seeds, goals, and prompts per task (Appendix E).

Success rises at every tested size (Figure 4a): A03 from 7% to 83% and D05 from 1% to 44%, and going from 200 trajectories to the full pool still adds 18 and 26 points. The functional scissors task remains much harder. These task-local curves do not establish a cross-task scaling law or isolate an architectural cause; normalization uses the full training pool at every size.

Table 3: Transfer-16 results (%): task-macro success (SR) and ordered progress (Prog., Appendix B.8) within the declared time limit over the 7 Gen, 6 Comb, and 3 Flex tasks; 50 episodes per task and checkpoint (Stack Boxes 37 and Spray Shoe 43, whose missing resets fail for every checkpoint; at most two invalid episodes elsewhere). Gen02 logs no graded hinge progress and is excluded from Prog. DP is not run (Section 5.1). Shaded: best per column.

Model	Asset/relation		Skill composition		Deformable		All 16	
	SR	Prog.	SR	Prog.	SR	Prog.	SR	Prog.
$\pi_{0.5}$	0.0	14.0	0.3	11.7	0.0	8.3	0.1	11.9
DM0.5	**1.4**	9.2	**0.7**	**14.4**	0.0	8.8	**0.9**	11.2
LingBot-VLA	0.0	13.3	0.0	4.2	0.0	11.3	0.0	9.3
HY-Embodied	0.0	9.6	0.0	4.6	0.0	6.2	0.0	6.9
OpenWAM	0.9	14.8	0.0	12.5	0.0	10.2	0.4	**12.9**
Motus	0.3	**15.8**	0.0	9.5	0.0	**11.7**	0.1	12.4
FastWAM	0.0	12.5	0.0	11.3	0.0	11.3	0.0	11.8
Spatial Forcing	0.3	12.7	0.0	7.9	0.0	10.5	0.1	10.3

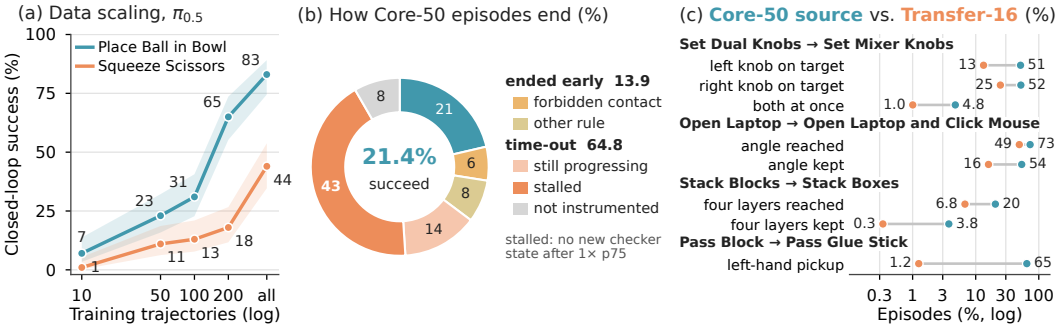


Figure 4: (a) Single-task data scaling for $\pi_{0.5}$, 20,000 updates per run (all: 438 and 439 trajectories); each point is one checkpoint on 100 closed-loop trials, with 95% Wilson bands over rollouts, not over training seeds or subsets. (b) How Core-50 episodes end, task-macro, mean of the eight multi-task checkpoints (each in Figure 7). (c) Matched hand transitions in Core-50 sources and Transfer-16, pooled over the eight checkpoints (800 source and 296–400 transfer episodes per task).

5.5 Why Do Episodes Fail?

Averaged over checkpoints, 21.4% of Core-50 episodes succeed, 13.9% end early, and 64.8% time out, mostly as stalls (Figure 4b): 65–83% of instrumented time-outs record no new checker state after 1× p75. The hands keep moving in a stall: in the last third of time-outs the commanded mean finger angle still reverses 3–4 times per 10 s while its range shrinks from 0.35–0.39 to 0.20–0.28 rad. Acquisition is not the main loss on Core-50. From native checker state we locate most remaining losses at three transitions of multi-finger bimanual control, pooling 800 episodes per task over the eight checkpoints (Figure 3b–d).

Forming a grasp can fire the function. The fingers that hold a tool often rest on its button, trigger, or handles, so closing the grasp can itself trigger an irreversible event. Move and Click Mouse ends as soon as a button is pressed before the mouse has moved 8 cm onto its pad. This ends 443 of 800 episodes, 404 of them before the mouse was ever ready and 297 before it had moved at all (Figure 3b); of the 362 episodes that reach the ready state without an early click, 196 succeed; the checkpoints differ mainly in the early clicks (16 for OpenWAM, 78 for Spatial Forcing). In Photograph Logo the first shutter press precedes aiming in 774 of 778 episodes, and 163 of the 185 episodes that ever aim with the right hand on the shutter succeed. In Water Plant the trigger is a separate finger joint and the grasp is clean, so the loss moves to that finger: of 514 episodes that aim the held sprayer, 238 never spray, 178 of them after moving the trigger at least halfway. In Squeeze Scissors the same fingers must hold and cut: failing episodes close the blades 1,368 times but score 182 cuts, because at 57% of the closings the scissors are not lifted and at 26% a handle contact has been lost. Where the functional part lies under the holding contacts, grasping fires it early; where

it is a separate finger, that finger is not driven to threshold at the right moment; where holding and actuating share fingers, the policy actuates only when it stops holding.

Each side reaches its target, but not at the same time. Set Dual Knobs requires both knobs within 5° of their targets at once, nearly still, held, and then released. Of 800 episodes, the left and right knob reach their target in 411 and 416, both at some point in 219, both at once in 38, and 5 succeed (Figure 3c). When the first knob arrives, the other is still a median 64° away; the first knob then leaves its 7° exit band in 570 of 608 episodes after a median 0.33 s, in 505 cases by overshooting and in 566 with the hand still in contact. The hands turn each knob through its target instead of holding it there while the other hand finishes. Timing alone is not the obstacle: in Pour Seasonings the two streams overlap in 306 of the 398 episodes where both hit their targets.

Reaching a goal and keeping it are different skills. In Open Laptop, Spatial Forcing and OpenWAM bring the display to the requested angle about equally often (73 and 71 of 100 episodes), but it is still there at the end in 23 and 67 (Figure 3d). Of the 154 pooled episodes that lose the angle, 148 end over-opened (median 31.5° past), all with the hand on the display: it is pushed through the target, not dropped. The Cover Pitcher lid stays inserted in 47 of the 86 episodes that insert it, and Stack Blocks keeps four layers in 30 of 164, with the hand already released at 70 of the 134 losses. In Pass Block, the receiver takes over alone in 188 of 367 episodes where both hands hold the block; without takeover the block drops in 175 of 179, a median 0.30 s after the giving hand lets go. Policies must stop pushing at the target and release only when the other hand holds.

Tools are bypassed, and failures are not shared across checkpoints. Line Up Blocks, Scoop Marbles, and Flip Sponge end on direct hand contact with the manipulated object in 81, 44, and 28% of episodes (2 successes in 2,400): policies pursue the object with the hand rather than through the tool. On the 41 fixed-goal tasks at least one checkpoint solves 62.4% of scenes, close to the 64.0% expected under independence (median pairwise $\kappa = 0.03$): the failed scenes differ by checkpoint, so no shared set of hard scenes explains the failures, although each checkpoint may have its own.

Transfer-16 fails at the same transitions, and new objects fail earlier. Every matched transition fails more often than in its Core-50 source (Figure 4c). In Spray Shoe grasps form in 232 of 344 episodes, but at most 3 of 43 per checkpoint complete a trigger cycle. On a solved source, a new object moves failure to acquisition: the left hand picks up the glue stick in 5 of 400 Pass Glue Stick episodes, against 40–88% per checkpoint in Pass Block.

Compositions add a fourth failure: steps are repeated or started early. After the laptop step (146 episodes), no episode brings the mouse onto its pad, and 111 push the display out of range again. The shutter is pressed before the fruit is in the bucket in 311 of 399 Place Fruit and Photograph Logo episodes, water runs in 107 of 400 Scrub Bottle Then Fill Cup episodes although none finishes scrubbing, and the box lid moves before the pitcher lid in 340 of 400 Cover Pitcher and Box episodes. Nothing marks a finished step, so completion must be inferred from the scene.

6 Discussion and Conclusion

DexOmni pairs 18,000 curated human demonstrations with Core-50 execution tests, the evaluation-only Transfer-16 suite, and native-stage diagnostics for comparing DP, VLA, and WAM policies. On Core-50 the best checkpoint reaches 29.1% task-macro success. Using a held tool, fitting parts to sub-centimetre tolerance, and actuating mechanisms with individual fingers are nearly unsolved, independently of demonstration count and duration, and two VLA checkpoints barely condition on the named object. Failures concentrate at three hand transitions, a grasp that fires a button early, two hands that reach their targets at different times, and reached goals that are pushed through or released; Transfer-16 compositions add a fourth, repeating completed steps or starting later ones early. Separating holding from actuating contacts, stopping and holding at a target, releasing only when the partner hand holds, and recognizing completed steps are concrete targets for future policies. Because every task in both suites is verified in person by the data-collection operators (Appendix B.2), the unsolved tool, precision, and composition tasks measure what current policies cannot yet do.

Ethics statement

The demonstrations were collected by teleoperation in simulation from operators who were informed of the purpose of the recordings and agreed to their release. The released trajectories contain simulated observations, robot states, commanded actions, and task language; they include no images, audio, or names of the operators, and the operator shown in Figure 5 consented to the photograph. Real-robot rollouts were recorded in a laboratory without bystanders. Task designs and assets adapted from prior benchmarks are credited in Appendix A.1. The benchmark evaluates general-purpose manipulation of household objects in simulation; we are not aware of applications that raise concerns beyond those of robot learning in general, and the reported results do not certify the safety of any physical deployment.

Reproducibility statement

Section 5.1 names every evaluated checkpoint and its training step. Section 4.2 and Appendices B.6–B.8 give the control frequency, time limits, stopping rules, and metric definitions; Appendices A and G list all 66 tasks with their instructions and success conditions; and Appendix B describes collection, curation, re-rendering, and the prompt sampler. Per-task results with the snapshot identity, replacement rules, and bootstrap procedure are in Appendix C, the hand-skill grouping and reliability checks in Appendix D, and the scaling settings in Appendix E. The dataset, task definitions, evaluation harness, and machine-readable result record will be released; recorded real-robot rollouts are included as supplementary videos.

AI use statement

In this work, we used generative AI tools (LLM-based coding and writing assistants) to refine the implementation of simulation tasks, to check the evaluation infrastructure and analysis code, and to generate candidate instruction paraphrases, which were screened by the checks in Appendix B.4. For the manuscript, the tools were used to polish the text, organize the content, and prepare figures and tables. The data analysis and the interpretation of results were carried out by the authors. The tools did not produce human demonstrations or evaluation outcomes: every reported number is computed from recorded episodes by scripts whose outputs the authors checked. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. DexArt: Benchmarking generalizable dexterous manipulation with articulated objects. *arXiv preprint arXiv:2305.05706*, 2023. URL https://arxiv.org/abs/2305.05706.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025. URL https://arxiv.org/abs/2512.13030.

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. *arXiv preprint arXiv:2506.18088*, 2025. URL https://arxiv.org/abs/2506.18088.

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. *arXiv preprint arXiv:2607.04434*, 2026. URL https://arxiv.org/abs/2607.04434.

Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuang Jiang, Zongqing Lu, Stephen Marcus McAleer, Hao Dong, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=D29JbExncTP.

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. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2303.04137.

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. *arXiv preprint arXiv:2302.04659*, 2023. URL https://arxiv.org/abs/2302.04659.

Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, Yashraj Narang, Jean-Francois Lafleche, Dieter Fox, and Gavriel State. DeXtreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5977–5984, 2023. doi: 10.1109/ICRA48891.2023.10160216. URL https://doi.org/10.1109/ICRA48891.2023.10160216.

Guanqi He and Wentao Zhang. WujiHand Retargeting. Software repository: https://github.com/wuji-technology/wuji-retargeting, 2026.

