MANIPARENA: A CONTROLLED BENCHMARK FOR DIAGNOSING GENERALIZATION IN REAL-ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

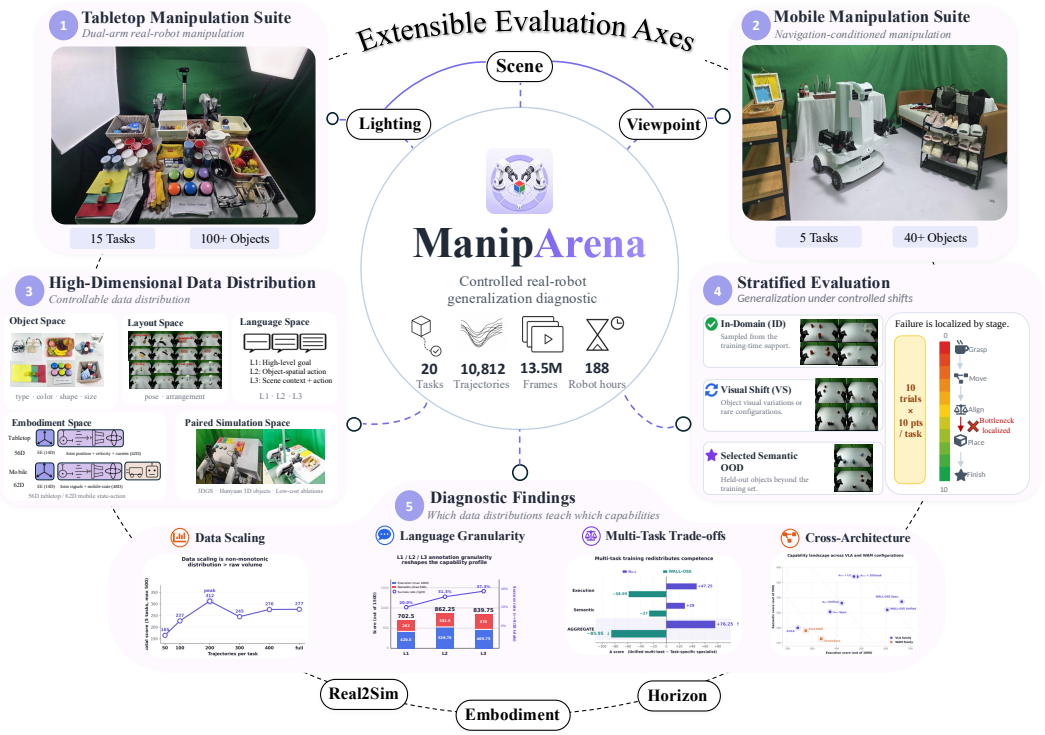


Figure 1: **ManipArena at a glance.** The benchmark comprises 20 tasks, 10,812 trajectories, 13.5M frames, and approximately 188 robot hours. The primary tabletop (**1**) and exploratory mobile (**2**) tracks provide task, language, and embodiment coverage (**3**). A stratified 10-trial protocol and subtask partial-credit scoring (**4**) support diagnostic studies of data scaling, language granularity, task sharing, and system provenance (**5**), with a paired-simulation extension. Language scores and SRs are aggregated from the task results in Appendix F.

ABSTRACT

Vision-Language-Action (VLA) models and world-action models have emerged as central paradigms for general-purpose robotic intelligence, yet their empirical progress remains constrained by the absence of evaluation protocols that are both physically realistic and diagnostically controlled. Simulator-centric benchmarks provide scale and reproducibility, but cannot fully capture the reality gap induced by perception noise, contact dynamics, latency, calibration error, and hardware constraints. Conversely, real-robot evaluations are often fragmented across platforms, scenes, objects, and scoring rules, making fair comparison and failure attribution difficult. We introduce **ManipArena**, a standardized real-robot evaluation framework for studying manipulation generalization under matched physical conditions. ManipArena comprises 20 tasks, 10,812 expert trajectories, 13.5M frames, and approximately 188 robot hours across tabletop and mobile

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.

manipulation. The framework combines schema-defined task variation, stratified in-domain, visual-shift, and task-specific held-out trials, subtask-level partial-credit scoring, three-level language annotations, low-level motor signals, and paired real-to-sim environments reconstructed from physical scenes. Using ManipArena, we evaluate eight tabletop configurations spanning VLA and world-action-model policies, with a single-policy exploratory mobile track. The comparisons examine initialization, fine-tuning, data quantity, and language under specified training budgets. They provide descriptive diagnostics within declared task distributions, rather than established scaling laws or evidence of unconstrained deployment robustness.

1 INTRODUCTION

Recent advances in robot learning have positioned Vision-Language-Action (VLA) models (Physical Intelligence, 2025; Yu et al., 2026; Brohan et al., 2023; Octo Model Team et al., 2024; Zheng et al., 2025; Zhang et al., 2026; Wu et al., 2026) and World-Action-Model (WAM) policies (Ye et al., 2026; Yuan et al., 2026; Wu et al., 2023; Cheang et al., 2024; Bi et al., 2025; Li et al., 2026a) at the forefront of manipulation research. By unifying vision, language, and action within a single policy interface, these models have demonstrated strong performance across tabletop and mobile manipulation, as well as long-horizon instruction following. As capabilities scale, however, evaluation has emerged as a key bottleneck. A policy may appear general under one robot, scene, or task distribution yet fail under another; conversely, aggregate metrics may obscure competent subskills masked by a few physical bottlenecks. Progress therefore requires evaluation protocols that are not only realistic, but also sufficiently standardized and diagnostic to enable interpretable performance assessment.

The dominant evaluation paradigm in robot learning reflects a persistent tradeoff between reproducibility and physical validity. Simulator-centric benchmarks (e.g., RLBench (James et al., 2020), CALVIN (Mees et al., 2022), LIBERO (Liu et al., 2023), ManiSkill (Tao et al., 2025), VLABench (Zhang et al., 2025)) offer reproducible protocols but cannot fully eliminate the reality gap, so simulated performance alone does not establish physical generalization. In contrast, real-robot evaluations capture realistic factors (*e.g.*, perception noise, contact dynamics, latency), yet are often tied to specific setups, limiting comparability across studies. Consequently, attributing performance to reasoning, data, or task design remains challenging.

To address these challenges, we introduce ManipArena, a real-robot evaluation framework for standardized and attribution-aware assessment (Figure 1; Table 1). It is built upon five core principles:

- **Reasoning-Oriented:** ManipArena incorporates complex spatial constraints, multi-stage bimanual operations, and semantic understanding beyond simple motor execution.
- **Multi-Level Generalization:** A green-screen enclosure with fixed lighting, systematic diversity guides, and layered held-out testing controls variation in materials, appearances, and spatial configurations.
- **Exploratory Mobile Track:** Five navigation-conditioned tasks extend the data resource beyond tabletop manipulation, with an initial single-policy evaluation rather than a cross-model mobile comparison.
- **Rich Sensory Diagnostics:** Open sensory signals include low-level motor currents and joint velocities, supporting force-aware action modeling and diagnostic analysis.
- **Real-to-Sim Synchronization:** Paired simulation counterparts reconstructed from physical scenes support data-source diagnostics, with real-robot execution retained as the capability endpoint.

Specifically, ManipArena comprises 20 real-robot tasks, 10,812 trajectories, and approximately 188 hours of demonstrations. The benchmark includes 15 tabletop tasks and 5 mobile-manipulation tasks. Tabletop tasks span both execution-heavy manipulation (*e.g.*, insertion, pouring, bimanual placement, rearrangement) and semantic manipulation (*e.g.*, classification, ordering, matching, category-based selection), while mobile tasks extend to navigation-conditioned, long-horizon scenarios. Tabletop trajectories are annotated at three language levels: L1 provides abstract task instructions, L2 specifies spatially grounded actions, and L3 incorporates scene-level context. This design treats language granularity as a controllable experimental variable rather than a fixed dataset attribute.

Table 1: **Scope of related benchmarks.** Descriptions summarize the cited evaluation designs, not a ranking of reasoning ability or generalization strength.

Benchmark	Env.	Evaluation emphasis
RLBench (James et al., 2020)	Sim	Diverse tasks and generated demonstrations
LIBERO (Liu et al., 2023)	Sim	Knowledge transfer in lifelong robot learning
CALVIN (Mees et al., 2022)	Sim	Language-conditioned long-horizon tasks
ManiSkill2 (Gu et al., 2023)	Sim	Diverse objects, embodiments, and dynamics
VLABench (Zhang et al., 2025)	Sim	Language grounding and long-horizon reasoning
RoboArena (Atreya et al., 2025)	Real	Distributed pairwise policy comparisons
RoboChallenge (Yakefu et al., 2025)	Real	Online real-robot policy evaluation
ManipulationNet (Chen et al., 2026)	Real	Physical-skill and embodied-reasoning tracks
ManipArena (Ours)	Real	Schema-controlled tasks and training diagnostics

Using the ManipArena protocol, we evaluate eight tabletop baseline configurations spanning VLA and WAM policy families, including $\pi_{0.5}$, WALL-OSS-0.5, X-VLA, DreamZero, Fast-WAM, and WALL-WM, with both task-specific and multi-task variants. **Key findings are as follows:**

- **Headroom under tested recipes:** The best reported tabletop score reaches $951/1500$ (63.4%); this does not establish that training has saturated.
- **Nontrivial effects of task sharing:** Multi-task training changes the observed scores in opposite directions for $\pi_{0.5}$ (626 $\rightarrow$ 702.5) and WALL-OSS-0.5 (951 $\rightarrow$ 865).
- **Fixed-update data comparison:** The 200/task recipe obtains a higher observed score than full-data recipes, without isolating data quantity from sample reuse or optimization variability.
- **Task-dependent language profiles:** L2 yields the highest execution score; L3 yields the highest semantic score and SR under the tested rubric (Section 4.2).
- **Exploratory mobile results:** The single task-specific $\pi_{0.5}$ configuration records 0% SR on all five mobile tasks (Appendix E).

