EBench: Elemental Diagnosis of Generalist Mobile Manipulation Policies

Ning Gao[1,2], Jinliang Zheng[1,3], Xing Gao[1], Haoxiang Ma[1], Hanqing Wang[†1], Yukai Wang[1], Jiantong Chen[1], Zanxin Chen[1], Shujie Zhang[1,6], Mingda Jia[1], Xuekun Jiang[1], Zihou Zhu[1], Xinyu Li[1], Shuai Wang[1], Hao Li[1,7], Wenzhe Cai[1], Yuqiang Yang[1], Xudong Xu[1], Zhaoyang Lyu[1], Yao Mu[1,4], Tai Wang[1], Jiangmiao Pang[1], Jia Zeng[1], Weinan Zhang[1,4] and Chunhua Shen[1,5]

[1]Shanghai AI Laboratory, [2]Xi'an Jiaotong University, [3]Institute for AI Industry Research (AIR), Tsinghua University, [4]Shanghai Jiao Tong University, [5]Zhejiang University, [6]Tsinghua University, [7]University of Science and Technology of China

Project Page | Code | Leaderboard

We present EBench, a simulation benchmark that diagnoses generalist mobile manipulation policies beyond a single success-rate scalar. EBench comprises 26 diverse and challenging manipulation tasks annotated along 5 capability dimensions and 4 generalization dimensions. We evaluate state-of-the-art generalist manipulation models including π_0, $\pi_{0.5}$, XVLA, and InternVLA-A1, and reveal that models with near success rates exhibit strikingly different capability profiles: $\pi_{0.5}$ achieves the highest test success rate and the best train–test retention, whereas InternVLA-A1 dominates mobile manipulation but collapses on dexterous tasks, and XVLA exhibits strengths on a disjoint set of atomic skills compared to other policies. Beyond capability profiling, EBench analyzes the generalization ability from 4 representative perspectives, identifying the impact of different distribution shift factors. The results reveal strengths and weaknesses of models behind an overall score. We hope this benchmark offers a broad set of diagnostic signals to guide iteration on generalist manipulation models.

1. Introduction

Despite substantial progress of simulation benchmarks, thoroughly evaluating general-purpose manipulation policies remains challenging. State-of-the-art generalist manipulation policies now report success rates on contemporary simulation suites to demonstrate their superior performance. However, there are fundamental questions that aggregate numbers cannot answer: *Where is a model strong, where does it break*, and *how does that pattern shift as the deployment distribution drifts away from the training distribution?*

The gap is structural. Single-scene tabletop suites such as RLBench (James et al., 2020), CALVIN (Mees et al., 2022), and LIBERO (Liu et al., 2023) cover a narrow slice of physical interaction. Larger-scale efforts such as RoboCasa (Nasiriany et al., 2024), RoboTwin (Chen et al., 2025; Mu et al., 2025), and GenManip (Gao et al., 2025) broaden task and embodiment coverage, but each focuses on a single regime: tabletop pick-and-place, mobile rearrangement, or one-shot manipulation. Diagnostic benchmarks such as RMBench (Chen et al., 2026) isolate a single capability axis, namely memory. BEHAVIOR-1K (Li et al., 2023) illustrates broader task types, while falls back to overall scores. The community yearns for a benchmark whose task suite is broad enough to cover long-horizon, dexterous, and mobile regimes together, and well-instrumented to support fine-grained analysis rather than a single leaderboard scalar. Motivated by this urgency, we introduce **EBench**, a simulation benchmark for generalist manipulation that addresses these three needs together. EBench has three core contributions:

[†]Corresponding author: Hanqing Wang (hanqingwang.c@gmail.com).

Figure 1. **EBench** is a simulation benchmark for generalist embodied manipulation that, within a single evaluation suite, simultaneously covers long-horizon, dexterous-and-precise, and mobile manipulation across 9 scene categories. Each of the 26 tasks is tagged along 5 capability axes and paired with 4 controlled generalization dimensions, so that a single scalar success rate decomposes into an interpretable capability profile.

1. **A benchmark codebase for mobile manipulation.** EBench's open-source infrastructure bundles three pieces normally maintained in isolation: a two-stream *data-synthesis* pipeline that combines human teleoperation for dexterous-and-precise tasks with a key-frame-pose plus cuRobo (Sundaralingam et al., 2023) motion planner for mobile and long-horizon tasks; a composable *scoring library* that assembles per-task success and partial-progress metrics from a shared set of evaluation primitives, including scene-graph relations between objects, articulation joint angles, object tilt and orientation, and temporal-ordering constraints over sub-goals; and a distributed *evaluation runner* that completes the full validation split on 8 consumer GPUs in roughly 30 minutes.
2. **Wide-spectrum tasks with rich annotations.** On top of this codebase we assemble 26 tasks that span three families rarely co-exist in a single suite: 10 mobile pick-and-place tasks, 9 long-horizon multi-stage tasks, and 7 dexterous-and-precise tasks with sub-centimetre tolerance. Scene assets are sourced from GRUtopia (Wang et al., 2024) and InternScenes (Zhong et al., 2026) and object assets from Objaverse (Deitke et al., 2023); Each task is then manually annotated along five dimensions: scene type, atomic skill, temporal horizon, precision, and operating mode. Aggregate scores thus decompose into interpretable capability coordinates.
3. **Controlled out-of-distribution evaluation via asset partitioning.** Beyond isolated train, validation, and test splits, EBench evaluates four axes: unseen backgrounds, unseen objects, paraphrased instructions, and their mixture. Train and test sets are isolated at the asset level.

We evaluate four recent VLAs on EBench: π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), XVLA (Zheng et al., 2025), and InternVLA-A1 (Cai et al., 2026). Their aggregate test success rates lie within a narrow band of 24.4–29.5%, yet the five-dimensional capability profiles diverge by tens of points. $\pi_{0.5}$ attains the highest test SR of 29.5% and the smallest train–test gap, with a retention ratio

of 0.92. InternVLA-A1 dominates mobile manipulation but has the biggest gap of 29 points between mobile and dexterous fixed-base tasks. Per-atomic-skill rankings are disjoint across models, so no single policy covers the capability space. We analyzes the generalization ability from 4 representative perspectives, identifying the impact of different distribution shift factors. The results reveal strengths and weaknesses of models behind an overall score. We hope this benchmark offers a broad set of diagnostic signals to guide iteration on generalist manipulation models.

2. Related Work

Simulation benchmarks for manipulation. Eval suites of tabletop tasks such as RLBench (James et al., 2020), CALVIN (Mees et al., 2022), and LIBERO (Liu et al., 2023) pioneered standardized evaluation but cover a narrow regime of short-horizon, fixed-base pick-and-place. Mobile and multi-scene suites such as Habitat (Savva et al., 2019), SAPIEN (Xiang et al., 2020), ManiSkill (Mu et al., 2021), and RoboCasa (Nasiriany et al., 2024) broaden scene diversity but seldom include sub-centimetre dexterous behaviors. Procedurally generated suites such as RoboTwin (Chen et al., 2025; Mu et al., 2025) and GenManip (Gao et al., 2025) scale task counts but still report a per-task scalar success rate without a structured taxonomy. Real-to-sim transfer benchmarks such as SimplerEnv (Li et al., 2024) mirror a fixed set of real-robot tabletop tasks in simulation, optimising for fidelity to one embodiment rather than coverage across regimes. Targeted diagnostic benchmarks such as RMBench (Chen et al., 2026) probe a single capability axis in isolation. EBench differs from this prior work on two axes: it hosts long-horizon, dexterous, and mobile manipulation under one evaluation protocol, and it pairs every task with a 5 capability axes and 4 generalize dimensions, so that aggregate scores decompose into interpretable coordinates rather than collapsing into a leaderboard scalar.

Vision–language–action models. π_0 (Black et al., 2024) and its successor $\pi_{0.5}$ (Intelligence et al., 2025) use flow-matching action heads on top of large multi-robot pre-training mixtures. XVLA (Zheng et al., 2025) decouples vision–language understanding from action execution via a modular decoder. InternVLA-A1 (Cai et al., 2026) pairs strong visual representations with a hierarchical planner. Adjacent generalist policies include OpenVLA (Kim et al., 2024), GR00T-N1 (Bjorck et al., 2025), RDT (Liu et al., 2025), Octo (Team et al., 2024), and Diffusion Policy (Chi et al., 2025); many of these are pre-trained on cross-embodiment datasets such as Open X-Embodiment (O'Neill et al., 2024). Recent open codebases such as StarVLA (Community, 2026) and StarVLA-α (Ye et al., 2026) explore lighter-weight and more modular VLA recipes. These models are typically compared on hardware-specific real-robot runs or on narrow simulation subsets; EBench provides a multi-dimensional comparison of recent VLAs under a matched generalist protocol.