HTC Corporation. VIVE Tracker (3.0). Manufacturer product page: https://www.vive.com/us/accessory/tracker3/, 2021. Accessed: 2026-08-26.

Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. OPEN TEACH: A versatile teleoperation system for robotic manipulation. *arXiv preprint arXiv:2403.07870*, 2024. URL https://arxiv.org/abs/2403.07870.

Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Jim Fan, and Yuke Zhu. DexMimicGen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *IEEE International Conference on Robotics and Automation*, 2025. URL https://arxiv.org/abs/2410.24185.

Sarthak Kamat, Adam Rashid, Satvik Sharma, Aseem Doriwala, Chelsea Finn, Phillip Isola, and C. Karen Liu. Pre-training visual dexterity in simulation. *arXiv preprint arXiv:2608.15917*, 2026. URL https://arxiv.org/abs/2608.15917.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2403.12945.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, et al. BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, 2022. URL https://proceedings.mlr.press/v205/li23a.html.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025. URL https://arxiv.org/abs/2510.12276.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023. URL https://arxiv.org/abs/2306.03310.

Qingtao Liu, Yu Cui, Zhengnan Sun, Gaofeng Li, Jiming Chen, and Qi Ye. VTDexManip: A dataset and benchmark for visual-tactile pretraining and dexterous manipulation with reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jf7C7EGw21.

Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, Scott Reed, Ken Goldberg, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. In *Proceedings of Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2503.24361.

MANUS. MANUS Prime Series Infokit 2019. Manufacturer infokit: https://a.storyblok.com/f/77127/x/9e90554ca1/infokit-v5.pdf, 2019. Accessed: 2026-08-26.

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

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. In *Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2406.02523.

NVIDIA Corporation. NVIDIA Isaac Sim 5.1 documentation. https://docs.isaacsim.omniverse.nvidia.com/5.1.0/, 2026. Accessed: 2026-08-26.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation*, 2024. URL https://arxiv.org/abs/2310.08864.

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. *arXiv preprint arXiv:1808.00177*, 2018. URL https://arxiv.org/abs/1808.00177.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

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. *arXiv preprint arXiv:2307.04577*, 2023. URL https://arxiv.org/abs/2307.04577.

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. *arXiv preprint arXiv:1709.10087*, 2017. URL https://arxiv.org/abs/1709.10087.

ROKAE Robotics. Humanoid Force-Controlled Arm AR Series. Manufacturer product page: https://www.rokae.com/en/product/show/595/Humanoid-Force-Controlled-Arm-AR.html, 2026. Accessed: 2026-08-26.

Zhipeng Tang, Sihang Chen, Sha Zhang, Peihao Yang, Yan Liu, Wentao Zhao, Xinrui Liu, Rui Huang, Wensheng Du, Yuting Huang, Jiajun Deng, Lidian Wang, Yuan Zhang, and Yanyong Zhang. LabDex: A hierarchical benchmark for dexterous manipulation in laboratories. *arXiv preprint arXiv:2608.18618*, 2026. URL https://arxiv.org/abs/2608.18618.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. DexCap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024. URL https://arxiv.org/abs/2403.07788.

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. *arXiv preprint arXiv:2605.16257*, 2026a. URL https://arxiv.org/abs/2605.16257.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, et al. OpenWAM: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026b. URL https://arxiv.org/abs/2609.07398.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic VLA foundation model. *arXiv preprint arXiv:2601.18692*, 2026. URL https://arxiv.org/abs/2601.18692.

Wuji Technology. Wuji Hand: Product Introduction. Manufacturer documentation: https://docs.wuji.tech/docs/en/wuji-hand/v1/overview/, 2026. Accessed: 2026-08-26.

Zhenjie Yang, Yideng Zhang, Dongjie Zhang, Chenyu Jiang, Xianshuai Liu, Yufeng Li, Zuhao Ge, Xingyu Jiao, Zheng Zhang, Kaiyu He, He Wang, Yuwen Zhong, Yi Deng, Muyun Jiang, Xianliang Huang, Haisheng Su, Donghang Zhang, Jian Zhang, Xue Yang, Hongyang Li, Zuxuan Wu, Yu-Gang Jiang, Xiaosong Jia, and Junchi Yan. Bench2Dex: Benchmarking visuo-tactile bimanual dexterous manipulation across dexterous hands. *arXiv preprint arXiv:2609.15726*, 2026. URL https://arxiv.org/abs/2609.15726.

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. *arXiv preprint arXiv:2607.08751*, 2026. URL https://arxiv.org/abs/2607.08751.

En Yu, Haoran Lv, Jianjian Sun, Kangheng Lin, Ruitao Zhang, et al. DM0: An embodied-native vision-language-action model towards physical AI. *arXiv preprint arXiv:2602.14974*, 2026. URL https://arxiv.org/abs/2602.14974.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

He Zhang, Lingzhu Xiang, Haitao Lin, Zeyu Huang, Minghui Wang, Dingyan Zhong, et al. Hy-Embodied-0.5-VLA: From vision-language-action models to a real-world robot learning stack. *arXiv preprint arXiv:2606.14409*, 2026. URL https://arxiv.org/abs/2606.14409.

A Task Inventory and Taxonomy

Table 4 lists all 66 tasks in DexOmni: Core-50 provides 50 tasks for learning and evaluation; Transfer-16 provides 16 additional tasks exclusively for generalization evaluation. Transfer-16 contains 6 compositional tasks (EVAL-Comb), 7 generalization tasks (EVAL-Gen), and 3 tasks in the EVAL-Flex family. Evaluation-only tasks are excluded from the dataset for policy training. Within each split, task identifiers follow alphabetical order of the English task names.

Source superscripts distinguish task-design adaptation (J: DexJoCo; V: DexVerse) from asset reuse (JA and VA, respectively). The sources and modifications are detailed in Appendix A.1. Unmarked entries do not imply that every asset or component is original.

The three label columns are the axes of the taxonomy in Figure 2. Interaction Type records the dominant manipulation requirement. Asset Type includes all applicable types of task-relevant assets, including designated targets and receptacles, according to their current simulation models and active degrees of freedom. The robot, generic support surfaces, and background assets are excluded. Task Structure describes the organization of semantic goals and operations; approaching, grasping, and releasing do not by themselves create separate stages.

Table 4: Complete task inventory and three-axis taxonomy of DexOmni; shaded rows are the evaluation-only Transfer-16 tasks. Abbreviations are defined below the table.

Task ID	Task Name	Interaction Type	Task Structure	Asset Type
01	Align Blocks in Midair	CR	SS	RB
02	Connect Charger and Cable	CR	LH	RB
03	Cover Box	CR	SS	RB
04	Cover Pitcher	CR	SS	RB
05	Fill Cup	F	LH	RB, AR
06	Fit Device Cover	CR	SS	RB
07	Flip Sponge	F	SS	RB
08^{J}	Hammer Nail	F	SS	RB, AP
09^{JA}	Insert Peg	CR	SS	RB
10^{V}	Lift Carton	CR	SS	RB
11	Line Up Blocks	F	RP	RB
12^{J}	Load and Start Microwave	F	LH	RB, AR, AP
13VA	Load Cans	P	SS	RB
14^{J}	Move and Click Mouse	F	LH	RB, AP
15	Open Drawers	P	SS	AP
16^{V}	Open Laptop	P	SS	AR
17	Open Latched Door	F	LH	AR
18	Open Micropipette Box	P	SS	AR
19	Orient Cup Upright	P	SS	RB
20	Pass and Orient Pen	CR	LH	RB
21	Pass Block	P	LH	RB
22	Pass Screwdriver	P	LH	RB
23^{J}	Photograph Logo	F	SS	RB, AP
24	Place Ball in Bowl	P	SS	RB
25	Place Cup on Plate	P	SS	RB
26	Place Fruit in Bucket	P	SS	RB
27	Place Largest Fitting Ball	P	SS	RB
28	Place Tableware	P	SS	RB
29	Place Wine Glass	P	SS	RB
30	Pour Seasonings	F	SS	RB
31	Prepare Coffee	F	LH	RB, AR, AP
32	Press Soap Pump	F	SS	RB, AP
33	Reposition Marker Grasp	CR	SS	RB

Continued on next page

Table 4: Complete task inventory and three-axis taxonomy of DexOmni (continued).

Task ID	Task Name	Interaction Type	Task Structure	Asset Type
34	Roll Ball Through Gate	P	SS	RB
35	Rotate Propeller	CR	RP	AR
36	Scoop Marbles	F	SS	RB
37	Scrub Bottle	CR	RP	RB
38	Set Dual Knobs	F	SS	AR
39	Set Timer	F	RP	AP
40^{V}	Slide Knife Blade	CR	SS	AP
41	Sort Nested Cups	CR	LH	RB
42	Sort Objects	P	LH	RB
43	Squeeze Scissors	F	RP	AR
44	Stack Blocks	P	RP	RB
45	Stir Pot	F	RP	RB
46VA	Store Charger	P	LH	RB, AP
47	Trigger Drill on Block	F	SS	RB, AP
48	Unfold Fan	CR	SS	AR
49	Unfold Ruler	CR	SS	AR
50^{J}	Water Plant	F	SS	RB, AR
EVAL-Comb01	Cover Pitcher and Box	CR	RP	RB
EVAL-Comb02	Open Laptop and Click Mouse	F	LH	RB, AR, AP
EVAL-Comb03	Pass Block and Close Box	CR	LH	RB
EVAL-Comb04	Place Fruit and Photograph Logo	F	LH	RB, AP
EVAL-Comb05	Scrub Bottle Then Fill Cup	F	LH	RB, AR
EVAL-Comb06	Season and Stir	F	LH	RB
EVAL-Gen01	Cover Lunchbox	CR	SS	RB
EVAL-Gen02	Open Trash Bin	P	SS	AR
EVAL-Gen03	Orient Bottle Upright	P	SS	RB
EVAL-Gen04	Pass Glue Stick	P	LH	RB
EVAL-Gen05	Set Mixer Knobs	F	RP	AR
EVAL-Gen06	Spray Shoe	F	SS	RB, AR
EVAL-Gen07	Stack Boxes	P	RP	RB
EVAL-Flex01	Compress Sponge into Mug	CR	SS	RB, DF
EVAL-Flex02	Fold Towel Twice	CR	RP	DF
EVAL-Flex03	Pack Pillow	CR	SS	RB, DF

Abbreviations. **Interaction Type:** P = Primitive; F = Functional; CR = Contact-rich. **Task Structure:** SS = Single-stage; RP = Repetitive; LH = Long-horizon (multi-stage). **Asset Type:** RB = Rigid body; AP = Articulated (prismatic); AR = Articulated (revolute); DF = Deformable.

Classification notes. Primitive covers object transport or reorientation whose success is decided by reaching a target configuration, including pushing, pulling, or rolling an object and moving an articulated part along its free degree of freedom (e.g., opening a drawer or a hinged lid). Functional requires the purposeful use of a tool, appliance, or control interface (buttons, knobs, levers, faucets, triggers) to produce an effect. Contact-rich denotes tasks dominated by fine contact coordination, such as regrasping, precise mating, sustained contact-force regulation, coordinated bimanual articulation, or controlled deformation. RB describes each non-articulated rigid asset and does not limit a task to one object. Rigid links of articulated assets are not counted separately. Deformable labels follow the current simulation model; visual liquid effects do not add a DF label. In Task 08, the nail has a prismatic joint relative to the wood block, and the hammer is a separate rigid object. EVAL-Flex03 combines a deformable foam pillow with a rigid carton. SS denotes a single semantic goal. RP denotes repeated applications of the same kind of manipulation operation, whereas LH requires distinct, dependent subgoals. Routine setup and terminal activation, release, or waiting do not by themselves turn a repetitive task into LH. Task 39 (Set Timer) is classified as RP because its main

manipulation consists of setting the minute and second values through button operations; starting the timer, withdrawing the hands, and observing a countdown tick are terminal completion conditions.