Our contributions are threefold:

- We design a green-screen-based evaluation setup with fixed hardware and environment configurations, together with schema-driven task construction and stratified protocols, enabling systematic control over task variation and distribution shifts.
- ManipArena comprises 20 real-robot tasks, 10,812 trajectories, and 188 hours of demonstrations, spanning both tabletop and mobile manipulation. Tabletop trajectories have three language levels (L1–L3), treating language granularity as a controllable experimental variable.
- We compare representative VLA and WAM policies on 15 tabletop tasks and provide training-recipe diagnostics; five mobile tasks form a separate exploratory extension.

2 Related Work

2.1 Benchmarking for Robot Manipulation

A large body of work has focused on developing benchmarks for robot manipulation to evaluate learning-based policies. Early simulation-centric benchmarks, such as RLBench (James et al., 2020), provide a diverse set of tasks with scalable demonstration generation and multi-modal observations, enabling reproducible evaluation across a wide range of learning paradigms. Subsequent efforts extend this direction toward more complex manipulation. For example, CALVIN (Mees et al., 2022) introduces compositional long-horizon tasks grounded in language instructions, emphasizing sequential decision-making and cross-task generalization. LIBERO (Liu et al., 2023) further explores lifelong learning and knowledge transfer, providing structured task suites for evaluating continual learning in embodied agents. More recent benchmarks aim to improve diversity and generalization. ManiSkill2 (Gu et al., 2023) introduces a unified simulation platform with large-scale object diversity, dynamic physics, and multi-task support, significantly improving coverage of manipulation scenarios. VLABench (Zhang et al., 2025) further incorporates reasoning-intensive tasks, requiring models

to integrate language understanding, spatial reasoning, and world knowledge for long-horizon manipulation.

Despite these advances, most existing benchmarks remain predominantly simulation-based. While simulators offer scalability and controllability, their approximations of perception noise, contact dynamics, and system latency require physical validation. Although some recent efforts (Fei et al., 2025; Zhou et al., 2025; Pumacay et al., 2024; Kim et al., 2026) have introduced richer object diversity and controlled perturbations, they still cannot replicate the full visual, physical, and temporal variation of uncontrolled deployment environments. As a result, performance in simulation often does not reliably translate to real-world deployment, limiting their effectiveness for evaluating general-purpose embodied intelligence.

2.2 Real-World Evaluation for Robot Learning

To overcome the limitations of simulation, a growing body of work has explored real-world benchmarks and evaluation frameworks for robot manipulation. FurnitureBench (Heo et al., 2025) introduces a standardized and reproducible platform for long-horizon manipulation through furniture assembly tasks, supported by large-scale teleoperation data and unified system configurations. SceneReplica (Khargonkar et al., 2024) proposes replicable real-world scenes using standardized objects and layouts, enabling consistent evaluation across different laboratories. Platforms such as the Real Robot Challenge (Bauer et al., 2022) provide remote access to standardized robotic systems, allowing policies to be evaluated under consistent hardware and protocols at scale. Recent frameworks including RoboArena (Atreya et al., 2025), RoboChallenge (Yakefu et al., 2025), and ManipulationNet (Chen et al., 2026) further extend this direction by enabling distributed real-world benchmarking and attempting to standardize evaluation pipelines across different settings.

Despite these advances, existing real-world evaluation approaches still exhibit several limitations. First, many benchmarks are task-specific or domain-specific, such as furniture assembly or warehouse picking, limiting their generality. Second, most setups lack systematic control over generalization factors, making it difficult to disentangle the effects of appearance, spatial configuration, and task variation. Third, evaluation priorities differ: ManipulationNet explicitly separates physical skills and embodied reasoning, whereas ManipArena jointly varies task schemas and training recipes under one physical protocol. As a result, there remains a critical need for a unified, controlled, and reasoning-oriented real-world evaluation framework that enables fair comparison, interpretable diagnostics, and systematic analysis of generalization in embodied AI systems.

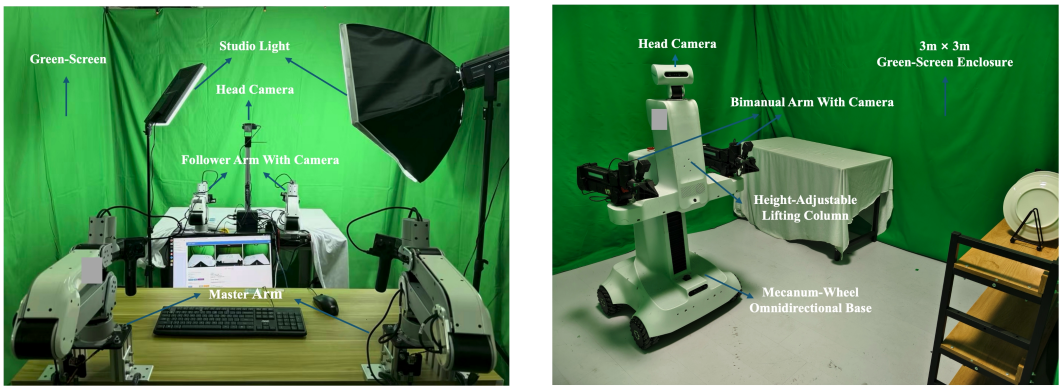
3 ManipArena Benchmark

ManipArena is a schema-driven real-robot benchmark for evaluating manipulation generalization under controlled physical conditions. It contains two robot platforms, 20 tasks, 10,812 demonstrations, a stratified 10-trial evaluation protocol, partial-credit scoring, three-level language annotations, and a paired simulation extension. We describe the benchmark components needed for the main experiments; per-task statistics, observation/action fields, sampling procedures, rubrics, and OOD assignments are deferred to Appendix I–N.

3.1 Controlled setup and benchmark scale

ManipArena uses physical execution as the final measurement. The tabletop platform is a fixed bimanual system with two 6-DoF arms, a head camera, and two wrist cameras; the mobile platform adds a lifting column and omnidirectional base inside a $3\,\mathrm{m} \times 3\,\mathrm{m}$ reconfigurable enclosure (Figure 2). Both platforms run at 20 Hz in green-screen booths with fixed lighting, camera placement, and workspace geometry, and release proprioceptive state/action vectors (Appendix J). Fixed backgrounds and lighting restrict visual variation; contact, sensing, latency, and actuation remain physical. This setup tests schema-defined variation, not robustness to arbitrary backgrounds or lighting.

The corpus contains 20 tasks, 10,812 demonstrations, 13.5M frames, and approximately 188 robot hours (Appendix I). Cross-model studies cover the 15 tabletop tasks; five mobile tasks form an exploratory single-policy track. Table 2 summarizes the task families.



(a) Tabletop platform.

(b) Mobile platform.

Figure 2: Physical platforms used by ManipArena. The tabletop platform supports 15 fixed-base bimanual manipulation tasks, while the mobile platform supports 5 navigation-conditioned tasks in an exploratory track. Both platforms use controlled green-screen environments.

Table 2: ManipArena task families and demonstration counts. The corpus contains 13.5M frames and approximately 188 robot hours.

Family	Tasks	Traj.	Evaluation role
Tabletop execution	10	5,157	Contact, placement, insertion, pouring
Tabletop semantic	5	2,783	Category, order, matching, object selection
Mobile manipulation	5	2,872	Exploratory navigation-conditioned manipulation
Total	**20**	**10,812**	**13.5M frames, 188 hours**

3.2 Schema-driven task design

The central unit of ManipArena is a task schema. A schema specifies the task stages, object and layout variables, distractor condition, and held-out condition reserved for evaluation. Demonstrations and evaluation trials are therefore sampled from declared regions of the same task space, making train/evaluation overlap and held-out variation explicit.

Formally, for task i, a sampled configuration $c = (d_{i,1}, \ldots, d_{i,M_i}, \delta)$ contains the values of its diversity dimensions and distractor condition. The training collection draws

$$c \sim P_i^{\text{train}}, \tag{1}$$

where P_i^{train} is specified by the task schema. The task families separate execution-heavy manipulation, semantic object reasoning, and navigation-conditioned mobile manipulation (Table 2); full schema definitions are given in Appendix K.