3. The EBench Benchmark

3.1. Task Design and Diversity

EBench comprises 26 manipulation tasks organised into three families: **Mobile Pick-and-Place** (10 mobile tasks, 600–1,000 simulation steps per episode), **Mobile Long-Horizon** (9 multi-stage mobile sequences, 3,000–5,000 steps), and **Table-Top Dexterous-and-Precise** (7 fixed-base tasks, 1,500–3,500 steps, covering sub-centimetre insertion, alignment, and bimanual coordination). Physics is simulated at 60 Hz, so the longest task corresponds to roughly 83 seconds of robot operation. Each task is annotated along five dimensions: scene, atomic skill, temporal horizon, precision, and operating mode. Table 1 summarises the categories within each dimension. The taxonomy supports interpretable queries such as "how does model X perform on high-precision, long-horizon mobile

Figure 2. **EBench end-to-end pipeline.** *Left*: 26 tasks span pick-and-place, long-horizon, and dexterous-and-precise families, instantiated on shared scene and robot assets. *Middle, two-track synthesis*: dexterous-and-precise demonstrations are collected via human **teleoperation** (top); mobile and long-horizon trajectories are generated by **motion planning** from key-frame end-effector poses fed to cuRobo (bottom). *Right*: EBench is evaluated through a client–server protocol. The IsaacSim-backed server returns observations and a step signal, and VLA or WAM clients (e.g. a VLM with a DiT action head) emit actions in response.

tasks?" and prevents covering up of weaknesses by strong performance on easy majority categories.

Table 1. EBench five axes task taxonomy. Numbers in parentheses indicate the number of categories within each dimension.

Axes	Categories
Scene (9)	Bedroom, Bathroom, Kitchen, Living Room, Study, Dining Room, Supermarket, Industrial, Logistics
Atomic Skill (11)	Grasp, Place, Push, Pull, Press, Insert, Pour, Flip, Sweep, Handover
Range (2)	Short Horizon ($< 2,000$ steps), Long Horizon ($\geq 2,000$ steps)
Precision (3)	Low ($\geq$ 10 cm), Medium ($<$ 10 cm, and $\geq$ 1 cm), High ($<$ 1cm)
Operating Mode (2)	Mobile, Fixed but Dexterous

All tasks share a unified action space for a dual-arm robot mounted on a mobile base: each arm can be commanded in either 6-DoF joint position or 6-DoF end-effector pose, paired with a per-arm gripper width, and the base accepts a 3-D velocity command (planar x, y, and yaw rate), which the model is free to emit on every task. A single model checkpoint can therefore be evaluated across regimes without any architectural modification.

3.2. Data Synthesis: Teleoperation and Motion Planning

To incorporate such diverse behaviors, the collection of post-training demonstration faces the following challenges: (1) Dexterous-and-precise tasks require complex interactions that motion planner can hardly program. (2) Collecting a successful Long-horizon demonstration is extremely hard through human teleoperation due to the exponentially amplified failure probability in the long sequence. (3) Mobile manipulation is hard to teleoperate since a single operator has to coordinate base motion and arm motion through the same controller, and small base disturbances destabilize the arm reference

frame. To solve the challenges, EBench couples two complementary collecting streams, as shown in Figure 2:

1. **Teleoperation for dexterous-and-precise tasks.** The 7 dexterous tasks are collected through a kinematically isomorphic actor-follower setup. This preserves the reactive feedback and dynamic adjustments needed for contact-rich micro-corrections such as peg-in-hole insertion, nut tightening, and gear meshing.
2. **Key-frame pose and cuRobo for mobile and long-horizon tasks.** For the remaining 19 tasks, where teleoperation is either too expensive (long-horizon) or too awkward to control (mobile), the annotator instead specifies a sparse sequence of key-frame end-effector poses, together with base waypoints for mobile cases. cuRobo (Sundaralingam et al., 2023) then solves a collision-free, minimum-jerk trajectory that connects them. This stream produces thousands of episodes per task without sacrificing kinematic feasibility, and the resulting trajectories are immediately re-rendered under randomized backgrounds, objects, and lighting to produce the generalization variants (§3.3).

The post-training dataset contains 91.4 hours demonstrations, 6,600 episodes ultimately, organized in LeRobot format. Each dexterous-and-precise task contributes 400 teleoperated episodes, each mobile pick-and-place task contributes 200 motion-planned episodes, and each long-horizon task contributes 200 motion-planned episodes.

3.3. Generalization Dimensions

EBench controls 4 generalization dimensions in evaluation: (1) **Background** replaces scene textures and lighting with unseen variants while objects and instructions are held fixed. (2) **Object** swaps each manipulated entity for a geometrically distinct unseen instance within the same category. (3) **Instruction** paraphrases each natural-language command while preserving its operational goal. (4) **Mix** applies background, object, and instruction perturbations jointly. Background and instruction probe perceptual and linguistic robustness without changing the underlying physics, object swaps require physical generalisation, and Mix compounds the two. Train and test sets share the same synthesis pipeline and the same randomisation axes; the training set is itself drawn under the full background, object, and instruction randomisation. Train/test isolation is enforced *at the asset level*: scene textures, object instances, and instruction paraphrases used for training are drawn from a pool that is disjoint from the test pool.

3.4. Evaluation Protocol and Metric Library

Since EBench is born to evaluate generalist policies, the primary protocol requires **one model checkpoint to solve all 26 tasks**. The validation set is split into VAL-TRAIN, with 130 in-distribution episodes (5 per task across 26 tasks), and VAL-UNSEEN, with 154 episodes containing object swaps drawn from the unseen asset pool. The TEST split comprises 510 episodes spanning all four generalisation dimensions (20 per task for 24 tasks; 15 per task for two long-horizon tasks). It is released publicly together with the rest of the benchmark, but is constructed from a pool of scene, object, and instruction assets that is disjoint from the training pool.

Two metrics are adopted: a binary success signal **SR (Primary)** and a task **Score** that rewards partial progress through the task. The score is computed stage by stage rather than from a single distance function. Each task is declared as an ordered sequence of stages, every stage holds a set of conditions over simulator state, and the score advances whenever the next stage in sequence

Table 2. **Overall performance of baselines across evaluation splits. SR** (%) as the primary metric is highlighted in **bold**; **Score** denotes continuous task progress (%), and **Retention** is the ratio of TEST to VAL-TRAIN. Results are reported as mean ± standard deviation over three evaluation runs, with the best value in each column shown in bold.

Model	VAL-TRAIN		VAL-UNSEEN		TEST		Retention	
	SR	Score	**SR**	Score	**SR**	Score	**SR**	Score
π_0 Black et al. (2024)	30.5±1.8	42.9±1.5	25.4±2.8	39.3±2.3	24.4±0.9	38.4±0.6	0.80	0.89
XVLA Zheng et al. (2025)	28.3±2.5	42.1±1.3	22.7±4.3	35.9±3.7	24.7±1.1	37.5±0.9	0.87	0.89
InternVLA-A1 Cai et al. (2026)	**33.1**±2.0	44.2±1.8	20.8±1.1	33.8±0.8	27.6±1.6	40.2±2.0	0.83	0.91
$\pi_{0.5}$ Intelligence et al. (2025)	32.1±5.1	**48.1**±5.6	**26.5**±2.0	**42.9**±0.7	**29.5**±0.3	**45.6**±0.2	**0.92**	**0.95**

is satisfied; SR fires only when the final stage is reached. Stage conditions are represented by a shared schema, namely **evaluation primitives**. They are generated directly from simulator state: scene-graph relations between objects (e.g. "cup on tray"), articulation joint angles for doors, drawers, and tools, object tilt and orientation, and end-effector or base pose tolerances. Composing these primitives into an ordered stage graph replaces the per-task hand-coded evaluators used in prior benchmarks and makes scores directly comparable across tasks within a family.

4. Experimental Setup

Evaluated Models. We evaluate 4 recent vision–language–action (VLA) models that span distinct architecture and pre-training mixtures: $\boldsymbol{\pi_0}$ Black et al. (2024), $\boldsymbol{\pi_{0.5}}$ Intelligence et al. (2025), **XVLA** Zheng et al. (2025), and **InternVLA-A1** Cai et al. (2026). All models are fine-tuned from pretrained checkpoints on the same EBench training data using a consistent recipe: 200K gradient steps, batch size 128, AdamW optimizer, and a cosine learning-rate scheduler with warm-up, where the peak lr is $1e-5$.

Post-Training and Evaluation Protocol. Post-training uses all 6,600 demonstration episodes described in §3.2, with teleoperated and motion-planned trajectories roughly balanced at the frame level. Observations consist of RGB images at 224×224 from three viewpoints: left, right, and topdown views, together with proprioceptive state and a natural-language instruction. Each frame is recorded at the simulation rate (60 Hz physics step), and the policy is queried at the same rate. Because the IsaacSim renderer is non-deterministic, each model is evaluated three times and we report the mean and standard deviation across runs.

