PARAMETER MERGING FOR CONTINUAL ADAPTATION IN VISION-LANGUAGE-ACTION MODELS *

Ardalan Aryashad
aryashad@usc.edu

Danie Craig Kulandai
dkulanda@usc.edu

Kevin Ngo
kevinngo@usc.edu

Louison Lu
louisonl@usc.edu

Raja Kumar
kumarraj@usc.edu

Shravan Shenoy
shenoysh@usc.edu

Wenwen Han
wenwenha@usc.edu

ABSTRACT

We study continual adaptation of vision-language-action (VLA) models through a controlled comparison of sequential fine-tuning and post-hoc parameter merging. Using SmolVLA as the backbone and three long-horizon LIBERO tasks as the primary benchmark, we train task-specific experts under full fine-tuning and three LoRA regimes (rank 64, tied-A, rank 8) and evaluate six merging operators — linear averaging, magnitude-based merging, task arithmetic, DARE, TIES, and Fisher merging — both as triple-merges and as pair-merges, totalling 63 merge configurations. All 21 triple-merges return 0% success on every task, and only seven of 42 pair-merges are non-zero, all of them recovering only the strongest single-task expert and only when its adapter participates in the merge. Sequential fine-tuning, by contrast, exhibits positive forward transfer at every stage but completely loses prior tasks. Diagnostics — early-checkpoint merging, capacity-reduced and tied-A LoRA, and qualitative trajectory inspection — localize the failure to the structure of the trained adapters rather than to operator choice or training-time drift. We additionally report a continual-learning study on RoboTwin in the appendix that is consistent with our LIBERO findings.

1 INTRODUCTION

Vision-language-action (VLA) models (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Shukor et al., 2025) have emerged as a unified interface for robotic manipulation, mapping camera observations, robot state, and natural language instructions into continuous control actions. By inheriting representations from large vision-language pre-training and adapting them with imitation learning on demonstration data, VLAs generalize across tasks far better than policies trained from scratch. However, deploying a VLA over time inevitably brings new tasks, new objects, and new environments, and a single fixed checkpoint quickly becomes insufficient. The natural response — continued fine-tuning on each new task — is well-known to overwrite previously acquired skills, a phenomenon called catastrophic forgetting (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017). The continual adaptation problem for VLAs is therefore not whether to adapt, but how to adapt without losing what was learned before.

A complementary line of work has shown that independently fine-tuned copies of the same pre-trained model can often be combined post-hoc by simple weight-space operations: averaging (Wortsman et al., 2022), task-vector arithmetic (Ilharco et al., 2023), magnitude-based selection, redundancy reduction (DARE) (Yu et al., 2024), sign-aware merging (TIES) (Yadav et al., 2023), and Fisher-weighted averaging (Matena & Raffel, 2022). In language and vision, these methods recover much of multi-task fine-tuning while requiring no joint training or replay data. If parameter merging

*Author ordering is alphabetical.

works equally well for VLAs, it would offer a clean alternative to sequential fine-tuning: train one expert per task in isolation, then merge.

(Yadav et al., 2025) tests this hypothesis and find that simple average merging strategy works well even for VLA. In this work, we take this a step further and validate the different merging strategy to find the best merging strategy. Using SmolVLA (Shukor et al., 2025) as a compact, reproducible VLA backbone and the LIBERO (Liu et al., 2023) long-horizon manipulation benchmark, we train task-specific experts under both full fine-tuning (FFT) and three LoRA (Hu et al., 2022) variants (rank 64, tied-A, rank 8). We then apply six merging strategies to combine those experts into a single multi-task policy. As an upper baseline of behavioral capacity, we also run sequential fine-tuning along the same task sequence. To complement the LIBERO study, we additionally examine continual fine-tuning behavior on RoboTwin 2.0 (Chen et al., 2025) dual-arm tasks and report those results in the appendix.