3.3 Stratified real-robot evaluation

For each tabletop task, ManipArena uses a fixed 10-trial evaluation protocol. Trials 1–4 sample in-domain configurations, trials 5–8 introduce controlled visual or configuration shifts, and trials 9–10 use the strongest held-out condition available for the task:

$$c_t \sim \begin{cases} P_i^{\text{ID}}, & t \in \{1, \ldots, 4\}, \\ P_i^{\text{shift}}, & t \in \{5, \ldots, 8\}, \\ P_i^{\text{held}}, & t \in \{9, 10\}, \end{cases} \tag{2}$$

where all three distributions are induced by the same task schema. The final band is task-dependent: 7 tabletop tasks use object-level OOD conditions, while the remaining 8 test spatial, ordering, or procedural held-out conditions (Appendix N). Task-level SR has 10-percentage-point resolution; the two-trial held-out band has 50-point resolution. Aggregate results pool the fixed 4/4/2 mixture, not an OOD-only test. Partial credit records progress but does not increase the number of independent trials.

3.4 PARTIAL-CREDIT SCORING

ManipArena scores each trial with task-specific subgoals rather than a binary success label. Each trial is worth 10 points, distributed over stages such as grasping, transport, alignment, insertion, pouring, semantic selection, and final retraction:

$$S_i^{(t)} = \sum_{k=1}^{K_i} w_{i,k}\, \mathbf{1}[g_{i,k} \text{ achieved on trial } t] - \rho_i^{(t)}, \qquad \sum_{k=1}^{K_i} w_{i,k} = 10. \tag{3}$$

Here $\rho_i^{(t)}$ is the task-specific penalty, zero where no penalty applies (Appendix L). For a task subset $\mathcal{I}$, we report aggregate partial-credit score and success rate (reported as a percentage), with success at $9/10$ points:

$$S(\mathcal{I}) = \sum_{i\in\mathcal{I}} \sum_{t=1}^{10} S_i^{(t)}, \qquad \mathrm{SR}(\mathcal{I}) = \frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} \frac{1}{10} \sum_{t=1}^{10} \mathbf{1}\left[S_i^{(t)} \geq 9\right]. \tag{4}$$

Partial credit localizes incomplete progress to grasping, alignment, semantic selection, or completion stages, which binary success would collapse. Rubrics are fixed before evaluation. Two on-site evaluators record subgoals; a third reviews disputed outcomes from video. Per-band aggregation and algorithms are given in Appendix K; rubrics and a scoring example appear in Appendix L.

3.5 LANGUAGE ANNOTATIONS AND SIMULATION EXTENSION

Tabletop trajectories include three language fields: L1 is a task-level instruction, L2 adds episode-specific object and spatial information, and L3 adds a generated scene description. This makes language granularity a controlled training variable in Section 4.2.

ManipArena also includes paired simulation environments for three tabletop tasks. They share task schemas and observation/action conventions with the real setup and are used only as a scoped screening diagnostic in Section 4.5, before real-robot validation.

4 EXPERIMENTS

We evaluate eight VLA/WAM configurations on 15 tabletop tasks, varying data quantity, language, training regime, initialization, or real/sim source under Sections 3.3–3.4. Within each study, interfaces and test prompts remain fixed unless explicitly varied (Appendix C). Primary recipes use one training seed; seed 42 is documented for the data sweep. A separate four-task, three-seed check uses 20k updates. It and two non-identical deployment batches do not supply uncertainty estimates for the main contrasts (Appendix P). Exploratory mobile results remain separate (Appendix E).

4.1 DATA QUANTITY UNDER FIXED UPDATE BUDGETS

We compare unified $\pi_{0.5}$ recipes with 50, 100, 200, 300, or 400 trajectories per task and the full dataset, holding architecture, language, physical protocol, and scoring fixed. Final EMA checkpoints use 60k updates, except a full-data 120k control, without selection using robot outcomes. This is a fixed-update comparison, not an equal-epoch scaling law. Appendix G gives per-task results.

The observed trend is non-monotonic (Figure 3): 200/task reaches 312/500 (42% SR), compared with 277/500 (34%) for Full-60k and 248/500 (26%) for Full-120k. A 15-task check preserves the ordering: 853/1500 (35.3%) for 200/task versus 727.3/1500 (23.3%) for Full-120k (Table 12); only these two recipes were extended to all 15 tasks. Task-bootstrap intervals include zero (Appendix P). Data quantity, sample reuse, and optimization variability remain entangled; neither superiority nor the cause of the non-monotonic curve is established.

Data	Steps	Sc.	SR
50/task	60k	165	10
100/task	60k	227	16
200/task	60k	**312**	**42**
300/task	60k	245	30
400/task	60k	276	36
Full	60k	277	34
Full†	120k	248	26



Figure 3: Fixed-update data comparison for unified $\pi_{0.5}$ on five tasks. Scores are out of 500; SR is in %. †Full uses 120k updates; other rows use 60k. The curve describes these runs, not a general scaling law. Per-task results are in Appendix G.

Optimization-budget control. Extending Full-60k to Full-120k lowers training loss from 0.008163 to 0.006867 but reduces the five-task score from 277 to 248 (Table 11). This extension does not establish convergence across dataset sizes or identify sample reuse as the cause.

4.2 Language Annotation Granularity

We next vary the language field paired with each demonstration while evaluating all policies with the same L1 task-level prompt. The three unified $\pi_{0.5}$ policies differ only in training-time annotation: L1 is a task goal, L2 adds episode-specific object and spatial information, and L3 further adds a generated scene description.

Table 3: Training-time language granularity for unified $\pi_{0.5}$, evaluated with common L1 instructions. L1: task goal; L2: grounded instruction; L3: scene description plus grounded instruction. SR in %. Scores are summed and SRs averaged from Appendix F.

	L1 abstract		**L2 grounded**		**L3 scene+grounded**	
Category	Sc	SR	Sc	SR	Sc	SR
Execution (/1000)	420.50	16.0	**529.75**	26.0	469.75	**28.0**
Semantic (/500)	282.00	28.0	332.50	42.0	**370.00**	**56.0**
Total (/1500)	702.50	20.0	**862.25**	31.3	839.75	**37.3**

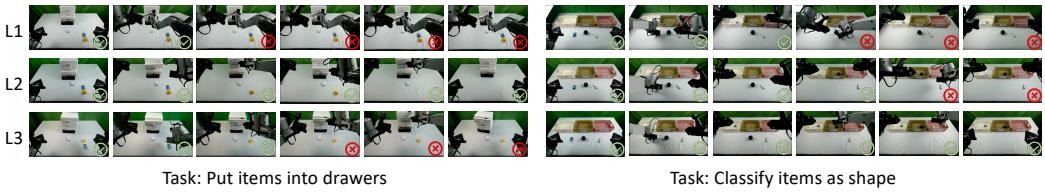


Figure 4: Qualitative comparison of language annotation granularity. Original L1/L2/L3 rollouts for drawer manipulation (left) and shape classification (right) illustrate execution and semantic tasks. Selected sequences do not establish aggregate language effects.

Summing the task scores in Appendix F, L2 obtains the highest execution score (529.75/1000) and total score (862.25/1500), whereas L3 obtains the highest semantic score (370/500) and overall SR (37.3%). L2 exceeds L3 by 60 execution points but trails by 37.5 semantic points. These are descriptive results under the stated rubric, not evidence of a universal language effect. Figure 4 illustrates selected behavior; it does not validate category rankings or rubric weights.

4.3 TASK-LEVEL TRADE-OFFS IN MULTI-TASK TRAINING

We compare task-specific specialists and unified multi-task policies for $\pi_{0.5}$ and WALL-OSS-0.5 on the same 15 tabletop tasks. The robot platform, test prompts, trial protocol, and scoring rule are fixed.

Table 4: Unified minus specialist category scores on 15 tabletop tasks; positive values indicate gains.

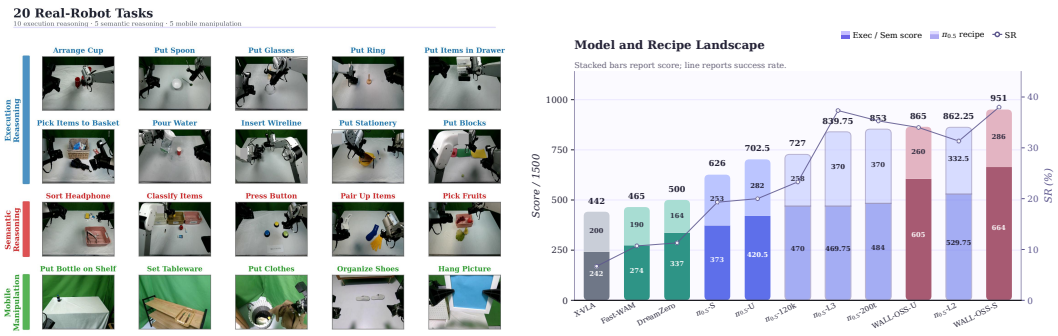
Policy	Δ Exec	Δ Sem	Δ Total
$\pi_{0.5}$	+47.3	+29.0	+76.3
WALL-OSS-0.5	-59.0	-27.0	-86.0

Under unified training, the reported total score changes by +76.3 for $\pi_{0.5}$ and -86.0 for WALL-OSS-0.5 (Table 4). Per-task results show both gains and regressions within each policy family (Appendix E). Task sharing thus changes the distribution of observed performance rather than improving or degrading every task.

The comparison retains each regime's training budget: $\pi_{0.5}$ specialists use 30k updates and the unified policy uses 60k. Paired task-level analyses do not establish a significant aggregate difference (Appendix P); these results describe the tested training regimes rather than an isolated, equal-compute effect of task sharing.