A.1 Task Provenance and Adaptation

DexOmni builds on the DexJoCo toolkit and acknowledges prior task designs from DexJoCo (Wang et al., 2026a) and DexVerse (Yao et al., 2026). Table 5 separates task-design lineage from the reuse of particular assets. The J and V markers identify documented adaptations; JA and VA identify asset provenance without asserting that the whole task was transferred. Shared infrastructure and generic similarities, such as grasping or stacking, are insufficient to establish task-specific lineage.

Table 5: Acknowledged task-design and asset sources. IDs follow Table 4. All listed tasks use our platform's execution and observation pipeline; source attribution does not imply identical reset distributions or success predicates.

ID	Upstream task or asset	Adaptation or reuse in DexOmni
08 J	DexJoCo: Hammer nail	Adds a supporting hand holding the wood block and a task-specific nail-depth condition.
12 J	DexJoCo: Bimanual microwave	Adapts the loading, door-closing, and activation sequence to the collection platform.
14 J	DexJoCo: Click mouse	Requires moving the mouse onto its pad before a permitted click.
23 J	DexJoCo: Bimanual photograph	Retains camera aiming and shutter activation with platform-specific contact and execution logic.
50 J	DexJoCo: Water plant	Adds grasp, trigger-contact, and aim-sensitive spray conditions.
10 V	DexVerse: BimanualLiftCarton	Adapts bilateral lifting to lower-side support and a three-second airborne hold.
16 V	DexVerse: OpenLaptop	Adapts the opening task to the ROKAE–Wuji platform; the released task uses a replacement laptop asset and a 60-degree target.
40 V	DexVerse: SlideUtilityKnife	Adapts blade sliding to coordinated hand roles and target-conditioned motion; the collection asset was replaced.
09 JA	IndustReal peg/socket via DexJoCo	Reuses the peg/socket asset family with a separately implemented single-arm insertion task.
13 VA	YCB soup can via DexVerse	Reuses the can asset in a new synchronized cross-loading task with a project-authored partitioned bin.
46 VA	Drawer and charger via DexVerse	Reuses the drawer mechanism and charger; adds a replacement cabinet appearance and a hold–store–release sequence.

Compositions and Transfer-16. Source credit also follows reused components into new evaluations. EVAL-Comb02 combines the adapted laptop and mouse skills, EVAL-Comb04 reuses the photograph skill, and EVAL-Gen06 transfers the spray interaction from Water Plant. These are compositions or task variants of the credited components, rather than entire tasks taken from an upstream benchmark. The evaluation-only designation specifies exclusion from policy-training demonstrations; it does not imply that all constituent skills or assets are new.

Attribution boundaries. Task names alone do not establish reuse. For example, Prepare Coffee and Squeeze Scissors have related counterparts in DexVerse but different documented workflows or asset lineages; we do not label them as direct ports on that basis. Historical asset names also require care: some locally authored assets retain a `dexverse_authored` prefix, while several preliminary DexVerse assets were replaced before collection. The source markers above describe the documented task lineage or retained asset reuse, not every historical candidate considered during development.

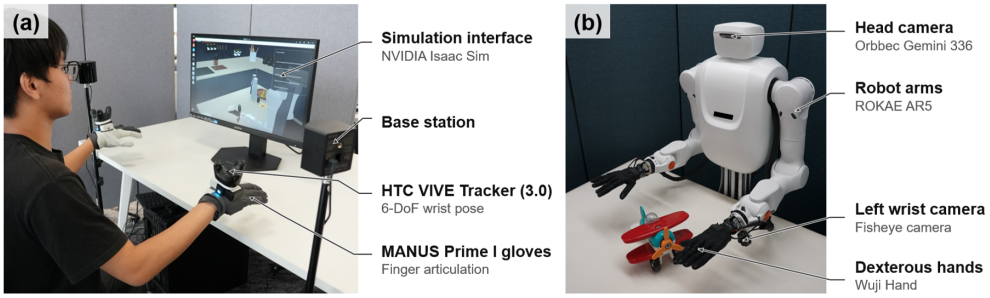


Figure 5: Human demonstration collection. Gloves record finger articulation and wrist trackers record hand motion. Retargeted commands drive the simulated arm–hand system while the operator observes the scene.

B Data, Platform, and Protocol Details

B.1 Collection, Dataset Identity, and Exposure

Demonstrations are collected in NVIDIA Isaac Sim 5.1 (NVIDIA Corporation, 2026). MANUS Prime I gloves capture finger articulation, and wrist-mounted HTC VIVE Trackers provide global wrist poses (MANUS, 2019; HTC Corporation, 2021). After frame registration and temporal alignment, wrist motion and a palm-centered 21-keypoint hand representation are converted to commands. The arm branch produces feasible end-effector targets through robot kinematics; Wuji-retargeting maps hand keypoints to joint targets, with 20 outputs for the first-generation Wuji Hand (He & Zhang, 2026). Simulation records synchronized observations, measured state, commanded actions, language, and source metadata. The collection setup is shown in Appendix B, Figure 5.

Selection uses task-group quotas of 500, 300, or 100 trajectories, favoring shorter eligible demonstrations within each quota. This is a curated distribution, not an unbiased sample of all collected behavior. Three stages of one multi-stage source task share a quota, giving 52 quota groups. Three simultaneous camera views do not multiply the interaction duration. At least ten operators contributed over 1,050 collection sessions. Across the package the median trajectory has 262 steps (10th–90th percentile 96–705); per Core-50 task, training trajectories range from 46 to 463 (median 268) and expert P75 durations from 2.5 to 49.2 s (median 12.7 s).

The released dataset contains 18,000 curated trajectories, comprising 15,178 training and 2,822 validation trajectories. Split membership is defined at trajectory/session level; re-rendered derivatives inherit source membership. Normalization fitted for this package uses training data. A checkpoint that inherits another normalization or has historical exposure retains that provenance.

Table 6: Distinct counting units in the dataset and catalog.

Unit	Count
Curated demonstration trajectories	18,000
Training / validation trajectories	15,178 / 2,822
Recorded time steps	6,053,555
Camera videos (three per trajectory)	54,000
Native task identities / quota groups	54 / 52
Core-50 / Transfer-16 tasks	50 / 16

The evaluation catalog contains Core-50 tasks and excludes Chinese Chess for every model. This scope exclusion does not remove demonstrations from the 18,000-trajectory package or rewrite any checkpoint's training provenance. The mapping record links paper task, source task/cohort, goal, asset identities, episodes, and executable evaluation task. Trajectory counts in this paper refer to the curated 18,000-trajectory package; the raw collection before deduplication and curation is not the training size of any checkpoint.

Table 7: Visual, language, and physical variation and the labels each must preserve. Geometry and dynamics changes require evidence beyond visual state replay.

Factor	Examples	Must preserve
Appearance	Material, lighting, background, ceiling	Preserve recorded interaction and task-relevant visual semantics.
Camera	Extrinsic perturbation about calibration	Preserve timing; transform camera-frame labels if affected.
Language	Canonical instruction and paraphrases	Preserve object, numerical goal, direction, order, and constraints.
Task geometry	Object pose, goal relation, native reset	Retain each source task's joint sampling distribution and compatible actions.
Dynamics	Mass, friction, compliance, actuation	Requires valid re-execution; not established by frozen-state re-rendering.

B.2 Task Verification

Tasks and demonstrations pass a written collection-and-review procedure with separated roles: data-collection operators record and self-check episodes, anonymous reviewers judge them, and a third person arbitrates disagreements; nobody reviews their own data. Each task has a task card that fixes the goal, the standard initial and final states, completion evidence independent of the program's success flag (object poses, joint states, or images), allowed and forbidden actions with known shortcuts, the time limit, and the environment, task, and checker versions with their code hashes. A new task or major revision is first piloted with at least 20 operator episodes, all reviewed by hand and covering success, failure, time-out, reset, and boundary states; it enters collection only with no severe defect, at least 98% agreement between labels and the objective evidence, and unambiguous rules. During collection the environment and checker are frozen by hash. A checker change passes reproduction, impact analysis, two-person review, regression tests on normal, failing, boundary, and known-defect cases, and a new fully reviewed pilot; affected episodes are replayed, relabeled, or quarantined. Every recorded episode passes automatic checks of completeness, version hashes, action plausibility, agreement between the success flag and the final physical state, and duplication. Stratified random blind review then covers about 20% of each batch (all of the first batch of a new task or version), and a further 5–10% of passed episodes are blindly re-reviewed. A success flag without objective completion quarantines the task as a suspected checker defect until it is repaired and piloted again.

Every Core-50 and Transfer-16 task was verified in person by the data-collection operators in the released environment before evaluation. Each Core-50 task is also completed by its curated demonstrations, and some multi-task checkpoint solves 47 of the 50 tasks.

B.3 Randomization and Re-rendering Records

Final-package capture records use 32 samples per pixel; higher-sampling presentation renders are identified separately. The implemented bank includes four table materials, three background configurations, and three ceiling styles, with texture scale/orientation, surface roughness/tint, and illumination direction, intensity, and color temperature variations. One verified source camera configuration samples each camera once per trajectory within ± 2 mm translation and $\pm 0.3^\circ$ rotation per axis, retaining intrinsics.

Background changes can be applied during replay; geometry and dynamics changes may require different actions and valid re-execution. Table 7 states what each factor must preserve.

Episode records identify source trajectory/runtime, render revision, appearance seed/configuration, and camera configuration. Appearance is sampled per trajectory and remains fixed during replay. Independently resampling appearance every frame is outside this static-domain protocol; any temporal sensor noise requires a separate model.

Camera perturbations are composed with task-specific nominal calibration. Within the Core-50 evaluation, which of the four table materials, three colour palettes, or two roughness levels is drawn does not measurably change success (Appendix D); these factors are inside the training distribution, so this is an invariance check, not a visual-generalization result.

B.4 Language Bank and Sampling Records

The 91 active pairs yield 2,912 training and 728 held-out variant entries. For example, "Pick the apple and place it into the bucket" and "Grab the apple and put it in the bucket" preserve the same goal. Replacing apple with banana, changing a knob angle, or changing operation order instead requires an episode with the corresponding executable goal. Language sampling does not resample the scene, cameras, or physics.

The frozen language bank contains 108 entries, of which 91 native-task/canonical-instruction pairs occur in the final 18,000-trajectory manifest. Each entry has 32 training and eight held-out variants. For the 91 active pairs, held-out strings have no exact overlap with any active canonical or training string. This is a lexical disjointness check; semantic equivalence requires preserving the episode's target and constraints. Inactive bank entries are not counted as demonstrated goals.

The packaged sampler hashes the language seed, epoch, episode identity, frame index, draw index, and split to choose an instruction. It offers training, canonical, and unseen modes. The reference training mixture is 20% canonical and 80% training paraphrases; fixed validation uses the run's declared language mode. The model record must identify the language bank revision, actual sampling policy, prompt preprocessing, and whether conditioning is text, a structured goal, or a fixed task identifier. Appendix D examines whether policies condition their actions on the object specified in the instruction.

B.5 Physical System and Alignment Record

Figure 1 compares real wrist/arm recordings with a released-training-video wrist view and a separate simulated arm detail. The views show finger contours, palm texture, and the orange-gold joint cover in its surrounding housing. Images are cropped and scaled only. Different poses and illumination make these illustrative appearance comparisons, not measurements of calibration, dynamics, or policy transfer.

The ROKAE AR5-08 has seven degrees of freedom, a rated payload of 5 kg, a 683 mm reach, and manufacturer-specified repeatability of ± 0.1 mm (ROKAE Robotics, 2026). The Wuji Hand has 20 actively actuated joints, four per finger, with position control over USB 2.0 (Wuji Technology, 2026). Its specified 1 kHz low-level frequency is distinct from policy inference and host command rates.

The correspondence record covers hand/robot geometry and limits, asset scale, camera calibration, state/action coordinate frames, joint/side ordering, timing, initial conditions, and success. Calibration data are disclosed independently of target-task demonstrations.

B.6 Execution, Scoring, and Validity

Simulation uses the fixed native end-effector execution protocol and checkpoint-owned representation, preprocessing, and normalization. Actions are executed at 30 Hz, with each action advancing the simulation by exactly 1/30 s; integration steps are recorded separately. The small source cohort collected with another controller is identified rather than assumed equivalent.