5. Capability Profiling

5.1. Overall Performance

Table 2 shows that the four models achieve similar test SRs within a narrow five-point range (24.4–29.5%), yet exhibit markedly different in-distribution behaviors. $\pi_{0.5}$ achieves the highest test SR (29.5%) and the strongest retention (SR: 0.92, Score: 0.95), indicating that its validation performance is the most reliable predictor of held-out capability. In contrast, although InternVLA-A1 attains the highest VAL-TRAIN SR (33.1%), its performance drops sharply on VAL-UNSEEN (20.8% SR) and yields relatively weak retention on the held-out test split (0.83 SR retention), suggesting strong in-distribution fitting but limited robustness to distribution shifts. Similarly, although π_0 achieves

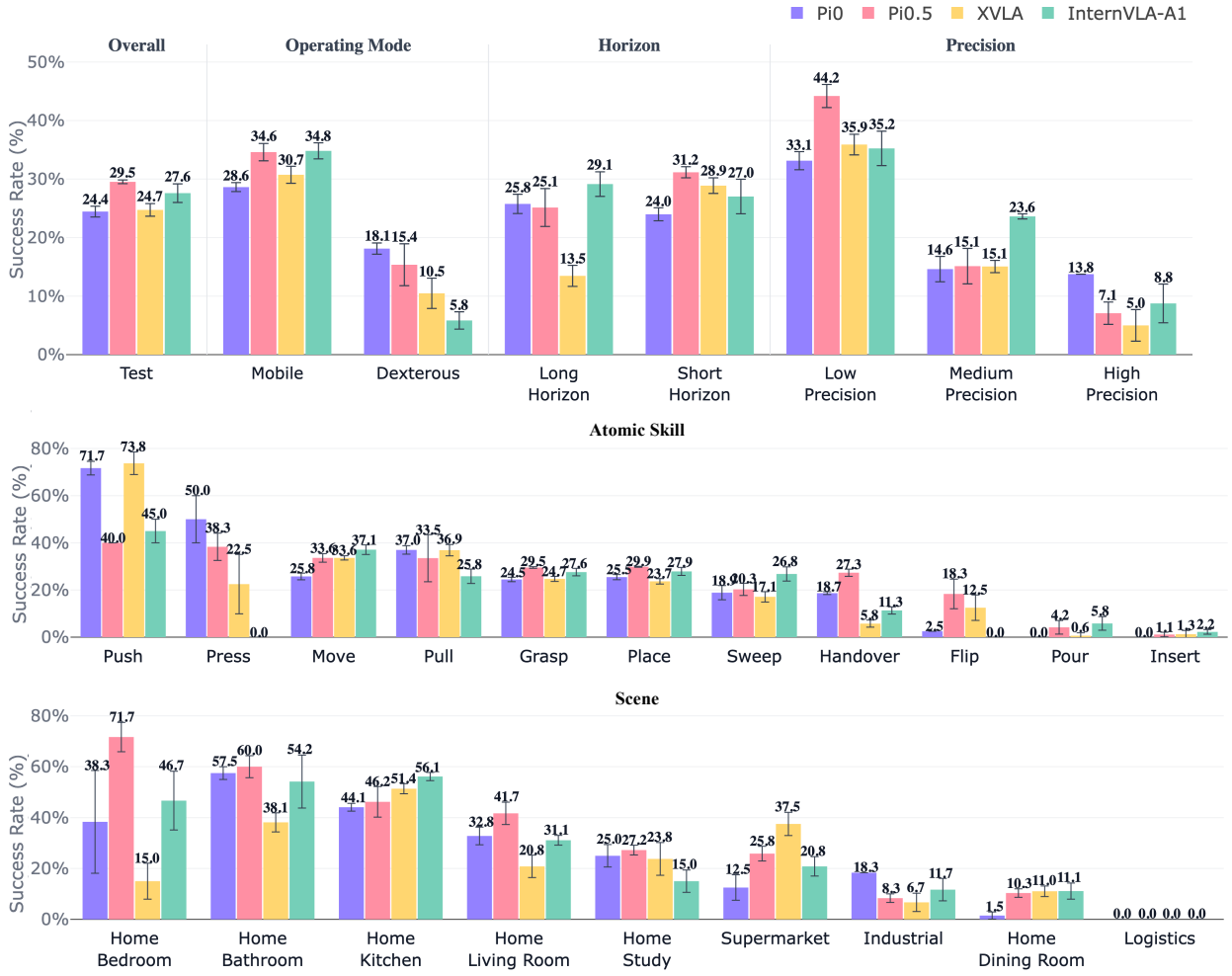

Figure 3. **Capability breakdown on the five axes.** The top row reports overall success rate and three task-level axes: operating mode, temporal horizon, and precision tolerance. The middle row breaks performance down by atomic skill, while the bottom row reports performance across scene categories. Bars denote the mean test SR, and error bars denote standard deviation across seeds.

a comparable test SR (24.4%), it exhibits the lowest retention ratio (0.80), indicating stronger overfitting to the training distribution.

5.2. Five-Dimensional Capability Breakdown

Figure 3 decomposes test SR along five complementary axes. The top row summarizes overall performance and three low-cardinality factors, namely operating mode, temporal horizon, and precision. The middle row breaks down performance by atomic skill, and the bottom row reports scene-wise success rates. While the models have relatively close aggregate SRs, their capability profiles differ markedly across these dimensions.

Operating mode. InternVLA-A1 performs competitively on mobile manipulation, achieving a test SR comparable to $\pi_{0.5}$ (both around 34.7%), but its performance drops sharply on dexterous fixed-base tasks (5.8% SR). This results in the largest mobile-to-dexterous gap in the cohort (29

points), suggesting that the model handles navigation-scale decision making effectively but lacks the fine-grained contact control required for dexterous manipulation. In contrast, π_0 exhibits the most balanced performance profile, with a smaller 11-point gap between mobile (29.2%) and dexterous (18.1%) settings, albeit at lower absolute performance than $\pi_{0.5}$.

Precision and horizon. On sub-centimetre high-precision tasks, π_0 leads at 13.8% SR, while all other models fall to single-digit SR. On low-precision tasks, $\pi_{0.5}$ achieves the best performance with 44.2% SR, and other models cluster around 35% SR. Task horizon reveals a different pattern. Short-horizon tasks are consistently easier, with all models achieving SRs in the 24–32% range. Long-horizon tasks expose substantially larger performance gaps: InternVLA-A1 achieves the highest SR (29.1%), whereas XVLA drops sharply from 28.9% on short tasks to 13.5% on long-horizon settings, suggesting weaker temporal credit assignment in its modular decoder architecture.

Atomic skills and scenes. No single model dominates all eleven atomic skills. π_0 leads on Pull at 47% and on Press at 50%. XVLA dominates Push at 73.8% but bottoms out on Handover at 5.8%. InternVLA-A1 wins on Move and Sweep yet scores 0% on Press and Flip. In contrast, $\pi_{0.5}$ is the only model with no catastrophic-zero categories. Scene-level rankings exhibit similarly heterogeneous patterns. $\pi_{0.5}$ performs best in Bedroom, Bathroom, and Living Room, InternVLA-A1 leads in Kitchen and Dining scenes, and XVLA achieves the highest SR in Supermarket settings.

6. Generalization Diagnosis

Aggregate SR at a single checkpoint provide only a static view of performance. To examine how generalization evolves during post-training, we evaluate each model at 25k, 50k, 100k, and 200k steps, and plot Validation-Train and Test SR in Figure 4.

Fit–generalization dynamics. The vertical gap between the dashed and solid curves in Figure 4 measures the empirical fit–generalization gap: smaller gaps indicate better transfer of in-distribution gains to held-out rollouts. Overall, additional post-training improves Test SR for all models by 200k steps, but the transfer from Validation-Train to Test is model-dependent. Specifically, $\pi_{0.5}$ shows the most stable dynamics, with the two curves rising largely together and the highest final Test SR. π_0 also improves steadily, though its late-stage gap becomes more visible, indicating incomplete transfer of additional fit. XVLA is more sensitive to training duration, showing a non-monotonic Test trajectory before recovering at 200k. InternVLA-A1 achieves the strongest final VAL-TRAIN SR but retains a larger Test gap, suggesting that its additional fitting benefits the training distribution more than OOD rollouts.