4.4 CROSS-ARCHITECTURE EVALUATION

Under the same tabletop protocol, we compare eight configurations from six policy families: task-specific and unified $\pi_{0.5}$ (Physical Intelligence, 2025) and WALL-OSS-0.5 (Yu et al., 2026), plus unified X-VLA (Zheng et al., 2025), DreamZero (Ye et al., 2026), Fast-WAM (Yuan et al., 2026), and WALL-WM (Li et al., 2026b). Table 5 reports all eight configurations; Figure 5 shows selected policies and training recipes. Each policy retains its initialization and adaptation settings (Appendices B–C).



(a) Task coverage.

(b) Score and success-rate landscape.

Figure 5: ManipArena task suite and evaluation landscape. (a) Ten execution, five semantic, and five exploratory mobile tasks. (b) Selected tabletop systems and recipes; stacked bars report score and the line reports SR. WALL-WM is reported in Table 5, not in this selected panel. Language variants use the task-level sums in Table 3.

The task-specific WALL-OSS-0.5 policy has the highest reported total (950.8/1500, 38% SR), with higher execution score than $\pi_{0.5}$ Unified (664.3 vs. 420.5) and similar semantic score (286.5 vs. 282). The maximum reaches 63.4% of the score ceiling under the tested recipes; it does not demonstrate optimization saturation. These are descriptive system-level results, not statistically established pairwise rankings. Record and score/SR consistency limitations remain in Appendices E and L.

Table 5: Model zoo on 15 tabletop tasks. Scores are category totals (Execution out of 1000, Semantic out of 500). SR is mean success rate (%). Spec. = task-specific fine-tuning; Unified = unified multi-task. Full per-task results in Appendix E.

	$\pi_{0.5}$				WALL-OSS-0.5				X-VLA		Fast-WAM		DreamZero		WALL-WM	
	Spec.		Unified		Spec.		Unified		Unified		Unified		Unified		Unified	
	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR
Exec (/1000)	373	15	420.5	16	**664**	**44**	605	35	242	4	274	9	337	11	570.5	30
Sem (/500)	253	28	282	28	287	26	260	32	200	12	191	14	164	12	**297.5**	**34**
Total (/1500)	626	19	702.5	20	**951**	**38**	865	34	442	7	465	11	500	11	868	31.33

Pretraining control. Architecture-matched controls compare pretrained and from-scratch initialization under the corresponding unified downstream setting (Table 6). The total gains are 357.5 for $\pi_{0.5}$ and 541.8 for WALL-OSS-0.5, covering 14 of 15 tasks (one tie) and all 15 tasks, respectively. Category gains are computed from the complete pairs in Appendix O. The observed gains indicate sensitivity to initialization in these settings, without separating the contributions of pretraining data, objectives, or embodiment alignment.

Table 6: Pretrained versus from-scratch unified policies. Totals are out of 1500; category gains are derived from task pairs, not specialist comparisons (Appendix O).

Architecture	Scratch	Pretrained	Δ Exec	Δ Sem	Δ Total
$\pi_{0.5}$	345.0	702.5	+274.5	+83.0	+357.5
WALL-OSS-0.5	323.0	864.8	+319.8	+222.0	+541.8

4.5 Paired simulation diagnostics

We evaluate the paired simulation extension of Section 3.5 by training $\pi_{0.5}$ on real demonstrations, simulated demonstrations, and a mixed real/sim dataset. The three paired tasks share a real-robot evaluation endpoint; complete scores and success rates are in Appendix H.

Transfer is task-dependent: simulated-data training improves `press_button` (31 to 65/100), whereas real-data training is strongest on `pick_fruits` (78 versus 51/100). Cotraining does not uniformly dominate either source. These results support using paired simulation to compare training-data recipes, with physical evaluation retained as the criterion for real-robot capability.

5 Conclusion

We presented **ManipArena**, a real-robot framework for diagnostic assessment within declared task distributions. Controlled physical environments, schema-driven tasks, and subgoal scoring support tabletop policy comparisons; five mobile tasks form an exploratory extension. Initialization and training recipe comparisons expose task-dependent behavior under the evaluated settings, not universal data-scaling or task-sharing effects. The dataset and paired simulation extension support further controlled studies, with physical execution retained as the evaluation endpoint.

Limitations. Primary recipes use one seed and 10 trials per task; the two-trial held-out band cannot give precise task-wise OOD estimates. Baseline record and score/SR checks still require verification (Appendices E and L). The single-policy mobile track and three-task simulation study have limited coverage; deployment repeatability and unconstrained visual robustness are not established (Appendices M and P).

AI USE STATEMENT

Generative AI tools assisted with manuscript editing, organizational feedback, LaTeX preparation, formatting, and arithmetic and consistency checks on existing experimental tables. These uses did not generate additional robot rollouts or training measurements. The annotation pipeline also includes generated scene descriptions, as described in Section 3.5. The authors remain responsible for the experimental evidence, its interpretation, and all content of the submission.

REPRODUCIBILITY STATEMENT

Sections 3–4 define the experimental protocol and its scope. Appendices C–D report available implementation settings and resource requirements; Appendices J–L describe the data interface, evaluation algorithms, and task rubrics. Complete recorded task results and supplementary controls are provided in the corresponding experimental appendices. Limitations of physical replication and record consistency are stated explicitly rather than resolved by substituting current software defaults or inferred measurements.

REFERENCES

Pranav Atreya, Karl Pertsch, Tony Lee, Moo Jin Kim, Arhan Jain, Artur Kuramshin, Clemens Eppner, Cyrus Neary, Edward Hu, Fabio Ramos, et al. Roboarena: Distributed real-world evaluation of generalist robot policies. *arXiv preprint arXiv:2506.18123*, 2025.

Stefan Bauer, Manuel Wüthrich, Felix Widmaier, Annika Buchholz, Sebastian Stark, Anirudh Goyal, Thomas Steinbrenner, Joel Akpo, Shruti Joshi, Vincent Berenz, et al. Real robot challenge: A robotics competition in the cloud. In *NeurIPS 2021 Competitions and Demonstrations Track*, pp. 190–204. PMLR, 2022.

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, 2025. URL https://arxiv.org/abs/2512.13030.

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

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation, 2024. URL https://arxiv.org/abs/2410.06158.

Yiting Chen, Kenneth Kimble, Edward H Adelson, Tamim Asfour, Podshara Chanrungmaneekul, Sachin Chitta, Yash Chitambar, Ziyang Chen, Ken Goldberg, Danica Kragic, et al. Manipulationnet: An infrastructure for benchmarking real-world robot manipulation with physical skill challenges and embodied multimodal reasoning. *arXiv preprint arXiv:2603.04363*, 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models, 2025. URL https://arxiv.org/abs/2510.13626.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. *The International Journal of Robotics Research*, 44(10-11):1863–1891, 2025.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Ninad Khargonkar, Sai Haneesh Allu, Yangxiao Lu, Balakrishnan Prabhakaran, Yu Xiang, et al. Scenereplica: Benchmarking real-world robot manipulation by creating replicable scenes. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8258–8264. IEEE, 2024.

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

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control, 2026a. URL https://arxiv.org/abs/2601.21998.

Shalfun Li, Victor Yao, Charles Yang, Truth Qu, Regis Cheng, Ryan Yu, Howard Lu, Newton Von, Vincent Chen, Yohann Tang, et al. Wall-wm: Carving world action modeling at the event joints. *arXiv preprint arXiv:2606.01955*, 2026b.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

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

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

Physical Intelligence. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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

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

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation, 2023. URL https://arxiv.org/abs/2312.13139.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic vla foundation model, 2026. URL https://arxiv.org/abs/2601.18692.

Adina Yakefu, Bin Xie, Chongyang Xu, Enwen Zhang, Erjin Zhou, Fan Jia, Haitao Yang, Haoqiang Fan, Haowei Zhang, Hongyang Peng, et al. Robochallenge: Large-scale real-robot evaluation of embodied policies. *arXiv preprint arXiv:2510.17950*, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Ryan Yu, Pushi Zhang, Starrick Liu, Brae Liu, Miracle Kang, Shalfun Li, Lights Shi, Ellie Ma, Ping Yang, Chris Pan, et al. Wall-oss-0.5 technical report. *arXiv preprint arXiv:2605.30877*, 2026.

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

Kaidong Zhang, Jian Zhang, Rongtao Xu, Yu Sun, Shuoshuo Xue, Youpeng Wen, Xiaoyu Guo, Minghao Guo, Weijia Liufu, Liu Zihou, Kangyi Ji, Yangsong Zhang, Jiarun Zhu, Jingzhi Liu, Zihang Li, Ruiyi Chen, Meng Cao, Jingming Zhang, Shen Zhao, Xiaojun Chang, Feng Zheng, Ivan Laptev, and Xiaodan Liang. A1: A fully transparent open-source, adaptive and efficient truncated vision-language-action model, 2026. URL https://arxiv.org/abs/2604.05672.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11142–11152, 2025.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model, 2025. URL https://arxiv.org/abs/2510.10274.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization, 2025. URL https://arxiv.org/abs/2510.03827.