Time budgets are fixed before policy evaluation and shared across models. For a Core-50 task i, let $F_{i,(k)}$ denote its demonstration lengths in frames, sorted in ascending order, with zero-based index k. The frozen duration table uses the non-interpolated quantile

$$q_i = F_{i,(\min(n_i-1,\lfloor 0.75 n_i \rfloor))}, \qquad H_i = 3q_i/30, \qquad B_i = \lceil 30 H_i \rceil. \tag{1}$$

Here H_i is the budget in simulated physical seconds and B_i is the maximum number of executed actions. Source demonstrations are 30 fps; evaluation advances exactly $1/30$ s per action, verified from the physics timestep and substeps. Inference latency, rendering time, and action-chunk length do not redefine this physical budget. The released table is computed from the curated package's source cohorts, including its trajectory validation partition, without using evaluated policy outcomes.

For the 13 non-Flex Transfer-16 tasks, no target-task demonstration distribution exists. We instead use $H_i = 3 \sum_{j \in \mathcal{S}_i} q_j/30$, where $\mathcal{S}_i$ is a predeclared mapping to supporting atomic tasks. This is a duration proxy, not the measured P75 of composed trajectories. For example, *Open Laptop and*

Click Mouse uses $(196 + 125)$ source frames, giving 32.1 s or 963 actions. The three Flex tasks retain explicit native budgets: 240 s for sponge compression and 180 s each for towel folding and pillow packing. Appendix B.7 lists all Transfer-16 budgets and their mappings.

After every executed action, the native environment checks success, task-defined failure, and timeout. The rollout stops at the first terminal outcome, including within an action chunk; unused predicted actions are discarded. Success requires the complete native predicate, including any release, stability, or dwell requirement. Native failure terminates with zero success; exhausting the budget without success also scores zero. Success reached on the final permitted action is accepted. There is no generic early stop for apparent stagnation or for completing only an intermediate stage. Task-specific irreversible violations retain their documented native rules.

An invalid policy output, such as a nonfinite action, is recorded as a policy failure under the declared adapter. Simulator defects, rejected resets, process interruption, and infrastructure failures are recorded separately; they do not become numerical policy failures. Wall-clock watchdogs bound infrastructure execution only. Results retain executed steps, physical time, terminal reason, validity, and protocol revision; historical longer-budget successes cannot be relabeled as successes within the new budget.

Primary scoring follows the task-native success predicates. Lift Carton requires bilateral lower-side support, lifting the carton clear of the table, and a continuous three-second airborne hold without dropping it.

Infrastructure interruption, nonfinite simulator state, and invalid task implementations are separate from policy failures. A verified nonfinite policy action is recorded as a policy failure under the declared adapter. Results identify their environment, renderer, controller, duration, and checkpoint revisions. A repaired environment requires fresh target-task trials. Missing tasks and applicable model/task coverage remain explicit in aggregates.

B.7 Transfer-16 Duration Map

Table 8 reports the 30 Hz protocol. The source-frame column sums each mapped atomic task's P75 before multiplying by three. Task validity is established by the verification in Appendix B.2; a repaired task requires fresh trials under its qualified implementation. Native source names and paper IDs differ; the task inventory (Appendix A.1) gives the correspondence.

Table 8: Transfer-16 physical-time budgets. Flex budgets (shaded) are native exceptions and are not described as P75 estimates.

Task ID	Supporting Core-50 tasks	P75 frames	Seconds	Actions
Comb01	Cover Pitcher + Cover Box	$382 + 210$	59.2	1776
Comb02	Open Laptop + Move and Click Mouse	$196 + 125$	32.1	963
Comb03	Pass Block + Cover Box	$195 + 210$	40.5	1215
Comb04	Place Fruit in Bucket + Photograph Logo	$110 + 158$	26.8	804
Comb05	Scrub Bottle + Fill Cup	$259 + 530$	78.9	2367
Comb06	Pour Seasonings + Stir Pot	$236 + 547$	78.3	2349
Gen01	Cover Box	210	21.0	630
Gen02	Open Micropipette Box	125	12.5	375
Gen03	Orient Cup Upright	74	7.4	222
Gen04	Pass Block	195	19.5	585
Gen05	Set Dual Knobs	431	43.1	1293
Gen06	Water Plant	139	13.9	417
Gen07	Stack Blocks	387	38.7	1161
Flex01	Native exception: Compress Sponge into Mug	–	240	7200
Flex02	Native exception: Fold Towel Twice	–	180	5400
Flex03	Native exception: Pack Pillow	–	180	5400

B.8 Diagnostic Definitions and Statistical Reporting

Task success and ordered progress. Primary scores are per-task success and macro-averages over declared task sets. Native executable success predicates and physical-duration holds determine completion. For a task with K_i stages, ordered progress is $P_e = L_e/K_i$, where L_e is the longest valid ordered prefix completed in episode e. Stages are defined by semantic operations; merely approaching or touching an object does not automatically complete a functional stage. Progress does not override the final success predicate.

Task-specific diagnostics. Functional completion separates acquiring a tool from operating it. Stability measures include drops, slip, and displacement of an object supported by the other hand when these quantities are observable. Deformable tasks additionally report task-specific geometric progress. Contact-force measures require validated sensors or simulator measurements; command magnitude is not treated as measured force. Diagnostics are reported on applicable tasks without an arbitrary cross-task composite.

Capability-specific diagnostics. We distinguish tool acquisition from functional activation; requested numeric setting from any actuator movement; and articulated travel from mere object contact. Diagnostics include target-angle/time error, valid cycle count, joint travel or unlocking progress, object-selection and relation accuracy, and bimanual synchronization where the native task supports them. Capability labels overlap and use disclosed membership; their averages are not added into a new overall score.

Trials and reporting. Trials preserve task-local resets, assets, source/goal weights, state/action semantics, and checkpoint-owned preprocessing and normalization. Simulation action execution uses 30 Hz with the corresponding physical time per action; integration steps and inference chunks are recorded separately. Independent evaluation seeds are shared across compatible model comparisons. Binary success receives Wilson intervals; aggregate uncertainty is reported with the resampling unit specified. Missing or environment-invalid trials are not numerical policy failures. Every aggregate identifies its task and trial coverage. Checkpoint selection, benchmark-development exposure, and historical data exposure are disclosed in the run record.

Native event-arrival rates and ordered completion scores are different quantities. A conditional transition rate requires joint, order-consistent events, not division of unrelated stage marginals. Retention diagnostics identify both the achieved condition and the subsequent observation interval; success-triggered termination can give episodes unequal exposure. Recorded action commands are not measured wrist motion or contact force.

B.9 Task Partitions and Generalization Design

A task is defined by its environment, initial-state distribution, observations, actions, success predicate, and physical time budget:

$$\mathcal{T}_i = (\mathcal{E}_i, p_i(s_0), \mathcal{O}_i, \mathcal{A}_i, G_i, H_i). \tag{2}$$

We distinguish training trajectories, held-out trajectories from training tasks, fresh closed-loop trials on those tasks, and evaluation-only tasks. Trajectory validation measures action prediction; fresh closed-loop trials measure execution under the task's native training-source distribution. Transfer-16 tasks contribute no demonstrations to benchmark training or task-specific finetuning. Their absence from this dataset does not establish their absence from a foundation model's external pretraining.

The evaluation catalog contains six EVAL-Comb, seven EVAL-Gen, and three EVAL-Flex tasks. These implementation groups support complementary generalization questions rather than a set of perfectly orthogonal factors. Each task must specify the supporting training skills and assets, the held-out change, and its executable goal. Multiple changes are identified when they co-occur.

Asset and relation transfer. This dimension tests the use of skills with new objects, asset pairings, or goal relations. An unseen combination of familiar assets is distinguished from a new instance or category. For example, covering a new container and orienting a different object test transfer of related functional behavior, but whether an asset is actually familiar is determined by source and asset identities, not by its task name.

Table 9: Training support and evaluation changes. Relations shown here describe task semantics; the source/asset mapping specifies actual exposure.

Dimension	Training support	Evaluation question
Asset/relation	Object interaction, covering, orienting, passing	Can a skill operate on a new target or asset relation?
Skill composition	Opening a laptop; moving and clicking a mouse; scrubbing; filling	Can trained behaviors be connected in a new sequence with valid stage transitions?
Deformable tasks	Rigid/articulated object manipulation	Can the policy adapt to compression, folding, and packing?

Skill transfer and composition. These tasks reuse skills in new sequences or contexts. *Open Laptop and Click Mouse* combines opening an articulated object with interface operation; *Scrub Bottle Then Fill Cup* connects tool interaction to a subsequent goal. Evaluation records where an execution stops and whether a successful stage leaves the state needed by the next stage. A previously unseen atomic operation is identified separately from a new composition of trained skills.

Transfer to deformable manipulation. *Compress Sponge into Mug*, *Fold Towel Twice*, and *Pack Pillow* require handling deformation without deformable-task demonstrations in the benchmark training set. These are transfer challenges across object, goal, and physical behavior. Isolating compliance as a causal factor would additionally require matched geometry and goals; the present suite does not assume that all differences are due to material properties.

B.10 Visual and Language Robustness Protocols

Training randomization and evaluation under domain shift serve different roles. Fresh samples from the training distribution measure in-distribution performance; held-out combinations or expanded ranges of visual factors measure robustness under specified shifts. These tests retain the task goal and identify the factors changed. They are reported separately from the evaluation-only tasks, which change asset relations, skills, or physical interactions. Combining both changes is an additional challenge condition, not a replacement for the matched-distribution baseline.

The eight held-out paraphrases per active goal define a language-only test: the environment, target values, reset seeds, and success predicate stay fixed while the instruction changes. Paired trials compare canonical, training-paraphrase, and held-out-paraphrase success. We report the success gap and paired outcome agreement, so consistently failing all wordings cannot masquerade as successful language robustness. This test is separate from Transfer-16 physical tasks and from visual domain shifts.

A complementary goal-sensitivity test changes the requested object, relation, or numerical setting only within a task's supported configurations, with its executable goal updated accordingly. For example, the timer must match the requested minutes and seconds; moving a knob must reach the requested angle and direction. Such tests assess whether the policy uses the instruction, rather than repeating a familiar motion. Unseen numerical values require an explicitly held-out goal split. Text-invariant policies are not scored for language robustness; their structured-goal performance is reported separately. These additional tests require dedicated trials and are not inferred from the canonical-prompt evaluations.

B.11 Matched Physical-Transfer Protocol

A paired task comprises simulation and real instances with a correspondence record:

$$\mathcal{P}_i = (\mathcal{T}_i^{\mathrm{sim}}, \mathcal{T}_i^{\mathrm{real}}, \mathcal{C}_i). \tag{3}$$

The record specifies robot and camera configuration, state/action frames, control timing, initialization conditions, and success definitions. Direct transfer deploys the same checkpoint without target-task real demonstration training or real-data finetuning. Calibration measurements and any real images used to configure the system are disclosed separately from policy training. External pretraining of a VLA or WAM retains its own data identity.

Only tasks with physical counterparts enter the paired analysis, which reports their coverage and per-task simulation and real trial counts and computes the absolute gap for model m and task i as

$$\Delta_{i,m} = S^{\text{sim}}_{i,m} - S^{\text{real}}_{i,m}. \tag{4}$$

Same-task real transfer and Transfer-16 generalization are independent claims. Real-data adaptation, when studied, belongs to a separate protocol with an explicit data budget (Appendix F).

B.12 Additional Task-Capability Examples

Articulated collection assets occur in 21 tasks: 12 involve revolute joints and 11 prismatic joints, with two containing both. Goals specify motion direction, travel, angle, or unlocking prerequisites. Set Timer uses requested minutes and seconds; Set Dual Knobs uses left/right angle and direction targets; and Unfold Fan targets 60, 90, or 120 degrees. Fixed three-cycle scissors and propeller requirements do not by themselves establish unseen-count generalization. Place Tableware combines identity, position, and orientation; Place Largest Fitting Ball and Sort Nested Cups add size comparisons. Prepare Coffee and Load and Start Microwave require intermediate states that enable later operations. These capability labels overlap the catalog's 17 primitive, 18 functional, and 15 contact-rich tasks and its 31 single-stage, seven repetitive, and 12 long-horizon tasks.