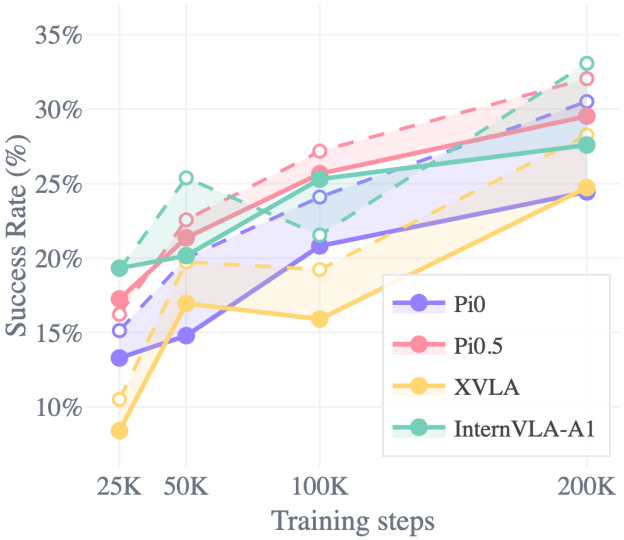

Figure 4. **SR** of baselines on VALIDATION-TRAIN and TEST split across different training steps. Dashed and solid lines denote Train and Test results, respectively.

Generalization across axes. Figure 5 decomposes test SR across four generalization dimensions: Background, Instruction, Object, and Mix, corresponding to unseen background, paraphrased instruction, unseen object instance, and their joint perturbation, respectively. The difficulty hierarchy is clear: background and linguistic perturbations are relatively mild, object-level physical changes are harder, and their combination is the most challenging. All four models maintain 27–35% SR under Background and Instruction perturbations, suggesting that their perceptual and language grounding remain relatively robust when object physics is unchanged. In contrast, Object swaps reduce SR to 21–29%, indicating that physical changes in the form of new object geometry and mass distribution pose a stronger generalization challenge. The joint Mix setting further lowers SR to 18–23%, showing that compositional distribution shifts amplify failure modes beyond any single perturbation. Overall, $\pi_{0.5}$ is the most robust baseline, leading on Background, Object, and Mix, while InternVLA-A1 achieves the best Instruction generalization.

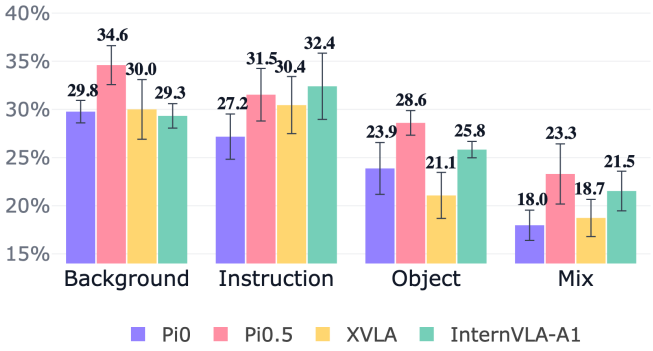

Figure 5. Test SR across four generalization dimensions.

7. Pretraining Sensitivity Across Benchmarks

A central question for evaluating generalist manipulation policies is: **Can the benchmark capture the effect of large-scale pretraining on policy performance?** We address this question by comparing five representative architectures – π_0 (Black et al., 2024), $\pi_{0.5}$ (Intelligence et al., 2025), XVLA (Zheng et al., 2025), Fast-WAM (Yuan et al., 2026), and StarVLA-OFT (Ye et al., 2026) – under two training regimes on three benchmarks: EBench, LIBERO (Fei et al., 2025; Liu et al., 2023), and RoboTwin 2.0 (Chen et al., 2025). In the **pretrained** regime, we evaluate the released checkpoint fine-tuned on the benchmark's training split. In the **from-scratch** regime, we initialize the same architecture randomly and train it only on the benchmark's training split. Fast-WAM and StarVLA-OFT have no released pretrained checkpoint and therefore appear only in the from-scratch regime.

Specifically, we additionally train π_0, $\pi_{0.5}$, XVLA, Fast-WAM, and StarVLA-OFT from scratch on EBench. LIBERO scores are average success rates over the four official suites Spatial, Object, Goal, and Long; RoboTwin 2.0 scores are average success rates over the hard-task split. Pretrained π_0, $\pi_{0.5}$, and XVLA values on LIBERO are taken from the respective model release papers; pretrained values on RoboTwin 2.0 Hard are taken from LingBot-VA (Li et al., 2026) for π_0 and $\pi_{0.5}$ and from MOTUS (Bi et al., 2026) for XVLA. The from-scratch entries for $\pi_{0.5}$ and XVLA on LIBERO and RoboTwin 2.0 are reported as '–' because these architectures are not trained from scratch on these benchmarks in published work, and we do not run such ablations ourselves; the Fast-WAM row supplies the without-pretrain data point on both benchmarks.

Results. On EBench, pretraining helps every architecture by a large margin: π_0 goes from 11.2 to 24.4% SR, $\pi_{0.5}$ from 8.5 to 29.5%, and XVLA from 15.7 to 24.7%. On LIBERO, pretraining makes essentially no difference: all five entries score between 94 and 98%, and from-scratch π_0 scores 95.7, slightly above pretrained π_0 at 94.1. On RoboTwin 2.0 Hard, both without-pretrain entries score above every pretrained baseline: Fast-WAM reaches 91.8 and π_0 reaches 88.8, while the three pretrained baselines span 58.4 to 76.8. Disentangling the contribution of pretraining requires a benchmark whose

Table 3. Pretraining ablation across EBench, LIBERO, and RoboTwin 2.0. The *Pretrain* column indicates whether each row evaluates the released checkpoint or a model trained from random initialization on the benchmark's training split. † The from-scratch π_0 entries on LIBERO and RoboTwin 2.0 Hard are taken from the StarVLA-π configuration reported in Community (2026) (Qwen3-VL-4B VLM backbone), which is architecturally equivalent to π_0 and is trained from random initialization on each benchmark's training split.

Model	Pretrain	EBench-Test(SR)	EBench-Test(Score)	LIBERO-Avg	RoboTwin 2.0-Hard
XVLA (Zheng et al., 2025)	×	15.7	27.7	–	–
π_0 (Black et al., 2024)	×	11.2	19.9	95.7†	88.8†
$\pi_{0.5}$ (Intelligence et al., 2025)	×	8.5	14.9	–	–
Fast-WAM (Yuan et al., 2026)	×	4.7	7.6	97.6	91.8
StarVLA-OFT (Ye et al., 2026)	×	0	0.2	98.8	88.3
XVLA (Zheng et al., 2025)	✓	24.7 **(+9.0)**	38.7 **(+11.0)**	98.1	72.8
π_0 (Black et al., 2024)	✓	24.4 **(+13.2)**	38.4 **(+18.5)**	94.1	58.4
$\pi_{0.5}$ (Intelligence et al., 2025)	✓	29.5 **(+21.0)**	45.6 **(+30.7)**	96.9	76.8

pretrained and from-scratch baselines do not coincide. LIBERO and RoboTwin 2.0 are not designed to evaluate this factor for generalist policies: both are largely saturated, so from-scratch models already reach 94–98% on LIBERO and match or exceed every pretrained baseline on RoboTwin 2.0 Hard, leaving essentially no gap that pretraining could account for. EBench instead recognizes the improvement brought by pretraining, exhibiting a large and consistent pretrained–from-scratch gap of 9–21 SR points, and is therefore well suited to measuring the effect of large-scale pretraining for generalist policies.

8. Limitations

EBench operates entirely in simulation, and we do not claim that simulation scores predict real-robot performance. However, we would like to treat EBench as a reproducible screening substrate that precedes physical evaluation rather than replacing it. We will also study the correlation between sim and real evaluation based on EBench tasks in future work. The 26-task suite covers 9 scene categories sparsely, so scene-level rankings should be read as preliminary; expanding toward 50 or more tasks is on our roadmap and will unlock multi-way regression in place of permutation tests.

9. Conclusion

We presented EBench, a simulation benchmark for generalist embodied manipulation that places long-horizon, dexterous-and-precise, and mobile manipulation under a single evaluation protocol, a combination that current public benchmarks individually approximate but jointly avoid. EBench pairs 26 capability-tagged tasks with four controlled generalization dimensions, drawn from a test asset pool that is disjoint from the training pool, and supplies them with a 91.4 hours dataset synthesised through two complementary tracks: teleoperation for dexterous-and-precise tasks, and key-frame poses with cuRobo for mobile and long-horizon tasks. Applied to π_0, $\pi_{0.5}$, XVLA, and InternVLA-A1, EBench shows that VLAs which look identical at the scalar-SR level differ by tens of points along interpretable axes, namely operating mode, precision, horizon, and atomic skill, and follow distinct fit–generalise trajectories as training proceeds. $\pi_{0.5}$ leads aggregate test performance, while InternVLA-A1, π_0, and XVLA each dominate disjoint subsets of the capability space, and the field has not converged on a single inductive bias. Beyond capability profiling, a controlled pretraining study (Section 7) shows that, among EBench, LIBERO, and RoboTwin 2.0, EBench is the only benchmark whose from-scratch

and pretrained baselines do not coincide: large-scale pretraining lifts every architecture by 9–21 SR points on EBench (e.g. $\pi_{0.5}$ from 8.5 to 29.5% and π_0 from 11.2 to 24.4%), whereas on LIBERO and RoboTwin 2.0 from-scratch models match or surpass their pretrained counterparts, so EBench is uniquely able to surface the contribution of pretraining. The tasks, synthesised dataset, evaluation code, and all splits are publicly released so that future generalist policies can be diagnosed along the same five capability axes.