A BROADER IMPACTS

ManipArena can accelerate progress toward more reliable embodied AI agents by making real-robot generalization measurable and interpretable. However, strong performance on the benchmark alone should not be assumed sufficient for real-world deployment. The controlled green-screen environment, fixed lighting, and standardized object sets do not capture the full visual diversity, clutter, and dynamic human presence of uncontrolled homes, offices, or factories. Care should always be exercised when transferring policies evaluated under these conditions to safety-critical applications involving physical human-robot interaction or high-stakes manipulation tasks.

B BASELINE CONFIGURATIONS

The main comparison includes task-specific and unified configurations of $\pi_{0.5}$ and WALL-OSS-0.5, and unified configurations of X-VLA, DreamZero, Fast-WAM, and WALL-WM. A specialist is trained for an individual task; a unified policy is trained jointly across the 15 tabletop tasks. The initialization controls in Appendix O compare the pretrained and from-scratch variants of $\pi_{0.5}$ and WALL-OSS-0.5 under their corresponding unified settings.

For all model-zoo comparisons, baselines are trained on the ManipArena demonstrations using the reported initialization and evaluated with the same 10-trial protocol. No baseline receives additional ManipArena task demonstrations, evaluation examples, or task-specific real-robot data beyond the stated training split. The comparison retains differences in upstream pretraining and model-specific adaptation; it evaluates complete policy configurations under a shared physical protocol.

C BASELINE IMPLEMENTATION CONSISTENCY

For each model family, we use a single implementation and configuration family throughout the corresponding comparisons. This avoids deliberate changes in preprocessing, action representation, or prompting within each study; it does not by itself control stochastic execution or deployment-batch variation. Within each controlled study, we keep the robot platform, observation streams, action interface, normalization assets, evaluation prompts, trial protocol, and partial-credit rubric fixed unless that factor is explicitly the subject of the study.

Controlled changes. Task-specific versus unified training changes the training data grouping while keeping the model family, action interface, test prompts, and evaluation protocol fixed. The data-scaling study changes the number of sampled demonstrations per task at a fixed update budget; sample-reuse frequency consequently differs. The language study changes only the training-time language field, while all policies are evaluated with the same L1 task-level instruction. The Real2Sim diagnostic changes only the data source or mixture used for fine-tuning, followed by the same real-robot evaluation endpoint.

Training and checkpoint selection. Original $\pi_{0.5}$ data-scaling recipes use seed 42, AdamW, global batch size 32, bfloat16, and EMA 0.99. We evaluate the final EMA checkpoint at the pre-specified 60k or 120k update budget, without choosing checkpoints using real-robot outcomes. Task-specific and unified $\pi_{0.5}$ budgets are 30k and 60k, respectively; task sharing is therefore not an equal-compute intervention. These settings do not specify the missing module-level initialization and per-model adaptation details for every other baseline.

Interface adaptation and comparison scope. X-VLA, DreamZero, and Fast-WAM are integrated through their public pipelines, mapping the common LeRobot v2.1 observation keys, camera inputs, and action outputs to the benchmark interface. All systems use the relevant ManipArena downstream split and the same physical evaluation protocol, without held-out data, test-scene adaptation, or hyper-parameter selection against robot outcomes. Native model recipes are retained: these are comparisons of deployable systems, not experiments with equalized upstream training. The initialization controls in Appendix O address a separate, within-architecture question.

Table 7: Implementation consistency across baseline studies. The table records which implementation choices are kept fixed within each model family; the main experiments then vary only the intended diagnostic axis. Mobile policy evaluation is currently limited to one task-specific $\pi_{0.5}$ configuration on the five mobile tasks.

Model family	Runs covered	Fixed implementation choices
$\pi_{0.5}$ / OpenPI	Task-specific, unified, data scaling, language, scoped mobile, Real2Sim	OpenPI policy implementation, LeRobot data loaders, normalization assets, action representation for each platform, and L1 test-time prompting. The task-specific/unified study changes task grouping and retains each regime's budget; data scaling changes sampled demonstration count at fixed updates; language experiments change only the training-time language field.
Fast-WAM	Tabletop model-zoo baseline	Tabletop data adapter, Fast-WAM joint-policy configuration, video preprocessing, text-embedding pipeline, and tabletop action/proprioceptive interface are held fixed across the evaluated tabletop tasks.
DreamZero	Tabletop WAM baseline	Tabletop 14-DOF data/config family, three-view video input convention, relative end-effector action representation, DreamZero policy head, and real-robot inference interface are held fixed.

D EXPERIMENTAL RESOURCES

We report the physical and compute resources used for the ManipArena study to make the evaluation cost and reproducibility assumptions explicit. These numbers describe data collection, real-robot evaluation, and fine-tuning for this paper; they exclude upstream pretraining of external foundation models. The released dataset contains $10{,}812$ trajectories, 13.5M frames, and approximately 188 robot hours collected on the tabletop and mobile robot platforms in controlled green-screen environments. Beyond robot execution time, data collection and evaluation require teleoperation, scene reset, calibration checks, failure recovery, safety supervision, and rubric-based scoring.

Each tabletop evaluation uses 10 real-robot trials per task. The model-zoo study alone covers eight configurations across 15 tabletop tasks, corresponding to at least $8 \times 15 \times 10 = 1{,}200$ tabletop robot trials, before including the data-scaling, language, Real2Sim, and scoped mobile diagnostic run. A complete 15-task policy evaluation comprises 150 physical rollouts and takes approximately 15 system-hours and 30 evaluator-hours, excluding calibration and video review. The seven-recipe, five-task data sweep contains 350 rollouts; the three-level, 15-task language study contains 450. These are study-specific counts, not a claim that every check is an independent repeat of the model zoo. Table 8 summarizes the approximate fine-tuning compute. GPU-hours are computed as the number of GPUs multiplied by wall-clock training time.

Table 8: Approximate fine-tuning compute for baseline checkpoints. GPU-hours exclude upstream pretraining and are intended as resource accounting rather than a hardware-normalized efficiency comparison.

Model / setting	Steps	Hardware	Time	GPU-hours
$\pi_{0.5}$ task-specific	30k	8×A800	6h	48 / checkpoint
$\pi_{0.5}$ unified multi-task	60k	8×A800	20h	160 / checkpoint
WALL-OSS-0.5 fine-tuning	–	8×A800	24h	192 / checkpoint
DreamZero / Fast-WAM	–	16×A800	72h	1,152 / checkpoint

For reference, 15 tabletop $\pi_{0.5}$ task-specific specialists correspond to approximately $15 \times 48 = 720$ GPU-hours. Each 60k-step unified $\pi_{0.5}$ diagnostic run costs approximately 160 GPU-hours, and a 120k-step control run is approximately 320 GPU-hours under linear scaling. The total training cost of the paper is therefore larger than a single model-zoo row, because the data-scaling and language studies train multiple additional diagnostic checkpoints.

E PER-TASK MODEL ZOO RESULTS

Table 9 reports the complete per-task scores and success rates for all eight tabletop baseline configurations. The mobile track is exploratory: one task-specific $\pi_{0.5}$ configuration records 0% SR on each of five tasks, with partial-credit scores from 0 to 14/100. These results neither compare mobile policies nor establish that all policies fail on the track. Some printed per-task sums differ from the reported aggregates; exact precision and evaluation-version alignment require the original unrounded records. The score/SR consistency checks in Appendix L also identify unresolved record–rubric mismatches. The recorded results are retained without inventing replacement values.

Table 9: Per-task scores and success rates. Score: out of 100 (10 trials $\times$ 10 pts). SR: success rate (%, $\geq$9/trial). Spec. denotes task-specific specialist fine-tuning and Unified denotes unified multi-task fine-tuning. Best score per task in **bold**. Mobile policy evaluation is currently limited to the task-specific $\pi_{0.5}$ configuration; unavailable evaluations are marked "–".

		$\pi_{0.5}$				WALL-OSS-0.5											
		Spec.		Unified		Spec.		Unified		X-VLA		Fast-WAM		DreamZero		WALL-WM	
#	Task	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR
Execution Reasoning																	
1	arrange_cup	18	0	47	10	**91**	80	58	20	21	0	55	20	25	0	48	0
2	put_spoon	52	20	32	0	80	40	80	70	38	20	18	10	54	20	**82**	70
3	put_glasses	80	60	67	50	**90**	80	66	60	14	0	38	10	37	0	63	30
4	put_ring	41	20	74	0	**97**	90	91	70	59	10	47	0	27	0	77	40
5	put_items_drw	34	0	1	0	36	0	52	10	16	0	0	0	7	0	**67**	40
6	put_blocks	48	30	83	70	95	80	**96**	90	46	10	45	20	27	0	90	80
7	pour_water	8	0	19	0	**21**	0	19	0	4	0	5	0	12	0	**21**	0
8	insert_wire	24	0	18	0	30	0	**50**	0	15	0	21	0	24	0	42	0
9	pick_items	47	20	67.5	30	90	70	75	30	28	0	41	30	**98**	90	60	40
10	put_stationery	22	0	12	0	**34**	0	19	0	1	0	4	0	26	0	20.5	0
Semantic Reasoning																	
11	sort_headphone	35	20	41	20	70	70	82	80	36	20	32	0	66	60	**84**	60
12	classify_items	59	20	55	10	55	0	31	0	40	0	25	0	36	0	**82**	50
13	press_button	**48**	20	31	0	22	0	16	0	35	20	0	0	3	0	18	0
14	pair_up_items	21	0	**77**	60	68	20	45	10	63	10	63	30	14	0	43.5	20
15	pick_fruits	**90**	80	78	50	72	40	86	70	26	10	71	40	45	0	70	40
Exploratory Mobile Track (task-specific $\pi_{0.5}$ only)																	
16	put_clothes	**4.5**	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–
17	hang_picture	**0**	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–
18	organize_shoes	**8**	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–
19	put_bottle	**14**	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–
20	set_tableware	**10**	0	–	–	–	–	–	–	–	–	–	–	–	–	–	–

F LANGUAGE EXPERIMENT DETAILS

Table 10 gives the task-level language results. L1 is the same unified $\pi_{0.5}$ checkpoint and evaluation batch as in Table 9. Category scores in Table 3 are sums over the corresponding task rows; category SR is the mean task SR. The 15-task total is the sum of all task scores, and overall SR is the mean of all 15 task SRs. Thus, L2 totals 862.25 points (31.3% SR), while L3 totals 839.75 points (37.3% SR). All policies receive the same L1 test instruction. The L1 score/SR checks described in Appendix L also apply here.