B.13 Related-Work Distinctions and Counting Conventions

Dexterous benchmarks and demonstration data. DexJoCo, DexVerse, Bench2Dex, and LabDex study functional manipulation, multiple embodiments, bimanual interaction, and hierarchical procedures (Wang et al., 2026a; Yao et al., 2026; Yang et al., 2026; Tang et al., 2026). Their evaluated policies include ACT and DP as well as OpenVLA, $\pi_{0.5}$, and GR00T N1.5, depending on the benchmark. DexMimicGen scales a small human demonstration set through automated trajectory generation (Jiang et al., 2025); SPD collects long-horizon simulation interaction for subsequent real-world adaptation (Kamat et al., 2026). DexOmni emphasizes a large curated set of human demonstrations combined with task holdouts, re-rendering, and comparison across contemporary policy families. Table 1 separates human and generated trajectories, since their counts measure different collection processes and do not imply equal interaction duration.

Generalization and policy evaluation. LIBERO organizes knowledge-transfer problems, CALVIN studies language-conditioned long-horizon execution, and DexArt evaluates transfer to unseen articulated objects (Liu et al., 2023; Mees et al., 2022; Bao et al., 2023). RoboDojo organizes a broad policy comparison around complementary capability dimensions, while RoboTwin 2.0 combines scalable data generation with domain randomization (Chen et al., 2026; 2025). DexOmni instantiates separate measurements of execution, robustness, and transfer for multi-finger functional manipulation, including evaluation-only combinations and deformable interactions. Its WAM coverage complements prior dexterous benchmarks; a world-model objective and a diffusion action head do not by themselves define mutually exclusive policy classes.

Visual augmentation and dexterous sim-to-real transfer. Trajectory replay already supports visual randomization in DexJoCo (Wang et al., 2026a). Our focus is its integration with a larger curated dataset, high-fidelity rendering, and controlled evaluation of diversity and transfer. Dexterous sim-to-real learning has a long history: Dactyl and DeXtreme transfer simulation-trained in-hand policies to physical hands (OpenAI et al., 2018; Handa et al., 2023). SPD uses physical demonstrations after simulation pretraining (Kamat et al., 2026), while sim-and-real co-training studies combining the two data sources (Maddukuri et al., 2025). DexOmni's direct-transfer protocol is defined to evaluate simulation demonstration learning without target-task real training data. It also differs from RoboDojo's complementary simulation and real suites, which are not constructed as one-to-one matched task pairs (Chen et al., 2026).

Table 1 reports trajectories, not matched interaction hours or equal task exposure. DexOmni's count includes its trajectory-validation partition; simultaneous views and re-rendering do not add human demonstrations. A dash indicates that no generated-trajectory count is used in that cell; NR means that the compared feature or experiment is not reported, rather than proof that the software cannot support it. The 5.7–16.4-fold range compares DexOmni with DexVerse and DexJoCo, respectively.

How to read the comparison. Table 1 compares the experiments and mechanisms described in each paper, not a ranking of visual fidelity. DexJoCo uses MuJoCo and replay-based visual augmentation (Sec. III-D); its language test changes digits into arithmetic expressions and words (Sec. IV), rather than using a train/test paraphrase bank. DexVerse uses Isaac Lab with six hand embodiments, configurable appearance and dynamics, and recorded-state replay (Secs. 3–4); its 100-task catalog is larger than DexOmni's, while its 3,180 demonstrations cover 56 single-goal and five long-horizon tasks. Bench2Dex uses Isaac Lab with 12 hand embodiments and offline multimodal replay (Secs. 3.1–3.4), including simulated tactile images. Its perturbation tests retain the goal, object set, and success criteria (Sec. 3.6), and its stage, safety, timing, and grasp diagnostics are already extensive (App. E). These capabilities are complementary to DexOmni's fixed embodiment and separate Transfer-16 task suite; neither multi-embodiment coverage nor fine-grained scoring is claimed as unique to DexOmni.

DexMimicGen's 21,000 generated trajectories and 60 human seeds are separate counts across dexterous and gripper settings. Its broader-state experiments train and test on the expanded distributions; they are not a test-only task split. Its physical experiment executes generated trajectories on hardware to obtain real observations for policy training (Sec. VI-D), which differs from deploying a policy trained only on simulated observations. SPD reports approximately 75 simulation hours, exceeding DexOmni's roughly 56 hours despite its smaller episode count. It renders recorded trajectories offline and applies mask-based visual augmentation; although collection prompts vary, the trained policy does not take language as input (Secs. 3.1–3.3). Its physical results follow real-data finetuning (Sec. 4). Thus, a larger human trajectory count is not an across-the-board claim of more interaction time, more task types, or higher physical accuracy.

Table 10: Broader benchmark context. Task-transfer holdouts, visual/language variation, and sim/real evaluation have precedents beyond dexterous benchmarks.

Resource	Relevant coverage	Relation to DexOmni
RoboDojo (Chen et al., 2026)	42 sim / 18 real tasks; 8 Open tasks excluded from training; 30 integrated policies; success and partial-progress scores.	Evaluation-only skill recombination is a precedent. Sim and real suites are complementary, not one-to-one direct-transfer pairs.
RoboTwin 2.0 (Chen et al., 2025)	50 bimanual gripper tasks; over 100,000 synthesized trajectories; clutter, lighting, texture, table-height and language randomization.	Already evaluates simulation-only real transfer and mixed sim/real training. Generated gripper trajectories differ from human dexterous demonstrations.
LabDex (Tang et al., 2026)	Atomic, compositional and long-horizon laboratory tasks; ACT, DP and $\pi_{0.5}$; real and simulation evaluation.	Hierarchical task-progress analysis is a precedent. Its compositional tasks have training demonstrations rather than constituting an exclusively test-only suite.

DexOmni therefore emphasizes the joint setting: a large human dexterous demonstration corpus, controlled visual and prompt diversity, DP/VLA/WAM comparison, and task holdouts covering asset relations, skill composition, and deformable interactions. The WAM claim is specific to this dexterous comparison, not to all robot benchmarks. Transfer-16 results and direct physical-transfer results must be reported with their own validated coverage; the existence of these protocols does not establish successful transfer.

C Core-50 Results and Coverage

Table 2 is computed from 45,000 episodes on all 50 Core-50 tasks. The eight multi-task checkpoints contribute 100 episodes per task (39,999 valid; HY-Embodied has 99 valid episodes on Place Wine Glass), and single-task DP contributes 5,000 episodes from task-specific checkpoints. Overall success is the unweighted mean of the 50 task success rates; interaction-type and hand-skill means use the same rule within each group. Chinese Chess remains outside the catalog.

Scope and validity. All multi-task evaluations passed the recorded runtime validation and match the selected checkpoints; the DP checks are described below. The evaluator samples each reset from the admitted source templates of its task, so the scores describe this implemented reset distribution. Native reset rejection is handled before policy scoring by selecting the first accepted candidate while retaining the assigned goal and language; rejected resets and infrastructure interruptions are not policy failures. On 41 tasks, the eight multi-task checkpoints share the reset seed, variant, goal, appearance, and prompt at nearly every episode index; on the nine goal-conditioned tasks, the goal and prompt are sampled per checkpoint.

Replacement and scoring records. Exact original-slot associations include the recovered minority-goal Fit Device Cover episodes and the recovered DM0.5 Place Wine Glass episode; no slot is counted twice. Each checkpoint contributes 100 Lift Carton episodes evaluated with the lift/hold criteria in Appendix B.

Single-task DP reference. DP uses a distinct checkpoint at 30,000 updates for each task. Ten tasks use completed retraining with corrected near-degenerate normalization; the other tasks retain their previously selected task checkpoints, and Lift Carton uses the same lift/hold criteria. Original and retrained model identities are preserved per task. The retained evaluations use the earlier native goal sampler, whereas the newer runs use frozen training-frame-weighted goal quotas, so DP is a descriptive reference rather than a goal-distribution-matched comparison. Three rejected knife-task resets are omitted under the original first-100-accepted-seeds rule. The DP audit checks checkpoint hashes, native completion, native success, and recorded three-view video completion.

The accompanying machine-readable record stores the task map, checkpoint identities, source plans, supplemental-slot associations, and the hash of the episode-level snapshot.

Uncertainty. Table 11 reports two 95% percentile bootstrap intervals. The episode-only interval (20,000 resamples) redraws each task's outcomes from its observed success rate with the 50 tasks fixed; it quantifies finite rollout sampling under an independent-episode assumption. The task-and-episode interval (2,000 resamples) also resamples the 50 tasks with replacement, so it describes a new draw of tasks from the same catalog. Neither interval includes training-run or checkpoint-selection uncertainty.

Table 11: Core-50 task-macro success on all 50 tasks with 95% bootstrap intervals over episodes only and over tasks and episodes. Shaded: best.

Model	Successes / episodes	SR (%)	Episodes CI (%)	Tasks+episodes CI (%)
OpenWAM	1,453 / 5,000	**29.1**	[28.0, 30.1]	[22.8, 36.0]
FastWAM	1,319 / 5,000	26.4	[25.4, 27.4]	[20.0, 33.8]
$\pi_{0.5}$	1,243 / 5,000	24.9	[23.8, 25.9]	[18.6, 31.7]
DM0.5	1,072 / 5,000	21.4	[20.5, 22.4]	[15.2, 28.1]
LingBot-VLA	984 / 5,000	19.7	[18.8, 20.6]	[13.6, 26.2]
HY-Embodied	947 / 4,999	18.9	[18.0, 19.8]	[13.2, 25.4]
Motus	903 / 5,000	18.1	[17.2, 18.9]	[12.3, 25.0]
Spatial Forcing	620 / 5,000	12.4	[11.6, 13.2]	[7.9, 18.0]
DP (single-task)	483 / 5,000	9.7	[9.0, 10.3]	[5.0, 14.9]

Table 12: Per-task success rates (%) on the 50 Core-50 tasks. Each cell has 100 episodes, so it also gives the number of successes; *HY-Embodied has 99 valid episodes on Place Wine Glass. Columns use the selected checkpoints from Section 5.1; OW, FW, LB, HY, and SF denote OpenWAM, FastWAM, LingBot-VLA, HY-Embodied, and Spatial Forcing. The DP sampler caveat above applies to every DP cell. Shaded: best on each task, ties included.