Our findings are negative for post-hoc merging in this setting. **Across 21 triple-merge configurations** (six operators applied to all three adapter regimes), every merged policy returns 0% success on every task. **Across 42 pair-merge configurations**, only seven cells are non-zero, and all of them recover partial behavior of the strongest single expert (the moka-pot task) when that expert participates in the merge — the other tasks are never recovered. Sequential fine-tuning loses each prior task immediately as soon as adaptation moves to the next, but does succeed at the most recent task. The contrast between merging and sequential fine-tuning, together with diagnostics that rule out training-time drift and adapter-capacity mismatch, suggests that independently-trained VLA experts in our setting do not occupy a parameter subspace that any of the tested operators can recombine into a working multi-task policy. Our contributions are:

- A controlled comparison of *sequential fine-tuning* and *post-hoc parameter merging* as continual-adaptation strategies for VLA models, using a common SmolVLA backbone, the same task sequence, and matched evaluation protocols.
- A merging sweep covering six operators across three LoRA regimes and one full fine-tuning regime, evaluated as both triple- and pair-merges.
- Diagnostics — early-checkpoint merging, capacity-reduced (rank 8) and tied-A LoRA, and qualitative trajectory inspection — that localize the failure mode to the structure of the trained adapters rather than to operator choice or training-time drift.

2 Related Work

Vision-language-action models. Recent VLA models adapt large pre-trained vision-language models to output low-level robot actions. RT-2 (Brohan et al., 2023) co-fine-tunes a VLM on web data and robot trajectories. OpenVLA (Kim et al., 2024) releases an open-source 7B-parameter VLA trained on the Open X-Embodiment dataset (Padalkar et al., 2023). Octo (Octo Model Team et al., 2024) is a transformer policy supporting flexible action and observation spaces, and the π_0 family (Black et al., 2024) introduces flow-matching action experts on top of a VLM backbone. SmolVLA (Shukor et al., 2025) is a compact ($\sim$450M parameter) open VLA built within the LeRobot ecosystem and trained with flow-matching action chunks; we use SmolVLA as our backbone because it can be fine-tuned end-to-end on a single GPU and supports both full fine-tuning and parameter-efficient adaptation.

Continual learning approaches in VLA and robot policies. Continual or lifelong learning addresses the loss of previously-learned skills when a model is adapted to new data (McCloskey & Cohen, 1989). Classical strategies fall into three families: regularization-based methods such as Elastic Weight Consolidation (Kirkpatrick et al., 2017) and Learning without Forgetting (Li & Hoiem, 2018), replay-based methods, and architectural approaches that assign new parameters per task. The LIBERO benchmark (Liu et al., 2023) was specifically designed to study these strategies in robotic manipulation and notably found that naive sequential fine-tuning is a surprisingly competitive baseline for forward transfer, while still suffering from severe forgetting. Continual learning in VLAs is comparatively underexplored; recent work in this direction has mainly focused on evaluation protocols and on parameter-efficient adaptation (Hu et al., 2022) as a way to reduce the footprint of each new task. Our study fits this line: we treat LoRA adapters and full fine-tuning checkpoints as task

experts and ask whether they can be combined post-hoc rather than trained jointly or with explicit replay.

Model merging. A separate body of work shows that independently fine-tuned models can be merged in weight space. Model Soups (Wortsman et al., 2022) averages checkpoints of the same fine-tuning run and improves accuracy at no inference cost. Task arithmetic (Ilharco et al., 2023) treats $\theta_i - \theta_{\text{base}}$ as a task vector and combines tasks by addition or negation. TIES-Merging (Yadav et al., 2023) addresses sign conflicts and parameter redundancy across task vectors, and DARE (Yu et al., 2024) randomly drops and rescales delta-parameters to suppress interference. Fisher-weighted averaging (Matena & Raffel, 2022) weights each task's parameters by its diagonal Fisher information. These methods have produced strong results in NLP and vision, but their behavior on VLAs — which combine a heterogeneous backbone (vision encoder, language model, action head) and are trained on demonstration trajectories rather than next-token prediction — has been studied much less. Our experiments evaluate all six operators on a common LIBERO expert set and an additional RoboTwin (Mu et al., 2025; Chen et al., 2025) continual-learning study (deferred to the appendix).