Table 10: Language annotation granularity: per-task scores and success rates for $\pi_{0.5}$. Score: out of 100 (10 trials $\times$ 10 pts). SR: success rate (%, $\geq$9/trial). Best score per task in **bold**.

#	Task	L1 Abstract Sc.	L1 Abstract SR	L2 Grounded Sc.	L2 Grounded SR	L3 Scene+Grounded Sc.	L3 Scene+Grounded SR
Execution Reasoning							
1	arrange_cup	47	10	47	10	**52**	30
2	put_spoon	32	0	64	30	**82**	80
3	put_glasses	**67**	50	62	50	**67**	50
4	put_ring	**74**	0	63	10	71	50
5	put_items_drw	1	0	**34**	10	0	0
6	put_blocks	83	70	**90**	90	55	20
7	pour_water	19	0	**22**	0	12	0
8	insert_wire	18	0	**32**	0	25	0
9	pick_items	67.5	30	**89.75**	60	76.75	50
10	put_stationery	12	0	26	0	**29**	0
Semantic Reasoning							
11	sort_headphone	41	20	58	50	**62**	50
12	classify_items	55	10	70	30	**94**	80
13	press_button	31	0	**34**	0	**34**	0
14	pair_up_items	77	60	70.5	30	**80**	50
15	pick_fruits	78	50	**100**	100	**100**	100
Total (/1500)		702.50	20.0	**862.25**	31.3	839.75	**37.3**

G DATA SCALING DETAILS

Table 13 reports the per-task score and success rate for each data recipe.

Scope of the fixed-update comparison. The five-task sweep measures the performance of the stated training recipes at fixed optimizer-update budgets. Changing demonstration count also changes sample reuse and potentially the distribution of training examples. Equal updates do not imply equal epochs, and trajectory counts alone do not specify exposure without the sampling-unit definition. The observed curve does not identify whether composition, exposure, or run-to-run variability causes its shape. Equal-exposure controls, a wider budget sweep, and repeated training would be needed to separate these explanations; these are not reported experiments.

Optimization-budget control. Table 11 places the training-loss endpoints and physical outcomes of Full-60k and Full-120k side by side. Both use seed 42, AdamW, global batch size 32, bfloat16, and EMA 0.99. The final EMA checkpoints are fixed by the update budget, not selected from intermediate checkpoints using robot outcomes. Extending the full-data recipe reduces training loss while lowering its measured physical score and SR. This is a within-recipe budget check, not evidence that all sampling recipes have converged. Training losses across different data distributions or reuse frequencies are not interchangeable measures of policy quality.

Table 11: Full-data optimization-budget control for unified $\pi_{0.5}$. Five-task physical evaluation uses the same subset as Figure 3. Loss endpoints come from the supplementary training diagnostics.

Recipe	Updates	Endpoint loss	Score /500	SR (%)
Full-60k	60,000	0.008163	277	34
Full-120k	120,000	0.006867	248	26

Fifteen-task coverage check. The two configurations in Table 12 were also evaluated on all 15 tabletop tasks. Their ordering agrees with the five-task sweep, but this extension remains one seed per recipe; it is not a 15-task evaluation of all seven recipes or a cross-architecture data-scaling study.

The complete 15-task rows for these two checkpoints are not supplied, so we report their provided aggregates without assembling rows from other checkpoints.

Table 12: Fifteen-task extension of two $\pi_{0.5}$ data recipes. The table reports provided aggregate scores; the remaining five recipes were not all evaluated on this larger task set.

Training recipe	Updates	Score /1500
200 demonstrations/task	60,000	853.0
Full data	120,000	727.3

Table 13: Detailed data-scaling results for the unified $\pi_{0.5}$ policy on five representative tasks. Entries are score / SR (%); Total SR is averaged over the five evaluated tasks. Best score per task is shown in **bold**.

	glasses		blocks		wire		button		fruits		Total	
Data	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR	Sc	SR
50/task	29	0	16	0	21	0	22	0	77	50	165	10
100/task	47	0	66	50	29	0	19	0	66	30	227	16
200/task	69	50	76	60	24	0	**54**	20	89	80	**312**	**42**
300/task	35	20	73	70	19	0	37	10	81	50	245	30
400/task	**74**	70	43	20	26	0	40	0	**93**	90	276	36
Full	67	50	83	70	18	0	31	0	78	50	277	34
Full[†]	49	30	**93**	90	**33**	0	27	0	46	10	248	26

[†]Full-data model trained for 120k steps; all other rows use 60k steps.

H REAL2SIM TRANSFER DETAILS

Table 14 reports the real-robot evaluation of checkpoints trained with real demonstrations, simulated demonstrations, and a 50% mixed real/sim recipe. The paired simulation extension is used as a diagnostic data source; real-robot trials remain the endpoint for capability assessment.

Table 14: Real2Sim transfer ($\pi_{0.5}$). Three checkpoints evaluated on the real robot. SR in %.

	Real ckpt		Sim ckpt		Cotrain 50%	
Task	Sc	SR	Sc	SR	Sc	SR
put_blocks	83	70	**85**	70	**85**	70
press_button	31	0	**65**	50	40	20
pick_fruits	**78**	50	51	30	68	50

I PER-TASK DATASET STATISTICS

Table 15 reports the demonstration count and primary capability for each task. The corpus comprises 10,812 trajectories, 13.5M frames, and approximately 188 robot hours. Of these trajectories, 7,940 belong to the 15 tabletop tasks and 2,872 to the five-task exploratory mobile track.

Table 15: Per-task demonstration counts and capabilities. The complete corpus contains 13.5M frames and approximately 188 robot hours.

#	Task	Family	Traj.	Primary capability
1	`arrange_cup`	Execution	528	Bimanual stacking and placement
2	`put_spoon`	Execution	525	Object placement into container
3	`put_glasses`	Execution	513	Bimanual shelf placement
4	`put_ring`	Execution	517	Ring threading and alignment
5	`put_items_drawer`	Execution	510	Drawer opening, placement, closure
6	`put_blocks`	Execution	451	Color-conditioned placement
7	`pour_water`	Execution	526	Pouring and container handling
8	`insert_wireline`	Execution	530	Cable handoff and insertion
9	`pick_items`	Execution	532	Repeated object transport
10	`put_stationery`	Execution	525	Zipper and container manipulation
11	`sort_headphone`	Semantic	515	Category-based selection
12	`classify_items`	Semantic	545	Shape classification
13	`press_button`	Semantic	538	Ordered button pressing
14	`pair_up_items`	Semantic	540	Relational matching
15	`pick_fruits`	Semantic	645	Semantic object selection
16	`put_clothes`	Mobile	540	Navigation-conditioned placement
17	`hang_picture`	Mobile	576	Mobile bimanual hanging
18	`organize_shoes`	Mobile	595	Mobile pickup and arrangement
19	`put_bottle`	Mobile	630	Mobile shelf placement
20	`set_tableware`	Mobile	531	Long-horizon table setting
Total			**10,812**	**13.5M frames, approximately 188 hours**

Figure anonymization. Small branding marks on robot housings in Figures 1, 2, 4, and 5 are masked for anonymous review. Task objects, action sequences, annotations, and plotted measurements are unchanged.

J DATA FORMAT AND OBSERVATION/ACTION LAYOUT

ManipArena trajectories are stored in LeRobot v2.1 format. Tabletop tasks use 56-dimensional state and action vectors. Mobile manipulation tasks add 6 dimensions for head, lifting column, and chassis, giving 62D in total. The end-effector frame uses $+x$ forward, $+y$ left, $+z$ up. The last element of each 7D joint group (indices 20, 27, 34, 41, 48, 55) is the gripper channel. Table 16 lists the full per-dimension layout.

Table 16: Per-dimension layout of ManipArena state and action vectors. Tabletop tasks use indices 0–55 (56D); mobile tasks additionally use indices 56–61 (62D).