ID	Task	OW	FW	$\pi_{0.5}$	DM0.5	LB	HY	Motus	SF	DP
01	Align Blocks in Midair	2	3	0	0	1	0	0	1	0
02	Connect Charger and Cable	1	0	0	0	0	0	0	0	0
03	Cover Box	61	0	34	34	57	38	32	0	0
04	Cover Pitcher	12	3	5	5	5	1	10	0	2
05	Fill Cup	4	12	6	0	2	0	0	0	0
06	Fit Device Cover	4	3	0	0	2	1	1	1	0
07	Flip Sponge	1	0	0	0	0	0	1	0	0
08	Hammer Nail	5	0	0	0	1	0	0	1	0
09	Insert Peg	14	3	2	4	2	1	4	0	0
10	Lift Carton	14	16	55	53	65	73	8	37	51
11	Line Up Blocks	0	0	0	0	0	0	0	0	0
12	Load and Start Microwave	47	46	36	4	23	22	8	23	0
13	Load Cans	23	43	17	19	14	13	10	17	0
14	Move and Click Mouse	44	23	16	38	33	28	13	1	29
15	Open Drawers	32	36	59	14	62	55	4	19	29
16	Open Laptop	63	61	42	64	54	54	49	20	23
17	Open Latched Door	69	68	63	54	76	73	75	64	53
18	Open Micropipette Box	36	53	60	43	46	51	8	37	46
19	Orient Cup Upright	35	26	38	34	36	26	21	8	38
20	Pass and Orient Pen	19	8	12	8	9	5	6	3	0
21	Pass Block	33	23	14	13	9	13	10	19	2
22	Pass Screwdriver	51	79	58	59	52	58	31	64	23
23	Photograph Logo	35	27	17	10	19	11	34	10	4
24	Place Ball in Bowl	68	76	75	72	22	20	76	16	6
25	Place Cup on Plate	50	50	62	69	61	55	72	18	25
26	Place Fruit in Bucket	49	38	57	65	13	13	60	10	6
27	Place Largest Fitting Ball	22	33	35	41	0	3	18	5	0
28	Place Tableware	18	20	19	25	1	10	13	4	0
29	Place Wine Glass*	42	39	22	14	15	15	42	0	9
30	Pour Seasonings	60	34	34	3	37	34	12	5	1
31	Prepare Coffee	16	27	8	2	1	9	11	5	0
32	Press Soap Pump	33	26	9	11	7	0	1	12	0
33	Reposition Marker Grasp	21	22	32	17	15	10	9	3	5
34	Roll Ball Through Gate	26	17	21	30	17	21	15	21	8
35	Rotate Propeller	95	91	82	81	75	76	78	75	79
36	Scoop Marbles	0	0	0	0	0	0	0	0	0
37	Scrub Bottle	18	9	2	9	16	7	0	4	22
38	Set Dual Knobs	0	1	1	0	1	1	0	1	0
39	Set Timer	0	0	0	0	0	0	0	0	0
40	Slide Knife Blade	6	14	15	9	2	6	3	6	0
41	Sort Nested Cups	7	6	2	2	0	2	0	0	0
42	Sort Objects	70	56	43	21	11	9	33	10	0
43	Squeeze Scissors	41	34	26	21	12	17	19	20	2
44	Stack Blocks	10	2	1	0	0	0	3	0	0
45	Stir Pot	21	21	4	3	6	1	3	4	0
46	Store Charger	66	74	63	58	45	45	47	36	9
47	Trigger Drill on Block	14	6	18	10	28	19	4	5	2
48	Unfold Fan	0	3	5	3	2	1	0	0	0
49	Unfold Ruler	57	48	49	26	17	25	33	33	2
50	Water Plant	38	39	24	24	12	25	26	2	7

Continued on next page

Table 12: Per-task success rates (continued).

ID	Task	OW	FW	$\pi_{0.5}$	DM0.5	LB	HY	Motus	SF	DP
	Task-macro mean	29.1	26.4	24.9	21.4	19.7	18.9	18.1	12.4	9.7

D CORE-50 ANALYSIS DETAILS

Hand-skill groups. Table 13 lists the grouping of Core-50 tasks by the hand skill that decides success, used in Figure 3a and Section 5.2. Each task is assigned to exactly one group by its success condition; tasks that also require reaching and grasping are assigned to the later skill. The grouping is an analysis lens and is independent of the interaction (P/F/CR) and structure (SS/LH/RP) labels of the catalog. Figure 6 shows the success of every checkpoint on every task in this grouping.

Table 13: Hand-skill groups of Core-50 tasks. Pooled: successes over all episodes of the eight multi-task checkpoints. Best: mean over tasks of the best checkpoint. Blue: the groups that contain every task above 40% mean success; orange: essentially unsolved groups, where no checkpoint exceeds 21% on any task.

Group (tasks)	Tasks	Pooled %	Best %
Nonprehensile (2)	Rotate Propeller, Roll Ball Through Gate	51.3	62.5
Articulated part (6)	Open Latched Door, Open Laptop, Open Micropipette Box, Open Drawers, Unfold Ruler, Load and Start Microwave	42.9	61.0
Grasp–transport–release (13)	Place Cup on Plate, Place Ball in Bowl, Place Fruit in Bucket, Place Largest Fitting Ball, Place Tableware, Place Wine Glass, Sort Objects, Load Cans, Pass Block, Pass Screwdriver, Store Charger, Cover Box, Lift Carton	34.9	58.0
Finger actuation (9)	Squeeze Scissors, Water Plant, Trigger Drill on Block, Move and Click Mouse, Press Soap Pump, Set Timer, Set Dual Knobs, Photograph Logo, Slide Knife Blade	14.0	26.2
Pour or fill (3)	Pour Seasonings, Fill Cup, Prepare Coffee	13.4	33.0
In-hand reorientation (5)	Orient Cup Upright, Reposition Marker Grasp, Pass and Orient Pen, Sort Nested Cups, Unfold Fan	11.4	20.2
Tool on object (6)	Hammer Nail, Scoop Marbles, Line Up Blocks, Flip Sponge, Scrub Bottle, Stir Pot	2.9	7.5
Precision fit (6)	Insert Peg, Fit Device Cover, Align Blocks in Midair, Connect Charger and Cable, Stack Blocks, Cover Pitcher	2.2	7.3

Demonstration count and duration. The groups differ in how many training trajectories their tasks have and how long the expert takes (Table 14), so group differences could reflect data volume or episode length rather than the skill. Across the 50 tasks, mean success over the eight checkpoints has Spearman $\rho = 0.27$ with the number of training trajectories and $\rho = -0.71$ with the expert P75 duration. Regressing task success on the group indicators alone explains $R^2 = 0.62$ of its variance; the two covariates alone explain 0.37. With both covariates added ($R^2 = 0.73$), duration keeps a clear effect and demonstration count none, and the finger-actuation, in-hand, tool, and precision groups remain 20–25 points below grasp–transport–release. The in-hand and tool groups have few demonstrations, but precision fits have as many as the best-solved group. By quota, the seven 100-trajectory tasks average 6.1%, the 24 300-trajectory tasks 23.4%, and the 19 500-trajectory tasks 24.4%; within the 500 quota, the four precision fits average 2.8% and the eight grasp–transport–release tasks 37.8%.

Language probe. To test whether a checkpoint reads the object name, we query it offline on 500 demonstration episodes of Place Ball in Bowl (training and validation trajectories) at frames 0 and 15, 1,000 inputs in total. For each input we sample twice with the true instruction and once with the instruction naming the other ball, and compare the palm position commanded 30 steps (1 s) ahead.

Table 14: Task success (mean of the eight multi-task checkpoints, %) regressed on hand-skill group over the 50 Core-50 tasks, with grasp–transport–release (GTR) as the reference; the GTR row gives the intercept of the raw model. Covariates are natural logs of the task's training trajectories and expert P75 duration. Brackets: 95% intervals from 4,000 task bootstrap draws stratified by group. Demos and P75: group medians over tasks.

Group	Demos	P75 (s)	Raw	With covariates
GTR (reference)	392	8.2	34.9 [26.9, 43.1]	—
Nonprehensile	362	10.5	+16.4 [−19.3, 52.2]	+18.7 [−16.1, 52.9]
Articulated part	259	6.7	+8.0 [−4.6, 21.9]	+7.4 [−3.0, 19.8]
Finger actuation	272	12.9	−20.9 [−31.2, −10.8]	−19.9 [−27.8, −12.2]
Pour or fill	208	17.7	−21.5 [−34.6, −6.7]	−13.8 [−24.8, −3.3]
In-hand reorientation	89	12.3	−23.5 [−35.2, −11.2]	−22.1 [−31.6, −11.7]
Tool on object	159	23.5	−32.0 [−40.7, −23.5]	−21.3 [−29.9, −11.8]
Precision fit	413	14.7	−32.7 [−41.0, −24.4]	−24.6 [−33.1, −16.6]
log training trajectories			—	−0.8 [−5.1, 5.4]
log P75 duration			—	−11.9 [−16.2, −7.6]
R^2			0.62	0.73

The probe uses checkpoints of the evaluated training runs, within 6,000 updates of the evaluated ones. Swapping the name moves the endpoint by 14.8, 17.0, 7.3, and 8.5 cm, against 7.0, 8.2, 8.8, and 8.1 cm between the two true-instruction samples. The true-instruction endpoint is closer to the demonstration than the swapped one in 92, 88, 61, and 71% of inputs. For a target-level test we use the $\pi_{0.5}$ true- and swapped-instruction endpoints as references for the two balls and keep the 314 first frames where they are at least 8 cm apart: DM0.5 heads for the named ball in 93% of them and, with the swapped instruction, still for the originally named ball in 10%; HY-Embodied does so in 58 and 57%, and LingBot-VLA in 74 and 71% (an earlier LingBot-VLA checkpoint: 55 and 55%). The frame-0 layout does not reveal the target (a logistic classifier on it reaches 53% cross-validated accuracy against a 52% majority), so HY-Embodied chooses a ball nearly at random, and LingBot-VLA, which keeps choosing the true ball under the swapped name, recalls it from the largely seen training scenes. The serving and offline paths produce bitwise-identical actions, so this is a learned shortcut, not an interface fault.

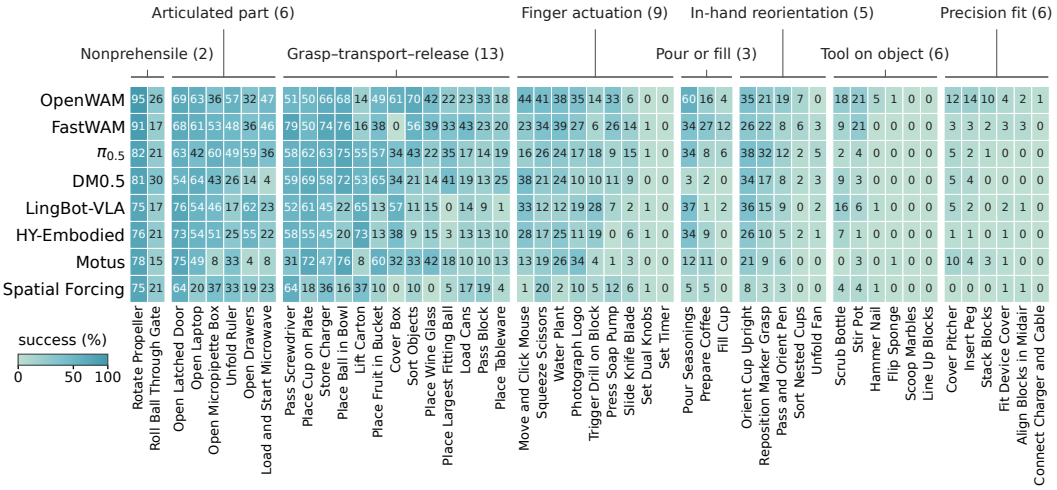


Figure 6: Core-50 success (%) of the eight multi-task checkpoints on each task, grouped by hand skill as in Table 13 and sorted within a group by mean success. Each cell has 100 episodes (99 for HY-Embodied on Place Wine Glass); DP and task IDs are in Table 12.

Episode outcomes. Figure 7 splits the Core-50 episodes of each checkpoint by how they end (averages in Figure 4b and Section 5.5). Early endings are native terminations before the time limit,

either forbidden or wrong contact (for example, the hand touching an object that must be moved with a tool) or another rule, such as a premature or out-of-order event or a drop. A time-out is stalled when its checker records no new state after the expert 75th-percentile duration and still progressing otherwise; it is not instrumented when the task checker records fewer than two progress states.

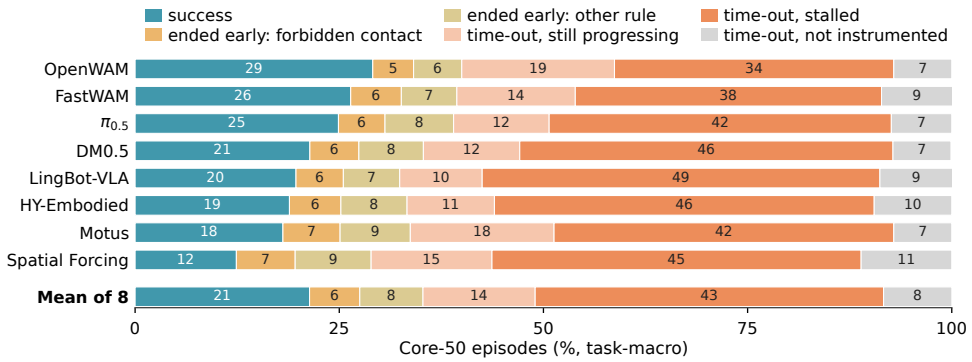


Figure 7: How Core-50 episodes end, per checkpoint (task-macro share of episodes over the 50 tasks).