References

H. Bi, H. Tan, S. Xie, Z. Wang, S. Huang, H. Liu, R. Zhao, Y. Feng, C. Xiang, Y. Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 35101–35113, 2026.

J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

J. Cai, Z. Cai, J. Cao, Y. Chen, Z. He, L. Jiang, H. Li, H. Li, Y. Li, Y. Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

T. Chen, Y. Wang, M. Li, Y. Qin, H. Shi, Z. Li, Y. Hu, Y. Zhang, K. Wang, Y. Chen, et al. Rmbench: Memory-dependent robotic manipulation benchmark with insights into policy design. *arXiv preprint arXiv:2603.01229*, 2026.

C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44 (10-11):1684–1704, 2025.

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

M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023.

S. Fei, S. Wang, J. Shi, Z. Dai, J. Cai, P. Qian, L. Ji, X. He, S. Zhang, Z. Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

N. Gao, Y. Chen, S. Yang, X. Chen, Y. Tian, H. Li, H. Huang, H. Wang, T. Wang, and J. Pang. Genmanip: Llm-driven simulation for generalizable instruction-following manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12187–12198, 2025.

P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023.

L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pages 29982–30009, 2025.

O. Mees, L. Hermann, E. Rosete-Beas, and W. 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.

T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. *arXiv preprint arXiv:2107.14483*, 2021.

Y. Mu, T. Chen, Z. Chen, S. Peng, Z. Lan, Z. Gao, Z. Liang, Q. Yu, Y. Zou, M. Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the computer vision and pattern recognition conference*, pages 27649–27660, 2025.

S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119. IEEE, 2023.

O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

H. Wang, J. Chen, W. Huang, Q. Ben, T. Wang, B. Mi, T. Huang, S. Zhao, Y. Chen, S. Yang, et al. Grutopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024.

F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020.

J. Ye, N. Gao, S. Yang, J. Zheng, Z. Wang, Y. Chen, P. Chen, Y. Chen, S. Liu, and J. Jia. StarVLA-α: Reducing Complexity in Vision-Language-Action Systems. *arXiv preprint arXiv:2604.11757*, 2026.

T. Yuan, Z. Dong, Y. Liu, and H. Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

J. Zheng, J. Li, Z. Wang, D. Liu, X. Kang, Y. Feng, Y. Zheng, J. Zou, Y. Chen, J. Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

W. Zhong, P. Cao, Y. Jin, L. Li, W. Cai, J. Lin, H. Wang, Z. Lyu, T. Wang, X. XU, et al. Internscenes: A large-scale simulatable indoor scene dataset with realistic layouts. *Advances in Neural Information Processing Systems*, 38, 2026.

A. Implementation Details

Baselines. To make cross-model comparisons fair, every baseline in this paper is trained and rolled out under a common training budget and action-chunk schedule, regardless of architecture. We use a global batch size of 128, a relative action-chunk prediction horizon of 50 timesteps, and an open-loop application horizon of 30 steps – the policy predicts a 50-step chunk at every replanning step, but only the first 30 actions are executed in the environment before the next prediction is issued. The resulting 20-step lookahead buffer absorbs the model's inference latency without introducing closed-loop instability. All other architecture-specific hyperparameters (optimizer, learning-rate schedule, dropout, action normalisation, etc.) follow the official open-source repository of each model unchanged.

Metrics. Each episode reports a binary success signal, SR, and a continuous task score that rewards partial progress such as correctly placing 2 of 3 objects. We report both metrics because the score captures "near misses" that binary success discards, particularly on long-horizon tasks where individual sub-goals may be partially satisfied.

Camera configurations. EBench supports two primary camera configurations for studying visual perspective effects: **Headview:** An egocentric camera mounted on the robot head, providing a local, first-person perspective aligned with the end-effector gaze. **Overview:** A bird's-eye camera positioned above the scene, providing a global, top-down perspective of the workspace and surrounding environment. Both configurations include left and right auxiliary cameras for stereo cues.

Computational cost. A full validation run takes approximately 30 minutes on eight RTX 4090 GPUs using the distributed evaluation toolkit. The complete benchmark, validation plus test, completes in under two hours on the same hardware, enabling rapid iterative development.

B. Camera-Perspective Sensitivity

EBench supports systematic comparison across alternative camera configurations for the primary input stream, while keeping the left/right auxiliary cameras fixed and the rest of the training and evaluation protocol identical. At present the public leaderboard exposes two such configurations for the π-family policies trained on the EBench training split: the **Overview** stream, a wide-angle bird's-eye camera positioned above the workspace, and the **Headview** stream, a tighter top-down camera with a smaller field of view that emphasises the end-effector workspace. For both π_0 and $\pi_{0.5}$, three independent 200k seeds are available under each configuration.

B.1. Overall Perspective Effect

Table 4 reports Test SR averaged over the three seeds of each (model, perspective) cell.

Table 4. Test SR (%) by camera perspective for the π-family policies at 200k steps, averaged over three seeds per cell. Δ = Headview – Overview.

Model	Overview	Headview	Δ (%)
π_0	24.44	26.92	+2.48
$\pi_{0.5}$	29.53	25.32	−4.21

The point estimates in Table 4 are opposite in sign: π_0 gains +2.48% of Test SR when the primary stream is switched from Overview to Headview (24.44 → 26.92), while $\pi_{0.5}$ loses −4.21% under the same switch (29.53 → 25.32). The two models share the openpi backbone and the same EBench training data, so the action heads, the only architectural component that differs between them, are a natural candidate for the locus of the sensitivity.

B.2. Decomposition by Operating Mode and Horizon

To localise the perspective effect, we decompose the Headview-minus-Overview Δ on the Test split by the operating-mode tag Mobile vs. Dexterous fixed-base and the horizon tag Long vs. Short from the main paper's task taxonomy, using the same set of three seeds per cell.

Table 5. Headview-minus-Overview Δ on Test SR (%), decomposed by operating mode – Mobile vs. Dexterous fixed-base – and by horizon – Long vs. Short. Positive Δ favours Headview; negative Δ favours Overview.

	Operating mode		Horizon	
Model	Mobile	Dexterous	Long	Short
π_0	+0.28	+8.38	−0.62	+3.60
$\pi_{0.5}$	−6.21	+1.19	−5.90	−3.60

For π_0, the modest +2.48% overall preference for Headview is concentrated in the dexterous fixed-base subset at +8.38% and is essentially zero on mobile tasks at +0.28%; the horizon decomposition tells the same story, with +3.60% on short-horizon tasks where most dexterous fixed-base tasks live and −0.62% on long-horizon tasks where the mobile-manipulation tasks live. For $\pi_{0.5}$, the −4.21% overall preference for Overview is concentrated in the mobile subset at −6.21% and in the long-horizon subset at −5.90%, with smaller but consistent negative deltas on the other two strata. The pattern is

therefore not a uniform global shift but a stratum-specific effect: π_0 benefits from Headview where the workspace is small and the camera frames the end-effector tightly, while $\pi_{0.5}$ benefits from Overview where the workspace is large and the camera covers the full mobile platform's reach.

B.3. Discussion

Two observations follow from Tables 4 and 5. First, the perspective sensitivity is real but small in magnitude compared to the cross-model differences reported in the main paper: the largest stratified $|\Delta|$ in Table 5 is 8.38%, an order of magnitude smaller than the $\sim$ 30% pretraining gap reported in Section 7. For the question *"which camera should I serve at deployment time?"* the choice is therefore architecture-dependent rather than universally dictated by the task family. Second, the stratum-localised pattern – π_0's Headview gain concentrated in the dexterous and short-horizon subsets, $\pi_{0.5}$'s Overview gain concentrated in the mobile and long-horizon subsets – is consistent with a per-task selection rule: a workspace whose extent matches the camera's field of view tends to be preferred, and the optimal field of view is itself a function of the action head's effective receptive field. Larger-scale follow-up that adds an explicit egocentric camera and a per-task camera-fusion ablation would be needed to test this rule beyond the two perspectives currently available, and is left to future work.

C. Capability Profiling Summary and Task-Level Analysis

C.1. Task-Level Complementarity and Hard Tasks

Figure 6 renders the full per-task picture: rows are the 26 tasks (sorted top-to-bottom by total Test SR), columns are the four baselines crossed with the three splits (VALIDATION-TRAIN, VALIDATION-UNSEEN, TEST), and the color encodes per-task success rate. Two structural phenomena pop out immediately from this view.