Index	Field	Dim	Description
End-effector (14D)			
0–2	`follow_left_ee_cartesian_pos`	3	Left arm position (x, y, z)
3–5	`follow_left_ee_rotation`	3	Left arm rotation (roll, pitch, yaw)
6	`follow_left_gripper`	1	Left gripper open/close
7–9	`follow_right_ee_cartesian_pos`	3	Right arm position (x, y, z)
10–12	`follow_right_ee_rotation`	3	Right arm rotation (roll, pitch, yaw)
13	`follow_right_gripper`	1	Right gripper open/close
Left arm joints (21D)			
14–20	`follow_left_arm_joint_pos`	7	Joint positions (6 joints + gripper)
21–27	`follow_left_arm_joint_dev`	7	Joint velocities (6 joints + gripper)
28–34	`follow_left_arm_joint_cur`	7	Joint currents (6 joints + gripper)
Right arm joints (21D)			
35–41	`follow_right_arm_joint_pos`	7	Joint positions (6 joints + gripper)
42–48	`follow_right_arm_joint_dev`	7	Joint velocities (6 joints + gripper)
49–55	`follow_right_arm_joint_cur`	7	Joint currents (6 joints + gripper)
Mobile extras (mobile tasks only, 6D)			
56–57	`head_actions`	2	Head rotation (yaw, pitch)
58	`height`	1	Lifting-column height
59–61	`velocity_decomposed_odom`	3	Chassis velocity (v_x, v_y, ω)

ManipArena additionally releases joint velocities and joint currents per timestep (indices 21–34, 42–55), providing additional signals related to contact and actuation dynamics that are useful for compliant- or impedance-based control policies. For mobile trials, the head, lifting-column, and chassis channels are released alongside the arm channels so that navigation-conditioned manipulation policies can be trained and evaluated on a uniform observation/action interface.

K Evaluation protocol details

This appendix collects the formal procedures referenced from Section 3.

Schema specification and sampling. The training-time distribution in Equation 1 is given by the per-task schema $\mathcal{S}_i = \{(D_{i,j}, p_{i,j})\}_{j=1}^{M_i}$, where $D_{i,j}$ is the j-th diversity dimension and $p_{i,j}$ is a categorical distribution over its admissible values. In the common case, $P_i^{\text{train}}(c) = \prod_{j=1}^{M_i} p_{i,j}(d_{i,j})\, p_i^{\delta}(\delta)$. Other tasks use joint distributions over subsets of dimensions (e.g., enumerated 4-item combinations in `pick_items_basket`, fixed orderings in `press_button`) or refresh certain dimensions only at batch boundaries (the `change_frequency` field). The full per-task schema (diversity dimensions, distributions, and constraints) is released alongside the dataset. Algorithm 1 summarizes the data-collection loop.

Algorithm 1 Schema-driven data collection for task i

Require: Schema $\mathcal{S}_i = \{(D_{i,j}, p_{i,j})\}_{j=1}^{M_i}$; target trajectory count N_i
Ensure: Trajectory set $\mathcal{D}_i$
$\mathcal{D}_i \leftarrow \emptyset$
while $|\mathcal{D}_i| < N_i$ **do**
  Sample a configuration $c = (d_{i,1}, \ldots, d_{i,M_i}, \delta) \sim \mathcal{S}_i$, where δ is the distractor condition specified by the schema
  Reset scene to c
  Collect teleoperated demonstration to obtain trajectory τ
  Generate three-level annotation $\ell = (\ell_{L1}, \ell_{L2}, \ell_{L3})$
  $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{(\tau, \ell)\}$
end while
return $\mathcal{D}_i$

Per-band aggregation. The stratified distributions in Equation 2, the per-trial score in Equation 3, and the aggregate metrics in Equation 4 are used for all real-robot results in Section 4. Per-band score means provide a finer-grained breakdown than the cross-task aggregate:

$$\bar{S}_i^B \;=\; \frac{1}{|B|} \sum_{t \in B} S_i^{(t)}, \qquad B \in \{T_{1\text{-}4},\, T_{5\text{-}8},\, T_{9\text{-}10}\}. \tag{5}$$

The single-task success rate is the special case $\mathrm{SR}_i = \mathrm{SR}(\{i\})$ of Equation 4 and counts a trial with score $\geq 9/10$ as a success, providing a near-completion signal alongside the partial-credit total.

Algorithm 2 Stratified 10-trial evaluation of policy π on task i

Require: Policy π; schema $\mathcal{S}_i$; rubric $\{(w_{i,k}, g_{i,k})\}_{k=1}^{K_i}$
Ensure: Per-trial scores $\{S_i^{(t)}\}_{t=1}^{10}$
for $t = 1, \ldots, 10$ **do**
  if $t \leq 4$ **then** ▷ in-domain
    Sample c_t from the training-time support of $\mathcal{S}_i$
  else if $t \leq 8$ **then** ▷ visual / configuration shift
    Sample c_t as a controlled visual or configuration shift specified by $\mathcal{S}_i$
  else ▷ strongest held-out
    Sample c_t from the strongest available held-out condition (Appendix N)
  end if
  Reset scene to c_t and execute π to produce a trajectory
  Compute $S_i^{(t)}$ using Equation 3, including any rubric-defined penalty
end for
return $\{S_i^{(t)}\}_{t=1}^{10}$

L SCORING RUBRIC SUMMARY

Rubrics were specified before model evaluation from task schemas, demonstrations, and collector input. Two on-site evaluators record milestones, and a third reviews disputed outcomes from video. The 9/10 threshold is operational: the cup example below is functionally complete before one-point retraction. This convention is not uniform across mobile tasks (the supplied `put_bottle` rubric assigns two points to retraction). The `pick_fruits` penalty follows its task-specific rule; exact normalization and clipping require the scoring implementation.

Score–success consistency. Under the stated binary-subgoal rubric, `put_ring` and `put_blocks` have weights $(3, 3, 3, 1)$. A failed trial therefore scores at most 7, so ten trials with m successes score at most $7(10 - m) + 10m$. The recorded unified $\pi_{0.5}$ `put_ring` result (74 points, 0% SR) and specialist WALL-OSS-0.5 `put_blocks` result (95 points, 80% SR) exceed the respective bounds of 70 and 94.

Table 17: Subgoal scoring example for cup arrangement. Credits sum to 10.

Observable milestone	Credit	Cumulative
Grasp the first cup	1	1
Place it upside down in the center	1	2
Grasp the second cup	1	3
Place it beside the first	2	5
Grasp the third cup	1	6
Place it on top of the lower pair	3	9
Retract the arm	1	10

Table 18: Task scoring rubrics used for partial-credit evaluation. Each trial is worth 10 points.

Task	Scoring stages
`arrange_cup`	Pick cup1 (1), place cup1 (1), pick cup2 (1), place cup2 (2), pick cup3 (1), stack apex cup (3), retract (1)
`put_spoon`	Pick bowl (2), place bowl center (2), pick spoon (2), place spoon in bowl (3), retract (1)
`put_glasses`	Pick glasses (2), place center (2), adjust orientation (2), bimanual shelf placement (3), retract (1)
`put_ring`	Move rod to center (3), pick ring (3), thread ring onto rod (3), retract (1)
`put_items_drawer`	Three repetitions of open drawer (1), place item (1), close drawer (1), plus retract (1)
`put_blocks`	Place red block (3), place yellow block (3), place green block (3), retract (1)
`pour_water`	Move cup to center (2), grasp kettle (2), pour (3), place kettle back (2), retract (1)
`insert_wireline`	Left hand picks cable (3), right hand receives cable (3), insert into socket (3), retract (1)
`pick_items`	Four repetitions of pick item (1) and place in basket (1.25), plus retract (1)
`put_stationery`	Hold case (1), unzip (1.5), open (1), place three items (3), hold case (1), zip (1.5), retract (1)
`sort_headphone`	Find headphone in clutter (2), pick up (3), place in basket (4), retract (1)
`classify_items`	Arrange first item (3), arrange second item (3), arrange third item (3), retract (1)
`press_button`	Press first button (3), press second button (3), press third button (3), retract (1)
`pair_up_items`	Pick match for upper-left item (2), place below (2.5), pick match for upper-right item (2), place below (2.5), retract (1)
`pick_fruits`	Pick each target fruit with weights normalized by target count; wrong picks receive a one-point penalty; retract completes the trial
`put_clothes`	Navigation and manipulation stages for shirt, pants, and socks: navigate, pick, transport, place, and final retraction
`hang_picture`	Navigate, bimanual pick (3), navigate, raise picture (2), hang picture (2), retract
`organize_shoes`	Navigate, pick left shoe (2), pick right shoe (2), navigate, place left shoe (1.5), place right shoe (1.5), retract
`put_bottle`	Navigate (2), pick bottle (2), navigate (2), place on shelf (2), retract (2)
`set_tableware`	Navigate, pick basket (1), navigate, place basket, place plate, place knife (2), place fork (2), retract

M DETAILED LIMITATIONS

Real-robot stochasticity. ManipArena evaluates policies on physical robots, where repeated executions of the same checkpoint on nominally identical trials can still vary due to contact conditions, object pose perturbations, sensing noise, latency, controller variation, and reset differences. The protocol fixes the trial mixture and partial-credit scoring preserves intermediate progress. Neither establishes lower measurement variance or adds independent trials. Ten trials per task and two in the held-out band provide coarse estimates; a shared task schema does not guarantee matched physical execution. Repeated training and repeated deployment measure different sources of variability (Appendix P).