Success within a time limit. Figure 8 counts, for each checkpoint, the successes completed within a given multiple of the expert 75th-percentile duration on the same trajectories (all 50 tasks). Shorter limits are prefixes of the evaluated $3\times$ rollouts, not reruns, so the curves do not estimate what a longer limit would add. Success at $1\times/1.5\times/2\times/3\times$ is OpenWAM 8.3/19.1/24.0/29.1; FastWAM 10.2/19.1/22.8/26.4; $\pi_{0.5}$ 14.6/19.6/22.2/24.9; DM0.5 10.9/16.9/19.3/21.4; LingBot-VLA 12.0/16.1/17.8/19.7; HY-Embodied 11.0/15.1/16.8/18.9; Motus 5.4/11.5/14.6/18.1; Spatial Forcing 5.6/8.7/10.2/12.4. The ranking at $1\times$ is $\pi_{0.5}$, LingBot-VLA, HY-Embodied, DM0.5, FastWAM, OpenWAM, Spatial Forcing, Motus; at $3\times$ it is the order of Table 2, and 12 of the 28 checkpoint pairs change order. Median completion times, over successful episodes and in units of the expert P75 duration, are OpenWAM 1.22, FastWAM 1.13, Motus 1.27, $\pi_{0.5}$ 0.90, DM0.5 0.99, LingBot-VLA 0.88, HY-Embodied 0.91, and Spatial Forcing 1.09.

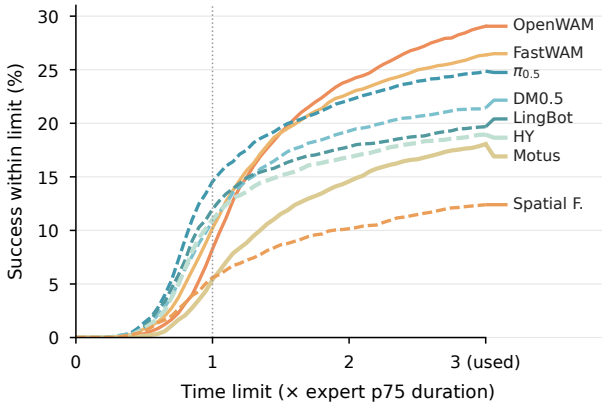


Figure 8: Task-macro success counting only episodes completed within a given multiple of the expert 75th-percentile duration. Solid: world-action models; dashed: VLA checkpoints.

Reliability details. Per-task signal-to-noise, the spread of success across checkpoints divided by its binomial standard error, is below 1 on 6 tasks and at least 3 on 14. Twelve tasks selected by this ratio on even-indexed episodes reproduce 96% of the 28 pairwise orders measured on odd-indexed episodes (88% for random twelve-task subsets), and 24 tasks reproduce all of them. Episode-only

95% half-widths of the 50-task macro score are about 1.9, 1.3, and 0.9 points for 25, 50, and 100 episodes per task. Selecting a checkpoint per task on half of the episodes and scoring it on the other half gives 32.9% versus 29.1% for the best single checkpoint. An earlier evaluation campaign of the same $\pi_{0.5}$ and OpenWAM checkpoints used the fixed canonical prompt and a single goal per task; on the 48 shared tasks it gives 24.8 versus 23.4% ($\pi_{0.5}$) and 30.5 versus 28.5% (OpenWAM), with per-task Spearman ρ of 0.98 and 0.97 and per-task differences whose standard deviation (4.3 and 6.3 points) is close to the binomial expectation (5.1 and 5.5). Within the Core-50 snapshot, success is not measurably affected by which of the four table materials, three colour palettes, or two roughness levels is drawn (largest within-task effect 0.8 points, permutation $p \geq 0.17$), and training paraphrases score within 2.2 points of the canonical sentence for seven checkpoints (OpenWAM: canonical $+4.0$ points, $z = 2.9$). All appearance factors and paraphrases are inside the training distribution, so these are invariance checks, not generalization results.

E Single-Task Demonstration Scaling

Training controls. The two tasks use independent $\pi_{0.5}$ finetuning runs initialized from the same generic pretrained base with a freshly initialized 52-dimensional action interface. All runs use full finetuning, batch size 32, training seed 0, and exactly 20,000 optimizer updates. The learning rate warms up for 1,333 updates to 5×10^{-5} and decays to 5×10^{-6}. The 10/50/100/200-trajectory subsets are a single frozen nested draw; A03 sampling is stratified approximately in proportion to the three ball types. The full tier contains every task-local training trajectory: 438 for A03 (192 tennis, 175 mini soccer, 71 golf) and 439 for D05. The 62 A03 and 61 D05 held-out trajectories are excluded at every size. Images, augmentation, state/action semantics, the V16 prompt revision, and the 20/80 canonical/paraphrase training sampler are unchanged across sizes.

Task-wide normalization statistics are fixed across sizes and computed from the full training pool, excluding validation. Thus small-data runs have access to larger-pool summary statistics; this is not a strictly subset-only data-access experiment. Fixed update counts also imply different reuse rates across dataset sizes. The selected checkpoint is always update 20,000, not the checkpoint with the highest test success. Full-inference offline validation runs every 1,000 updates on fixed anchors; A03's held-out trajectories contain golf balls only, so this validation does not represent the complete ball distribution. The curves here report fresh closed-loop success, not offline reconstruction error.

Closed-loop evaluation. Each checkpoint has 100 accepted, runtime-valid episodes, giving 1,000 episodes in total with no missing slots. Sizes share the same planned task-specific reset seeds, goals, and prompts. Native reset rejection may advance an accepted seed; shared planned draws do not certify identical complete physical states. Evaluation keeps the native EEF controller, 30 Hz action execution, the accepted rendering revision, and checkpoint normalization. The physical budgets are 3×P75: 321 actions (10.7 s) for A03 and 663 (22.1 s) for D05. Inference uses a 30-action horizon and ten denoising steps. The prompt bank and goal distribution are fixed across training sizes rather than refitted to each subset. A03 evaluates 42 tennis, 40 mini-soccer, and 18 golf goals; D05 requires lifting the scissors and completing three cuts in the air.

Table 15: Single-task scaling results. Success counts are out of 100 trials per cell; intervals are pointwise 95% Wilson intervals in percent. Full-pool sizes are A03 / D05. Shaded: best.

Trajectories	A03 s/n	95% CI	D05 s/n	95% CI
10	7/100	[3.4, 13.7]	1/100	[0.2, 5.4]
50	23/100	[15.8, 32.2]	11/100	[6.3, 18.6]
100	31/100	[22.8, 40.6]	13/100	[7.8, 21.0]
200	65/100	[55.3, 73.6]	18/100	[11.7, 26.7]
Full (438 / 439)	**83/100**	[74.5, 89.1]	**44/100**	[34.7, 53.8]

Figure 4 and Table 15 use raw success fractions under these fixed evaluation draws. Reweighting A03's per-goal success by the frozen training-frame prior instead gives 6.99%, 22.89%, 30.88%, 64.67%, 82.80%, in increasing data-size order; D05 has one goal and is unchanged. The intervals describe approximate rollout-level binomial uncertainty conditional on each checkpoint; they do

not estimate training-seed or subset-selection variance and are not simultaneous confidence bands. There is only one training seed and one subset draw at each size.

Result scope and provenance. The runtime checks pass. As for Core-50 (Appendix C), the evaluator reuses admitted source templates, so these curves describe the implemented closed-loop reset distribution rather than every training-source cohort. Machine-readable counts, result hashes, checkpoint steps, goal strata, and evidence-manifest hashes accompany the manuscript in `data/scaling_results_20260926.json`.

F Supplementary Real-Data Adaptation Case

This Place Ball in Bowl case concerns real-data adaptation, separate from the direct-transfer protocol in Appendix B.11.

An earlier simulation-only policy was observed to approach the target in the tabletop plane but failed to descend to a grasp-feasible height after camera and controller alignment. A policy trained on 200 real demonstrations showed appropriate grasping height but unreliable planar alignment. Manually placing the ball into its hand enabled subsequent placement; this assisted continuation is not autonomous success.

Reported qualitative improvement from joint simulation and real-data training motivates a separate budget-controlled comparison. It establishes neither a numerical gain nor a perception/control decomposition. Related work studies simulation pretraining and sim-and-real co-training (Kamat et al., 2026; Maddukuri et al., 2025).

G TASK PROFILES

Table 16 follows the task order in Table 4. Each row presents the task identifier and name, a screenshot, and the task instruction; the interaction, task-structure, and asset labels of each task are listed in Table 4. Tasks 01–50 are used for both training and evaluation; the 16 tasks with EVAL-prefixed identifiers form Transfer-16. Source superscripts match Table 4; Appendix A.1 explains task adaptations and asset reuse. For training tasks, we show one canonical prompt from the training set; for Transfer-16 tasks, we show the configured task prompt. When a task has multiple target-specific configurations, one instance is shown.

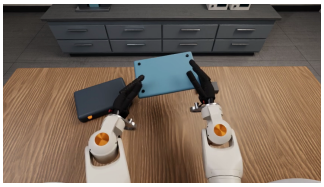
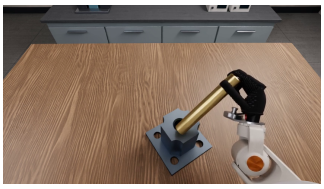
Within the Task Instruction column, **bold text** marks the randomly initialized task parameters, such as target objects, spatial relations, orientations, angles, durations, and action choices. Concrete values show one example configuration. Visual randomization that does not alter the instruction is not marked. Meaning-preserving prompt variants are a separate language dimension (Section 3.2); the instructions here do not enumerate the 32 training and eight held-out paraphrases associated with each goal-specific canonical instruction.

Table 16: Task profiles of DexOmni.

Task	Screenshot	Task Instruction
01 Align Blocks in Midair		Lift both assigned blocks with their assigned hands, then mate their marked faces in midair and hold the physical contact stable for 0.4 seconds.
02 Connect Charger and Cable		Insert the charger plug into the power strip first, then insert the charging cable connector into the charger head.
03 Cover Box		Align the box lid and place it onto the clear box.
04 Cover Pitcher		Align the pitcher lid and place it onto the pitcher body.
05 Fill Cup		Hold the mug upright under the faucet with the left hand. Open the right lever outward past 70 percent, fill for half a second, place the mug on the marked pad, and close the faucet.

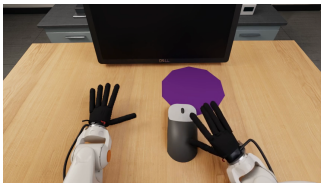
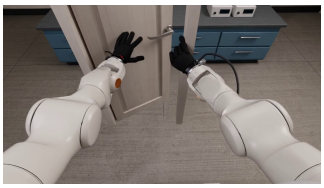
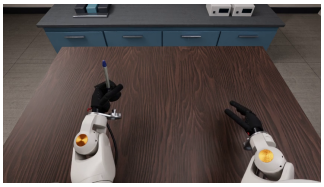
Continued on next page

Table 16: Task profiles of DexOmni (continued).

Task	Screenshot	Task Instruction
06 Fit Device Cover		Use both hands on opposite sides of the device cover to carry it together over the device body, then align it with its matching interface and seat it.
07 Flip Sponge		Hold the skillet level with the left hand. Use the right-hand spatula to lift and flip the sponge once, leaving its opposite face stable inside the skillet. Do not touch the sponge directly.
08 Hammer Nail[J]		Hold the wood block with the left hand, keeping its nail inserted. Pick up the hammer with the right hand and strike the nail so it sinks into the block.
09 Insert Peg[JA]		Insert the round peg into the socket tray hole.
10 Lift Carton[V]		Support opposite lower carton sides with both hands, lift the entire carton clear of the table, and keep it continuously airborne for 3 seconds.
11 Line Up Blocks		Use only the set square tool to push three scattered wood blocks onto the baseline and align them side by side.
12 Load and Start Microwave[J]		Open the microwave door, carry the plate with its food box into the microwave, close the door, then press the green start button.
13 Load Cans[VA]		Pick up both soup cans with their assigned hands and cross-load them into the opposite bin partitions. Release them within the synchronization window and leave them stable.