3 METHOD

3.1 PROBLEM FORMULATION

We study continual adaptation of a vision-language-action (VLA) policy $\pi_\theta : \mathcal{O} \times \mathcal{L} \to \mathcal{A}$ over a sequence of manipulation tasks $\{\tau_1, \ldots, \tau_T\}$. Each task τ_i is associated with demonstrations $\mathcal{D}_i = \{(o^{(k)}, l_i, a^{(k)})\}_{k=1}^{N_i}$, where $o^{(k)}$ contains multi-view RGB observations (and proprioceptive state when available), l_i is a natural-language instruction, and $a^{(k)}$ is a low-level action sequence. Our base checkpoint is θ_0 (SmolVLA initialized from a LIBERO-aligned checkpoint; see Section 4.1). We compare two routes to a single multi-task policy:

Sequential fine-tuning. Starting from θ_0, we train stage-wise: $\theta_1 = \text{Train}(\theta_0, \mathcal{D}_1)$, $\theta_2 = \text{Train}(\theta_1, \mathcal{D}_2)$, and so on. After stage i, we obtain a continual candidate θ_i^{seq}.

Parameter merging. We train task experts independently from θ_0: $\theta_i = \text{Train}(\theta_0, \mathcal{D}_i)$ for $i = 1, \ldots, T$. A merge operator M combines them without additional gradient updates: $\theta^{\text{merge}} = M(\theta_1, \ldots, \theta_T \mid \theta_0)$. This decouples per-task training from cross-task integration and enables parallel expert training.

Evaluation metrics. For each candidate policy and each task j, we report success rate S_j over fixed evaluation episodes (with fixed seeds and episode counts per experiment cell). We report:

- **In-distribution (ID) success**: performance under nominal simulator settings.
- **Out-of-distribution (OOD) success**: performance under controlled simulator perturbations (e.g., lighting, camera pose, material appearance, and pixel noise).
- **Forward transfer (FWT)**: improvement on later tasks from sequential initialization relative to training those tasks from θ_0 alone.
- **Backward transfer / forgetting**: performance drop on earlier tasks after subsequent stages of sequential training, e.g., $S_j(\theta_j^{\text{seq}}) - S_j(\theta_i^{\text{seq}})$ for $j < i$.

3.2 PARAMETER MERGING STRATEGIES

We instantiate M with six operators that span the main families used in the literature. For each strategy below, θ_{base} denotes the shared pre-trained initialization and $\tau_i = \theta_i - \theta_{\text{base}}$ the task vector (Ilharco et al., 2023). For LoRA experts, $\theta_i = \theta_{\text{base}} + B_i A_i$.

Linear weight averaging (Model Soup). The simplest baseline assumes the fine-tuned models lie within a single low-loss basin so that their average remains low-loss (Wortsman et al., 2022):

$$\theta_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} \theta_i, \qquad \Delta W_{\text{avg}}^{\text{LoRA}} = \frac{1}{N} \sum_{i=1}^{N} B_i A_i.$$

Magnitude-based merging (Max-Weight). For each parameter j, retain the value with largest absolute deviation from base, on the assumption that the expert that moved a parameter the most is the expert most committed to it:

$$\theta^{(j)}_{\text{merged}} = \underset{x \in \{\theta^{(j)}_1, \ldots, \theta^{(j)}_N\}}{\arg\max} |x - \theta^{(j)}_{\text{base}}|.$$

Task arithmetic. Sum task vectors and add back to the base, scaled by a coefficient λ (Ilharco et al., 2023):

$$\theta_{\text{merged}} = \theta_{\text{base}} + \lambda \sum_{i=1}^{N} \tau_i.$$