Per-task complementarity. The π-family π_0 and $\pi_{0.5}$, taken together, and XVLA exhibit strong complementarity that the cluster bars in the main paper smooth over. Tasks where the π-family outperforms XVLA by the largest Test-SR margin include `detergent` at +71%, `perfume_to_cosmetics_rack` at +35%, and `remote_to_holder` at +19%; these read as a vertical pair of dark cells in the π-family blocks above pale cells in the XVLA block. Conversely, XVLA wins on `apple_to_fruit_bowl` at +51%, `soap_to_dish` at +44%, and `utensils_to_holder` at +38%, which produce the inverse stripe – pale π-family cells against a dark XVLA column. InternVLA-A1 sits between the two families on most rows but resolves several of the π-family weaknesses (notably `apple_to_fruit_bowl` and `soap_to_dish`) at the cost of being weaker on the dexterous tabletop tail.

Universally-hard tasks. At the bottom of the heatmap, five rows stay near-white across every column and every split: `shop`, `bottle`, `peg_in_hole`, `collect_coffee_beans`, and `flip_cup_collect_cooki`. All four baselines score $\leq$ 5% SR on these tasks across multiple evaluation snapshots, indicating that they lie beyond the current frontier of generalist policy capability. `peg_in_hole` is a classic high-precision insertion task, and `flip_cup_collect_cookies` requires coordinated flipping and collection; both demand force-aware feedback loops that current open-loop action models lack. We therefore propose these five tasks as a small "hard suite" that future generalist papers can use as a low-floor stress test: any model that crosses 10% SR on the full set is moving the frontier, while the cluster aggregates over the full 26 tasks remain dominated by the easier majority.

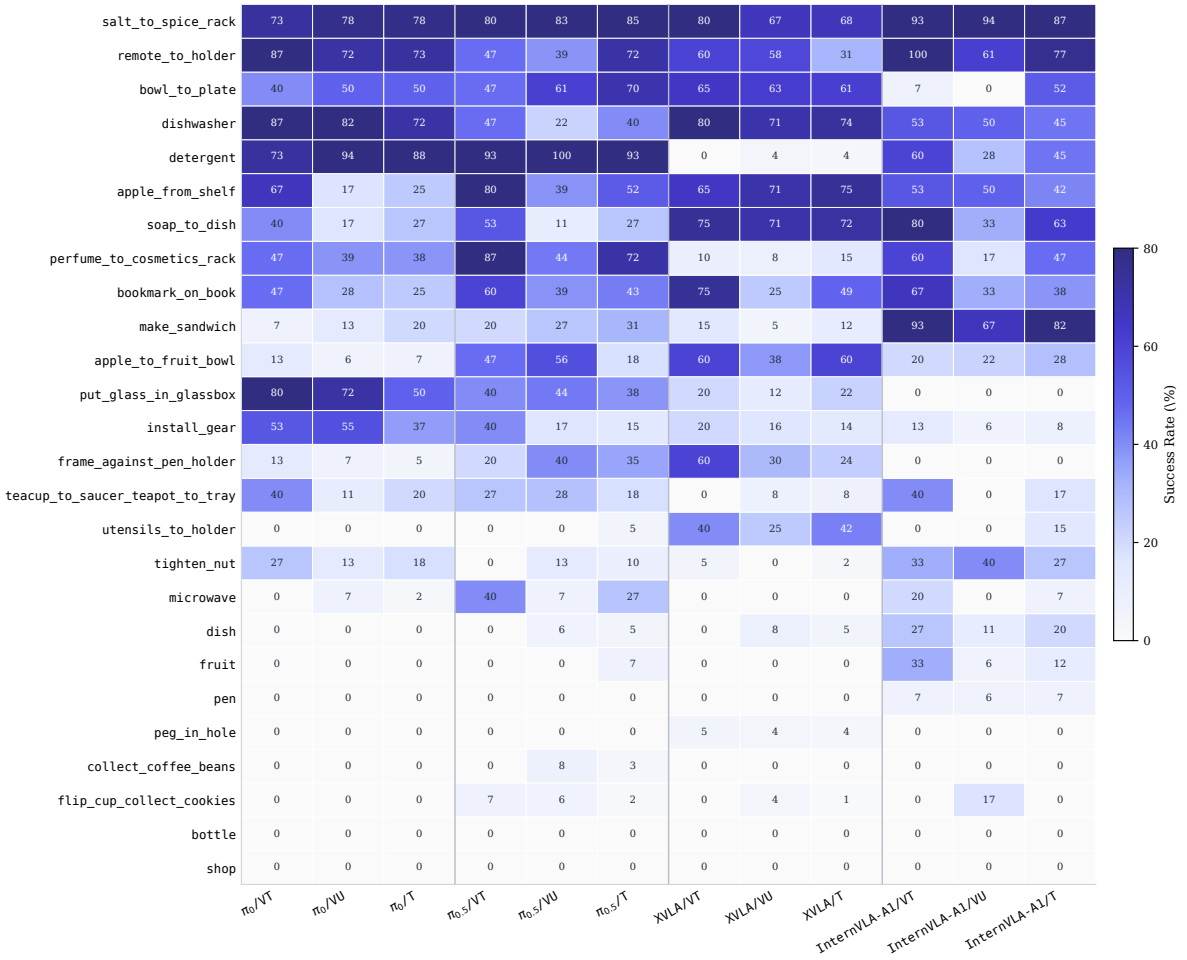

Figure 6. Per-task success-rate heatmap across the four baselines and three splits. VT: VALIDATION-TRAIN, VU: VALIDATION-UNSEEN, T: TEST.

D. Training Loss

For reproducibility, Figure 7 plots the full training loss trajectory of each baseline from initialization through the 200k checkpoint, parsed directly from each model's trainer log. In all cases the loss is the value computed and reported by the model's *official open-source repository*: π_0 and $\pi_{0.5}$ use the flow-matching action loss emitted by openpi, XVLA uses the diffusion-policy denoising loss emitted by X-VLA, and InternVLA-A1 uses the combined action plus auxiliary generative loss emitted by OT-Train. Because each repository defines the loss in its own units, with different normalisations, action chunk sizes, and auxiliary-term weightings, the absolute magnitudes are not directly comparable across models. We therefore display the four curves on a single logarithmic vertical axis to preserve the within-model convergence shape across the four orders of magnitude the runs span between initialization and the 200k tail. π_0 and $\pi_{0.5}$ already log a loss value averaged over every 100 optimizer steps in their respective trainers; we apply the same 100-step block averaging to XVLA's higher-frequency log so all three curves share the same reporting cadence. InternVLA-A1's log records once every 200 steps and is plotted as-is.

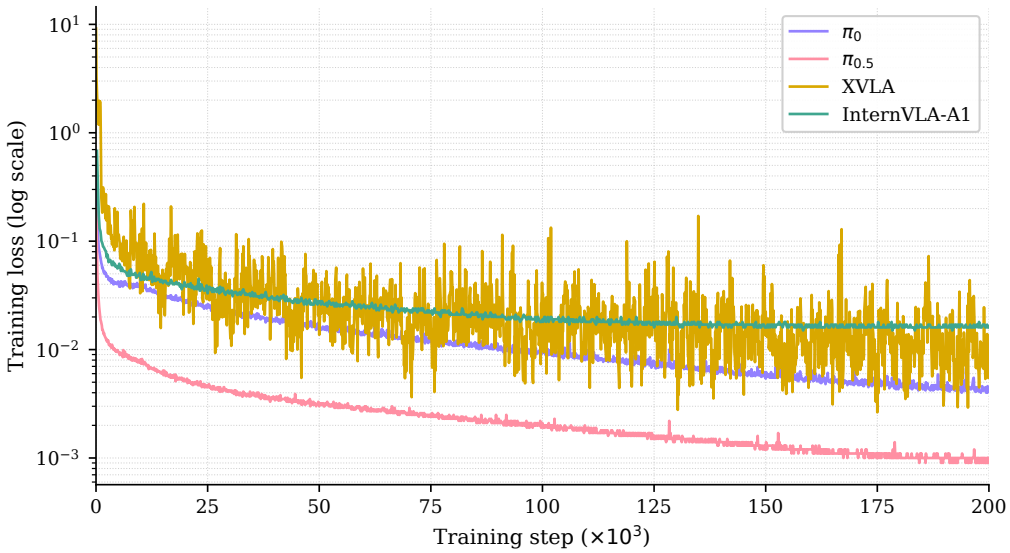

Figure 7. Training loss versus optimizer step for the four evaluated baselines (logarithmic vertical scale). Each loss value is computed by the model's official open-source implementation. Each point on the π_0, $\pi_{0.5}$, curves is the mean training loss over a 100-step window; InternVLA-A1's loss is logged once every 200 steps and is plotted as-is. All four runs target 200k optimizer steps.