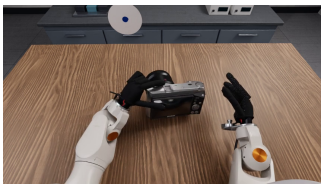
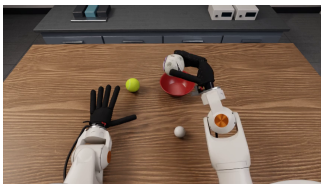
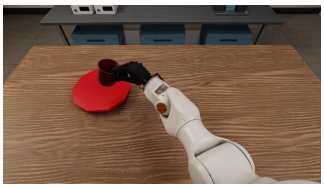
Continued on next page

Table 16: Task profiles of DexOmni (continued).

Task	Screenshot	Task Instruction
14 Move and Click Mouse[J]		Move the mouse onto the mouse pad without pressing either button, then click the left mouse button.
15 Open Drawers		Pull each **highlighted drawer** with its assigned hand past 60 percent travel, crossing both thresholds within 1.2 seconds.
16 Open Laptop[V]		Lift the laptop display from its front edge with the right hand and open it to 60 degrees.
17 Open Latched Door		Depress the near lever past 45 degrees with the left hand for five frames. Push the unlatched door to 45 degrees with the right hand, avoiding the table, frame, and far handle.
18 Open Micropipette Box		Open the hinged lid of the micropipette box until it is fully open.
19 Orient Cup Upright		Pick up the cup and place it upright on the table.
20 Pass and Orient Pen		Pass the pen from right-hand-only contact through both-hand contact to left-hand-only contact without dropping it. Hold it tip-up within 2 degrees of vertical.
21 Pass Block		Pick up the left block with the left hand, transfer it to the right hand in the middle, then place it stably inside the right box.

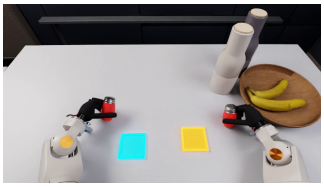
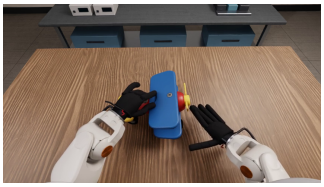
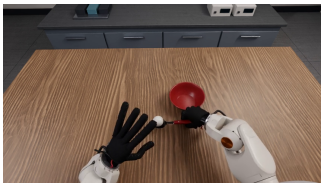
Continued on next page

Table 16: Task profiles of DexOmni (continued).

Task	Screenshot	Task Instruction
22 Pass Screwdriver		Touch the screwdriver with the right hand first, then touch it with the left hand while it is airborne, and let it land in the horizontal green target zone.
23 Photograph Logo[J]		Pick up the camera, aim it at the logo, and press the shutter button.
24 Place Ball in Bowl		Pick the **tennis ball** and place it into the bowl.
25 Place Cup on Plate		Pick up the mug and place it upright in the center of the plate.
26 Place Fruit in Bucket		Pick the **orange** and place it into the bucket.
27 Place Largest Fitting Ball		Choose the largest ball that fits the **large yellow J cup**. Place it in that cup, release both hands, and leave ball and cup stable.
28 Place Tableware		Pick up the **mug** and place it **above and to the right of the plate**, **upright with its opening facing up**. Move only the **mug** and leave the plate and distractors in place.
29 Place Wine Glass		Place the wine glass upright at the plate center, release it, and leave it stable.

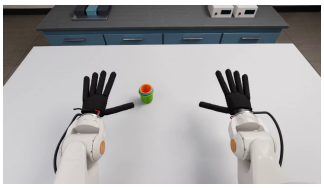
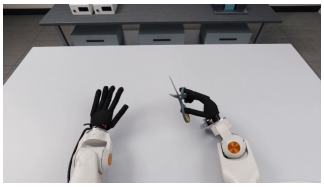
Continued on next page

Table 16: Task profiles of DexOmni (continued).

Task	Screenshot	Task Instruction
30 Pour Seasonings		Use the assigned hands to pour salt into the cyan target and pepper into the orange target, with overlapping pouring. Keep each jar's total pouring time below four seconds.
31 Prepare Coffee		Take the mug from the cabinet and close both doors. Place it under the coffee outlet, press the green button, and brew for two seconds. Release the filled mug upright on the serving tray.
32 Press Soap Pump		Stabilize the sponge with the left hand, align the soap nozzle above it with the right hand, and physically depress the dispenser.
33 Reposition Marker Grasp		Lift the capped marker with the right hand. Move both thumb and index contact points at least 2.5 cm toward its writing tip while keeping it lifted and grasped.
34 Roll Ball Through Gate		Roll or flick the tennis ball from its starting side through the opening between the cracker box and the sugar box. Keep it on the table without picking it up or touching either box.
35 Rotate Propeller		Spin the airplane propeller through three full turns of accumulated travel, in either direction.
36 Scoop Marbles		Keep the spoon level or slightly nose-down, scoop the marble, carry it over the bowl, and pour it into the bowl.
37 Scrub Bottle		Hold the water bottle with the left hand. Brush its lower, middle, then upper bands with the right hand. Release both objects, leaving the bottle upright and stable.

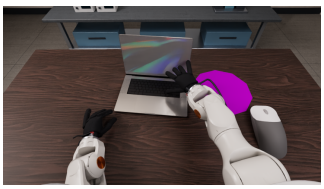
Continued on next page

Table 16: Task profiles of DexOmni (continued).

Task	Screenshot	Task Instruction
38 Set Dual Knobs		Turn both knobs together using their assigned hands, set the left knob to **60 degrees** counter-clockwise and the right knob to **60 degrees** clockwise, release both knobs, and keep the settings stable.
39 Set Timer		Hold the timer with the left hand. Enter **1 minute** and **1 second** with the right, press START last, then withdraw both hands for one countdown tick. Clear incorrect entries first.
40 Slide Knife Blade^{V}		Pick up the utility knife with the right hand, then use the left fingertips on its slider to **extend** the blade fully. (The action goal is random: **extend** may be replaced with **retract**.)
41 Sort Nested Cups		Separate the three nested YCB cups, then place them upright in a straight row from left to right in the main camera: large, medium, small. Release all cups.
42 Sort Objects		Find six targets among the clutter and sort boxes, cans, and dishes into their marked baskets. Transfer the far-right target through both hands before placing it.
43 Squeeze Scissors		Lift the scissors clear of the table and make three complete cuts in the air. Try one hand first; the other hand may reposition or assist.
44 Stack Blocks		Arrange the four blocks into four distinct height layers with the large wood block on the bottom layer, then keep the hand from touching the blocks for 0.5 seconds.
45 Stir Pot		Hold the open pot with the left hand and stir with the right-hand spoon through the six ordered checkpoints and over 650 degrees.

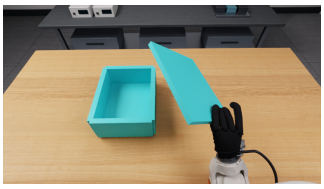
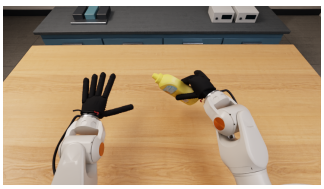
Continued on next page

Table 16: Task profiles of DexOmni (continued).

Task	Screenshot	Task Instruction
46 Store Charger[VA]		Hold the drawer open with the left hand and place the charger fully inside with the right. Release the charger, then the drawer, and hold stable for half a second.
47 Trigger Drill on Block		Use the right hand to bring the drill into contact with the wood block, then press the drill's trigger and hold it for half a second while keeping the contact.
48 Unfold Fan		Use both hands to grasp the two outermost fan ribs and unfold the fan symmetrically to **60 degrees**, then hold it steadily on the table.
49 Unfold Ruler		Pinch the two outer ruler segments with their assigned hands and unfold them together until the ruler is straight.
50 Water Plant[J]		Grasp the spray bottle and aim it at the plant before pressing, then press the sprayer so that the water lands on the plant or its soil.
EVAL-Comb01 Cover Pitcher and Box		First align and seat the pitcher lid, then release it and let it settle. Next align and seat the clear-box lid, release it and let it settle. Both lids must remain correctly seated on their own containers.
EVAL-Comb02 Open Laptop and Click Mouse		Open the initially closed laptop to the **60 degrees** with either hand and let it settle; you may release the screen. Move the mouse onto its pad without pressing either button, then click the left button once. Keep the laptop at the **target angle** and the mouse on the pad while the scene settles.
EVAL-Comb03 Pass Block and Close Box		Pick up the left block with the left hand, hand it to the right hand in the middle, and release it stably inside the clear box. Then use the right hand to pick up the lid, align and close the box, and release.

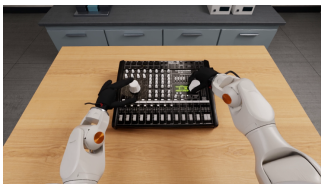
Continued on next page

Table 16: Task profiles of DexOmni (continued).

Task	Screenshot	Task Instruction
EVAL-Comb04 Place Fruit and Photograph Logo		Pick the **orange**, place it in the bucket and release it. Then pick up the camera, aim at the logo on the bucket and press the shutter.
EVAL-Comb05 Scrub Bottle Then Fill Cup		Stabilize the bottle with the left hand and grasp the cleaning brush by its handle with the right hand. Scrub the bottle bands in order, either lower-middle-upper or upper-middle-lower, then release both objects and leave the bottle upright and stable. Next, hold the mug under the faucet with the left hand and open the faucet with the right hand. Keep the stream entering the upright mug for 0.5 seconds, then place the mug on the marked counter pad outside the sink and close the faucet. Leave the scrubbed bottle upright.
EVAL-Comb06 Season and Stir		Use the left hand for salt and the right hand for pepper. Pour both into the same open pot together, then put down and release the jars. Hold the pot with the left hand and stir with the right-hand spoon through all six angular sectors and more than 650 degrees, keeping the pot stable.
EVAL-Gen01 Cover Lunchbox		Align the lunchbox lid and place it onto the lunchbox.
EVAL-Gen02 Open Trash Bin		The trash bin is fixed after reset at a randomized tabletop pose and yaw. Open its hinged lid from closed to at least 75 degrees. Success is immediate when the physical lid reaches that opening. Either hand may be used; no assigned contact, release, or stable hold is required.
EVAL-Gen03 Orient Bottle Upright		Pick up the soap bottle and place it upright on the table with its bottom facing down. Release the bottle and keep it upright for 0.5 seconds.

Continued on next page

Table 16: Task profiles of DexOmni (continued).

Task	Screenshot	Task Instruction
EVAL-Gen04 Pass Glue Stick		Pick up the glue stick with the left hand, transfer it to the right hand in the middle, then place it stably inside the right tray and release it.
EVAL-Gen05 Set Mixer Knobs		Use your left hand to turn the left knob counterclockwise and your right hand to turn the right knob clockwise, in either order. Set the left knob to **60 degrees** and the right knob to **-60 degrees**, release both knobs, and keep the settings stable.
EVAL-Gen06 Spray Shoe		Grasp and aim the spray bottle at the shoe upper before squeezing the trigger. Keep spraying the shoe to complete the task; a spray that entirely misses the shoe upper fails the attempt.
EVAL-Gen07 Stack Boxes		Arrange the four package boxes into four distinct height layers with the large package box on the bottom layer, then keep the hand from touching the boxes for 0.5 seconds.
EVAL-Flex01 Compress Sponge into Mug		Use the left hand to stabilize the YCB mug. Sponge 3 uses an isotropic soft-foam material, so clamp it from any reachable pair of opposing sides and compress it along X, Y, or Z; close slowly until it is compressed, keep the pinch while lifting it edge-on through the mug rim until it is mostly inside, then release both objects and leave the mug upright and stable.
EVAL-Flex02 Fold Towel Twice		Fold the flat towel in half twice, in any directions. The two folds may be in the same direction or different directions. Release the folded towel and leave it stable on the table.
EVAL-Flex03 Pack Pillow		Use both hands to squeeze and lift the foam pillow, lower it into the open carton so it rests inside with most of the pillow below the rim, then release both hands and keep it stable for one second. Tilted placement and raised corners are allowed.