DARE (Drop-And-REscale). Sparsify each task vector before summing (Yu et al., 2024): draw a Bernoulli mask M with drop rate p, rescale by $1/(1-p)$, then sum:

$$\hat{\tau}_i = \frac{M \odot \tau_i}{1-p}, \qquad \theta_{\text{merged}} = \theta_{\text{base}} + \sum_{i=1}^{N} \hat{\tau}_i.$$

TIES-Merging. Resolve sign conflicts and trim noise across task vectors (Yadav et al., 2023): (i) **Trim** each τ_i to its top-$k\%$ entries by magnitude; (ii) **Elect Sign** $S^{(j)} = \text{sgn}\Big(\sum_i \tau^{(j)}_i\Big)$; (iii) **Disjoint merge** averages only entries that agree with the elected sign:

$$\theta^{(j)}_{\text{merged}} = \theta^{(j)}_{\text{base}} + \frac{1}{|C_j|} \sum_{i \in C_j} \tau^{(j)}_i, \quad C_j = \{i : \text{sgn}(\tau^{(j)}_i) = S^{(j)}\}.$$

Fisher merging. Weight each expert's parameters by its diagonal Fisher information (Matena & Raffel, 2022):

$$F^{(j)}_i = \mathbb{E}_{x \sim \mathcal{D}_i}\Big[\big(\nabla_{\theta^{(j)}_i} \log p(y \mid x, \theta_i)\big)^2\Big], \qquad \theta_{\text{merged}} = \frac{\sum_i F_i \odot \theta_i}{\sum_i F_i}.$$

This is the strongest theoretical candidate of the six because it explicitly preserves parameters that are important to each task's likelihood.

4 Experiments

4.1 Setup

Backbone and pre-training. All experiments use SmolVLA (Shukor et al., 2025) (∼450M parameters) as the underlying VLA. The model encodes multi-view RGB observations, robot state, and a natural language instruction into a shared representation, and decodes continuous action chunks of horizon 50 with flow matching. To better align the backbone with the LIBERO embodiment and instruction style, we additionally pre-train SmolVLA on LIBERO-90 (the 90-task pretraining split of LIBERO (Liu et al., 2023)) before any task-specific fine-tuning. The resulting checkpoint, SmolVLA-LIBERO90, serves as θ_0 for all downstream experiments. We considered the larger OpenVLA (Kim et al., 2024) (7B parameters) as an alternative backbone but found its compute footprint incompatible with the iteration speed required for a six-operator $\times$ four-regime sweep.

Tasks. We focus on three representative LIBERO-10 tasks chosen for their long-horizon, multi-object structure: **T2** ("*turn on the stove and place the moka pot on it*", 41 episodes), **T4** ("*put the white mug on the left plate and the yellow-and-white mug on the right plate*", 38 episodes), and **T7** ("*put both the alphabet soup and cream cheese box in the basket*", 43 episodes). T2 contains a single coordinated subtask (knob-turn followed by pick-and-place); T4 and T7 each require coordinating two distinct objects. We additionally study two structurally similar RoboTwin (Chen et al., 2025) tasks (*Click Bell* and *Click Alarm Clock*) to probe ID/OOD behavior under bimanual control; those results are deferred to Appendix A.

Sequential vs. merging pipelines. Figure 1 summarizes the three policy generation routes evaluated in this paper. (i) *Expert policies* are obtained by fine-tuning SMOLVLA-LIBERO90 on a single task under either full fine-tuning (FFT) or one of three LoRA variants. (ii) *Sequential policies* are obtained by continuing fine-tuning across the task order T2→T4→T7, where each stage is initialized from the best-validation checkpoint of the previous stage. (iii) *Merged policies* are obtained by combining independently-trained experts using one of the six operators of Section 3.2, with no additional training.