E. Controlled Breakdown Analysis

The cluster-level capability analyses in the main paper report the mean performance for each tag category, for example the average SR of all "Mobile" tasks against all "Fixed" tasks. A well-known risk of such cluster-level breakdowns is *multi-factor confounding*: because EBench tasks carry multiple tags simultaneously, an observed difference may reflect the influence of correlated tags rather than the target tag itself. For example, if most "Mobile" tasks are also "Low Precision" and most "Fixed" tasks are "High Precision," then the observed mobile–dexterous gap may partially be a precision effect in disguise.

To obtain *controlled* estimates of each tag's net effect, we use task-level permutation tests with 10,000 iterations. With only $T = 26$ tasks, multiple linear regression is under-powered: the number of tag categories, more than 20, approaches the number of observations, producing unreliable coefficient estimates and inflated standard errors. Permutation tests make no distributional assumptions and preserve the task-level correlation structure, making them the only statistically defensible inference tool at this scale.

For each tag category, for example Mobile vs. Fixed, we compute the observed mean performance difference. We then generate a null distribution by repeatedly shuffling tag labels *at the task level*, not at the trial level, and re-computing the difference, thereby preserving within-task model correlations. For multi-hot atomic skills, we use *stratified permutation*: labels are shuffled only within tasks that share the same scene category, preventing scene–skill confounding. The two-tailed p-value is the proportion of permuted differences whose absolute value exceeds the observed absolute difference.

Table 6 reports the observed mean differences and permutation p-values for the four models.

How to read each cell. Every cell in Table 6 and every panel in Figures 8–10 reports two numbers about one (model, tag-contrast) pair. The first is Δ, the observed mean difference in Test SR between

the two task subsets the contrast names: positive means the model scores higher on the category subset than on the reference subset, in percent. The second is the two-tailed p-value, which equals the fraction of the 10,000 task-level label shuffles whose absolute mean difference was at least as large as $|\Delta|$. A small p means the observed Δ is unusual under random re-assignment of the contrast tags across the same task pool, so the gap is unlikely to be driven by the particular task split alone; a p near 1 means the observed Δ sits inside the bulk of what random re-assignments produce. Concretely, the InternVLA-A1 cell in the Mobile-vs-Fixed row carries $\Delta = +30.9\%$ and p=0.008: InternVLA-A1 scores 30.9 Test-SR points higher on Mobile than on Fixed tasks, and only roughly 80 of the 10,000 random Mobile/Fixed re-shuffles produced an absolute gap that large. Throughout, Δ is the effect size and p is the chance-consistency check; both should be read together with the cell's sample size n, which the table reports in each block header.

Table 6. Controlled tag effects from task-level permutation tests on the Test split, provided as reference. Entries show observed mean difference, category minus reference, in Test SR (%). Bold + stars mark $p < 0.05$ from 10,000 task-level shuffles using the conventional thresholds * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$ (two-tailed); the marks are a visual aid rather than a hypothesis-test claim, and contrasts with p somewhat above 0.05 are still informative when read together with their effect size and sample size. Positive = higher SR than reference.

Tag	π_0	$\pi_{0.5}$	**XVLA**	**InternVLA-A1**
Operating Mode (ref: Fixed, n_{Mobile}=19, n_{Fixed}=7)				
Mobile	+12.0	+20.2	+20.6	**+30.9****
Horizon (ref: Short, n_{Long}=7, n_{Short}=19)				
Long	+1.8	-6.0	-15.4	+2.1
Precision (ref: High, n_{Low}=14, n_{Medium}=8, n_{High}=4)				
Low	+19.4	**+37.1***	+30.9	+26.5
Medium	+0.9	+8.0	+10.1	+14.9
Scene (ref: Industrial, $n_{\text{Industrial}}$=3)				
Bedroom	+20.0	+63.3	+8.3	+35.0
Bathroom	+39.2	+51.7	+31.5	+42.5
Kitchen	+25.7	+37.9	+44.7	+44.5
Living Room	+14.4	+33.3	+14.2	+19.4
Study	+6.7	+18.9	+17.1	+3.3
Supermarket	−5.8	+17.5	+30.8	+9.2
Dining Room	−16.8	+2.0	+4.4	−0.6
Logistics	−18.3	−8.3	−6.7	−11.7
Atomic Skill (has-skill − not-has-skill; scene-stratified within-group shuffle)				
Place (n_+=24)	+13.0	+4.0	−13.8	+3.7
Sweep (n_+=5)	−6.9	−11.5	−9.5	−1.0
Handover (n_+=5)	−7.2	−2.7	−23.5	**−20.1***
Pull (n_+=2)	+13.6	+4.3	+13.2	−1.9
Flip (n_+=2)	−23.8	−12.1	−13.3	−29.9
Pour (n_+=2)	−26.5	−27.5	−26.1	−23.6
Insert (n_+=3)	−27.7	**−32.1***	−26.5	−28.7

Reference statistics. Table 6 and Figures 8, 9, and 10 report the observed mean differences and permutation p-values for every (model, contrast) cell, including the Δ values that already appear in the main paper's cluster-level breakdowns alongside the p-values produced by the shuffle described above. We provide these numbers as a reference for readers who want to gauge how much of each cluster-level observation is consistent with the task-level null; we do *not* claim a binary significance

threshold at $p < 0.05$, and contrasts with p-values somewhat above 0.05 – or even substantially higher – are still informative when read together with their effect size and sample size. The % sign on each Δ refers to a difference in Test SR between two task subsets, not a percentage point of an absolute score.

A few patterns in the table are worth pointing out. InternVLA-A1's Mobile advantage carries the largest effect-size-to-null-spread ratio in the Operating Mode block (Δ= + 30.9%, p=0.008). We omit the Move atomic skill from the Atomic Skill block because Move-tagged tasks coincide almost exactly with the Mobile tasks, so a Move contrast would re-test the Operating Mode signal rather than provide a new piece of information. $\pi_{0.5}$'s Low-precision advantage (Δ= + 37.1%, p=0.030) and its Insert penalty (Δ= – 32.1%, p=0.040) are the two largest single-tag effects on the Precision and Atomic Skill axes. By contrast, the Horizon contrasts and the eight Scene contrasts produce wider nulls than typical effect sizes, which is consistent with their small per-cell sample sizes (e.g., one to six tasks per scene against an Industrial reference of three); this is the controlled-analysis correlate of a hypothesis raised in the main text – the apparent scene rankings, such as Bedroom for $\pi_{0.5}$ at +63% or Bathroom for π_0 at +39%, likely reflect the operating-mode and precision composition of each scene rather than scene-specific visual priors.

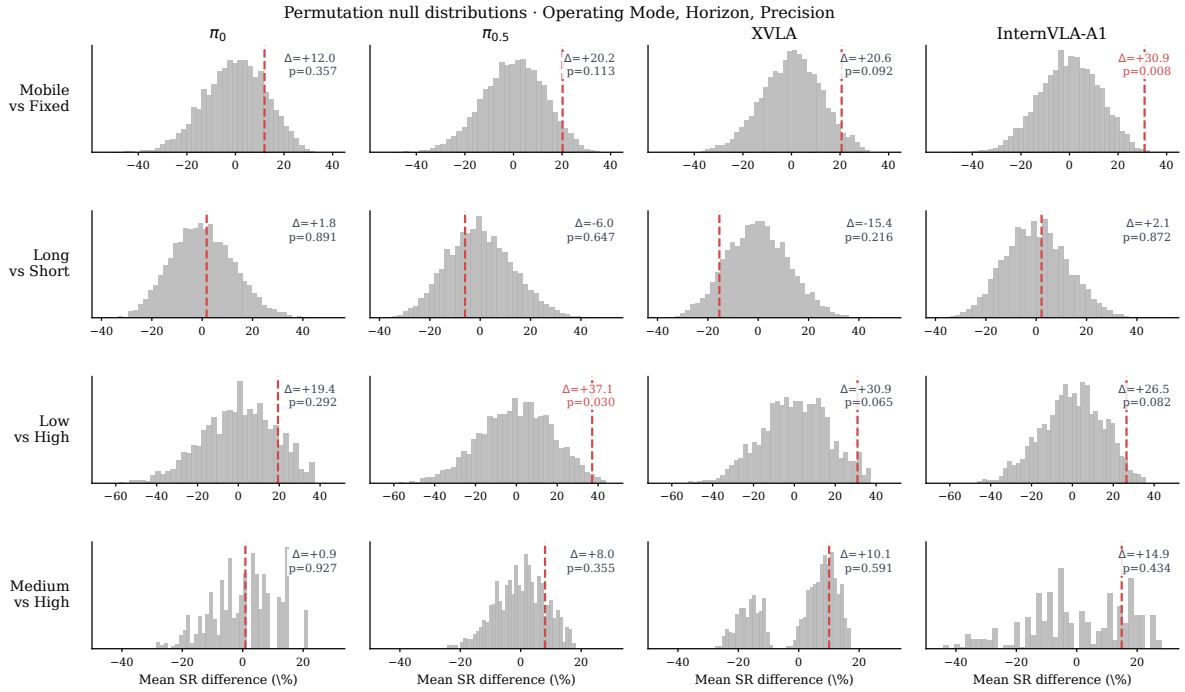

Figure 8. Permutation null distributions and observed differences for the Operating Mode, Horizon, and Precision contrasts in Table 6. Histograms show the null distribution of mean Test SR differences under 10,000 task-level label shuffles; red dashed lines mark the observed differences. Per-cell labels report the observed Δ in % and the two-tailed p-value; the red-text $p < 0.05$ marker is a visual aid rather than a hypothesis-test threshold.