Cost and coverage. Real-robot evaluation requires robot time, scene reset, failure recovery, and hardware supervision; the controlled studies also require training multiple checkpoints across data

recipes, language fields, and model families. This cost limits the breadth of the current sweep. The tabletop model zoo includes eight configurations. The exploratory mobile track has one task-specific $\pi_{0.5}$ configuration with 0% SR on all five tasks; its low partial-credit scores demonstrate limited progress for this configuration, not a cross-policy conclusion. The paired simulation extension covers three tabletop tasks. These choices provide controlled evidence for the factors studied in the main paper, but do not exhaust all robot foundation models, training schedules, prompt formats, data mixtures, or real/sim recipes.

Platform and environment scope. The controlled green-screen environments restrict background and lighting variation; they do not replace real contacts, sensors, actuators, or control latency with simulated dynamics. Visual external validity is nevertheless limited: robustness to natural illumination, arbitrary backgrounds, and uncontrolled scene changes is not measured. The purpose is controlled physical diagnosis within the specified schemas, not full deployment realism. The current results are also tied to the tested robot embodiments, object sets, reset procedures, and scoring rubrics. Future instantiations can reuse the same task schemas and partial-credit framework while varying scene, lighting, camera viewpoint, object set, or robot embodiment.

Instruction interface. The cross-model evaluation uses a common high-level L1 prompt to keep the test-time instruction interface fixed across VLA and WAM policies. This supports controlled comparison, but it does not characterize every prompting regime. Some policy families may benefit from more explicit scene-grounded or action-level instructions. Systematically varying test-time prompt granularity and model-specific instruction interfaces is therefore left for future work.

N OOD AUTHENTICITY AUDIT

The 10-trial stratification (Section 3.3) reserves trials T9–T10 for the strongest held-out condition available per task. Whether that condition introduces an unseen object depends on whether object identity is part of the task semantics. Table 19 records the per-task assignment used in this paper.

Table 19: Per-task held-out condition for the 15 tabletop tasks. *Object OOD* tasks introduce previously unseen objects at T9–T10. *Configuration OOD* tasks hold out spatial, ordering, or procedural combinations from the training schema while reusing in-distribution objects.

Task	T5–T8 shift	T9–T10 condition
Object OOD (7 tasks)		
`arrange_cup`	mug → paper cup	plastic cup (blue)
`sort_headphone`	wired → over-ear → Bluetooth	neckband sport (white)
`put_spoon`	stainless → children's spoon	black plastic spoon
`put_glasses`	adult → children's → goggles	sunglasses
`put_items_drawer`	object-combination change	marker, watch, top
`put_stationery`	color-combination change	fountain pens
`pick_fruits`	banana introduced + new basket	orange/cherry + toy doll
Configuration OOD (8 tasks)		
`insert_wireline`	white → black cable	rare position combinations
`put_blocks`	block-size combinations	rare color orderings
`pour_water`	cup-type combinations	rare cup-position pairings
`put_ring`	ring-position variation	rare ring/rod-position pairings
`press_button`	button-order permutations	held-out button orders
`classify_items`	shape-order permutations	held-out shape orders
`pair_up_items`	matching-pair combinations	held-out match pairs
`pick_items_basket`	basket position + item subset	basket-left configurations

For the 8 Configuration OOD tasks, T9–T10 stresses spatial, ordering, or procedural generalization rather than object semantics. As discussed in Section 3.3, this separation is intentional. Future versions of the benchmark can introduce object OOD for the remaining 8 tasks (e.g., triangular rings for `put_ring`, orange or purple buttons for `press_button`) without altering the partial-credit framework or the controlled environment.

O SUPPLEMENTARY PRETRAINING CONTROLS

Table 20 reports the task-level initialization controls. Each entry is a partial-credit score out of 100. The pretrained $\pi_{0.5}$ and WALL-OSS-0.5 columns correspond to their unified settings, not their specialist results. WALL-WM and Fast-WAM are each evaluated as a from-scratch configuration and provide system-level context, rather than additional paired initialization comparisons.

Table 20: Task-level initialization controls. FS denotes from scratch and PT denotes pretrained; paired comparisons use the unified configurations.

	$\pi_{0.5}$		WALL-WM	Fast-WAM	WALL-OSS-0.5	
Task	FS	PT	FS	FS	FS	PT
arrange_cup	8	47	48	55	35	58
put_spoon	22	32	82	18	52	80
put_glasses	18	67	63	38	28	66
put_ring	33	74	77	47	15	91
put_items_drw	1	1	67	0	27	52
put_blocks	3	83	90	45	66	96
pour_water	7	19	21	5	5	19
insert_wire	15	18	42	21	19	50
pick_items	35.5	67.5	60	41.25	29.5	74.8
put_stationery	3.5	12	20.5	4	9	18.5
sort_headphone	26	41	84	32	0	82
classify_items	28	55	82	25	19	31
press_button	13	31	18	0	0	16
pair_up_items	71	77	43.5	62.5	4.5	44.5
pick_fruits	61	78	70	71	14	86
Total (derived sum)	345.0	702.5	868.0	464.75	323.0	864.8

Category-wise initialization gains. Using the first ten execution and final five semantic task rows, the category scores are 146.0 $\rightarrow$ 420.5 and 199.0 $\rightarrow$ 282.0 for $\pi_{0.5}$, and 285.5 $\rightarrow$ 605.3 and 37.5 $\rightarrow$ 259.5 for WALL-OSS-0.5. Subtraction gives the Δ Exec and Δ Sem columns in Table 6. Pretraining increases $\pi_{0.5}$ scores on 14 tasks, with a tie on `put_items_drw`; it increases WALL-OSS-0.5 scores on all 15 tasks. These counts and category sums are arithmetic summaries of the same 15 pairs, not additional experimental repetitions. No matched pretrained comparison is reported for WALL-WM or Fast-WAM.

Interpretation and configuration scope. The controls compare initialization within each architecture under the corresponding ManipArena downstream setting. The totals summarize the same task-level pairs. They describe differences in these runs, but do not separately identify the effects of pretraining scale, data quality, objectives, or embodiment alignment. Module-level randomization and paired optimization manifests remain necessary for exact replication; current public code defaults must not be substituted for historical run settings.

P SUPPLEMENTARY RELIABILITY AND TRAINING CHECKS

P.1 INDEPENDENT TRAINING SEEDS

Three $\pi_{0.5}$ policies use 20k updates with the same data, hyperparameters, checkpoint selection, and evaluation protocol. Table 21 reports the four illustrative tasks. Sample SD spans independently trained and deployed policies, not repeated trials of one fixed policy.

Table 21: Three-seed check at 20k steps. Scores are out of 100 per task.

Task	Seed 42	Seed 43	Seed 44	Mean $\pm$ sample SD
sort_headphone	76	75	76	75.67 ± 0.58
put_stationery	14	10.5	12	12.17 ± 1.76
press_button	19	15	22	18.67 ± 3.51
insert_wire	31	32	25	29.33 ± 3.79

These four tasks show different observed dispersions, with sample SDs of 0.58–3.79 points on the 100-point scale. The 20k setting is not a replication of the primary 60k comparisons; no full-suite seed-wise total is inferred, and these SDs are not used as error bars for other recipes. Optimization variability and stochastic execution are not separately identified.

P.2 Task-level uncertainty

The supplementary paired analyses resample task-level scores, not individual rollouts or training seeds. For the five-task data sweep, the reported bootstrap intervals for 200/task against Full-60k and Full-120k include zero. The reported paired analyses also indicate that the unified–specialist comparisons do not establish significant pairwise effects. These findings limit claims of superiority without negating the observed task-wise differences.

The retained statistical records do not contain complete numerical intervals and sign-flip outputs, so no additional numerical estimates are reported here. The language comparison in Appendix F reports row-summed scores and mean success rates descriptively; it makes no claim of statistical significance or cross-seed replication. A two-trial held-out band gives only 50-percentage-point success-rate resolution; a ten-trial task gives 10-point resolution. These increments describe the metric's granularity, not a confidence interval. Cross-task resampling probes sensitivity to task composition; it cannot recover unobserved rollout variance, remove deployment-batch effects, or establish replicability across training seeds. A nonsignificant contrast is not evidence of equivalence.

P.3 Repeated physical deployment

Two evaluations of the same frozen $\pi_{0.5}$ checkpoint obtained 644.5/1500 and 702.5/1500. Model-side settings were unchanged, but the second batch used a more stable robot client. The 58-point difference is descriptive end-to-end deployment variability, not a pure estimate of model or training variance. Task resampling, repeated training, and repeated deployments have different units of variation and must not be pooled into one error bar. The 58-point gap is 3.87 points after normalization to a 0–100 scale. With only two batches and a changed client condition, it neither isolates a client causal effect nor estimates a common 58-point error floor for other policies. It cannot establish that every smaller difference is noise, or that a larger one is reliable. Repeated, matched deployments would be needed to quantify batch effects on pairwise comparisons.

Table 22: Two physical evaluations of one frozen $\pi_{0.5}$ checkpoint. The client-stability difference prevents interpreting the observed gap as pure policy or training-seed variance.

Evaluation batch	Score /1500	Reported deployment condition
First	644.5	Original robot-client condition
Second	702.5	More stable robot client