Implications for benchmark design. Cluster-level tables are intuitive and match how practitioners browse capabilities, but they can attribute effects to the wrong tags when categories are correlated. The permutation p-values reported here are not used to make accept/reject claims; they let a reader gauge how much of any given cluster-level observation is consistent with task-level chance, and we recommend that future fine-grained benchmarks include the same kind of reference statistics alongside their cluster-level tables.

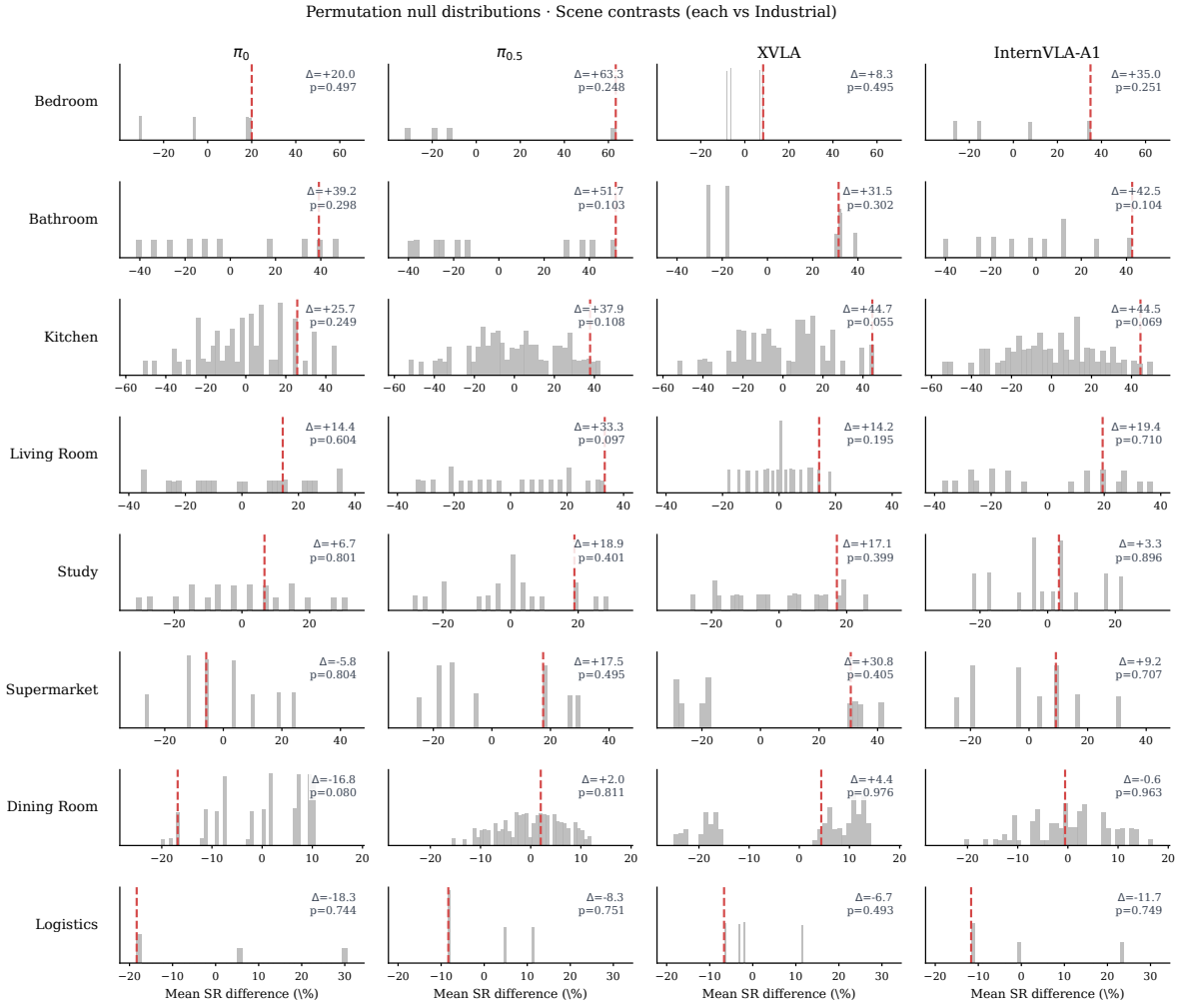

Figure 9. Permutation null distributions and observed differences for the eight Scene contrasts in Table 6, each scene compared against the Industrial reference. Same conventions as Figure 8; the disjoint-category shuffle is restricted to the two scenes' tasks per row. Per-scene sample sizes range from one to six tasks, which is reflected in the relatively wide null distributions.

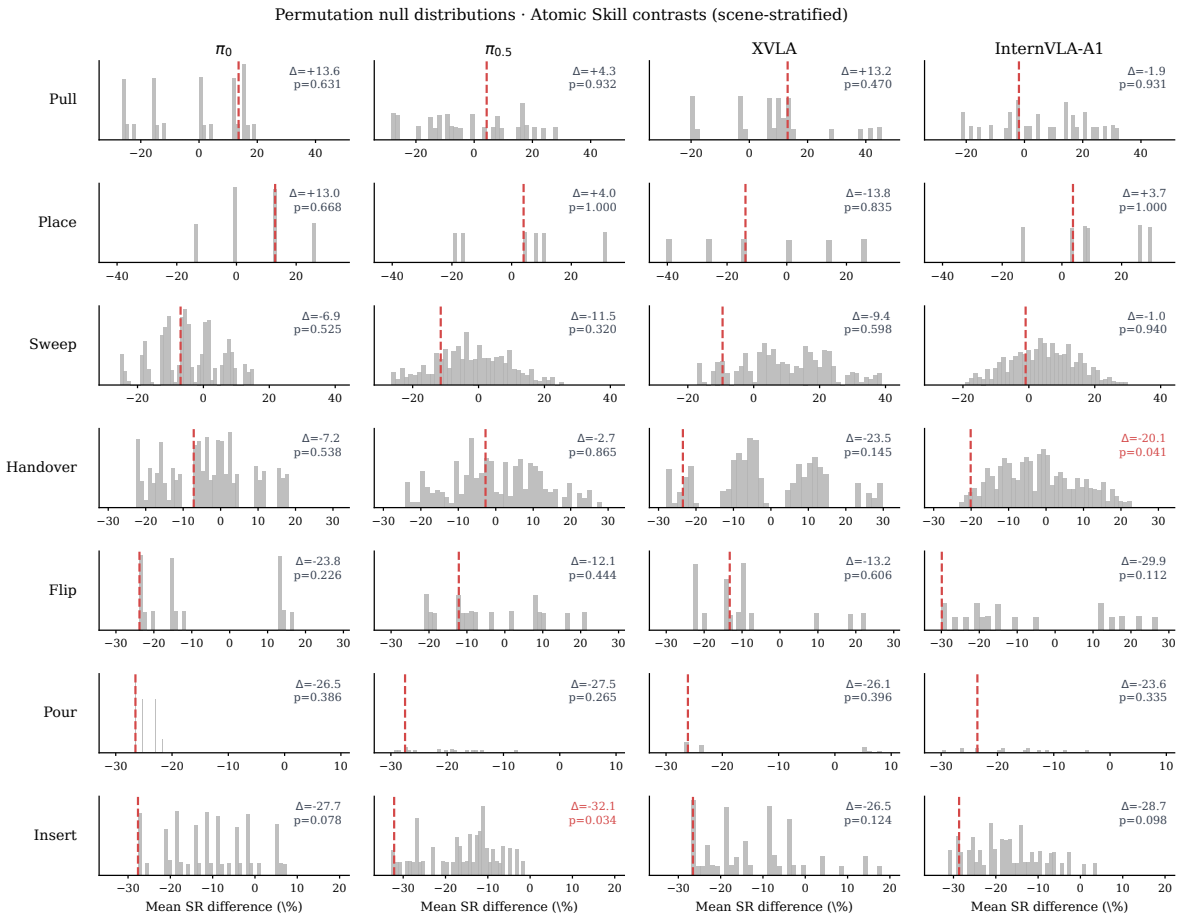

Figure 10. Permutation null distributions and observed differences for the eight Atomic Skill contrasts in Table 6, each *has-skill* versus *not-has-skill*. Same conventions as Figure 8; the scene-stratified within-group shuffle controls for scene–skill confounding.