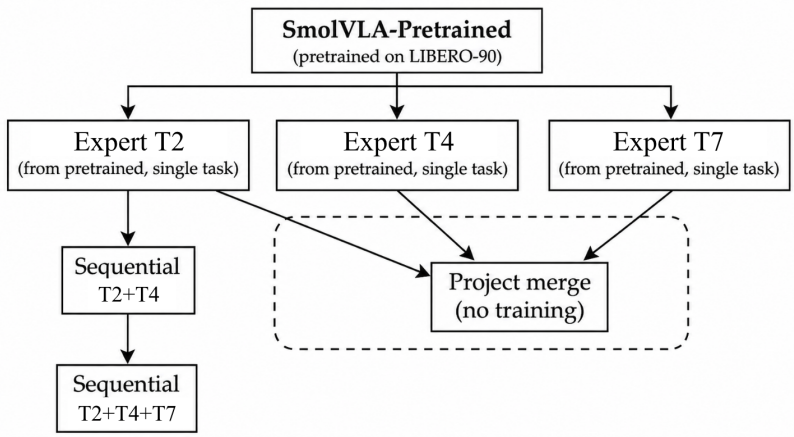


Figure 1: Policy generation pipelines. Single-task experts are fine-tuned independently from SMOLVLA-LIBERO90. Sequential policies continue fine-tuning along T2→T4→T7. Merged policies combine independently-trained experts via the operators in Section 3.2.

Adapter regimes. For LoRA experts we evaluate three configurations designed to test whether constraining adapter capacity changes the merging outcome: (i) **LoRA** $r = 64$ (standard, $\alpha = 64$, $\sim$3.0M trainable parameters); (ii) **Tied-A**: A is initialized from a shared seed and frozen across experts so only B is trained ($\sim$1.5M parameters), making the A matrices trivially identical at merge time; (iii) **LoRA** $r = 8$ ($\alpha = 8$, $\sim$0.37M parameters), which compresses each expert by $8\times$ and forces it toward low-dimensional, potentially shareable structure. We compare against **FFT** ($\sim$457M trainable parameters), $\sim 150\times$ the standard LoRA budget.

Evaluation. LIBERO success rates are computed over 50 evaluation episodes per task. For each fine-tuning run, intermediate checkpoints are saved every 500 steps and the best-validation checkpoint is selected by per-task success rate, rather than using the final checkpoint, to reduce sensitivity to late-training instability. RoboTwin success rates are computed over 10 simulation runs per setting (ID and OOD) and follow the protocol described in Appendix A.

4.2 IMPLEMENTATION DETAILS

All training was run with the LeRobot `lerobot-train` pipeline. Each LIBERO fine-tuning run is 10,000 steps with batch size 64, action chunk size 50, action horizon 10, Adam optimizer, and the standard LeRobot LR schedule; intermediate checkpoints are saved every 500 steps. RoboTwin fine-tuning uses the same optimizer settings with 1,000 steps and a 14-dimensional action space (Aloha AgileX dual-arm). Because RoboTwin is not natively in the LeRobot format, we built a conversion pipeline that re-formats trajectories into LeRobotDataset v3.0 (Appendix A). All triple-, pair-, and early-checkpoint merge configurations were implemented in PyTorch on top of the saved expert checkpoints; for Fisher merging the diagonal Fisher information was estimated from a held-out subset of each task's demonstration data. Hyperparameters used in this study are listed in Appendix E.

4.3 RESULTS

We report LIBERO results in this section and defer the RoboTwin study to Appendix A.

Single-task expert performance. Table 1 reports the best-validation success rate per task under each regime. Two patterns are consistent. First, T2 is the easiest task: all four regimes reach $\geq 72\%$. T4 and T7 are substantially harder, requiring two-object coordination. Second, the gap between FFT and LoRA $r = 64$ widens with task difficulty (20pp on T2, 34pp on T4, 30pp on T7), and reducing LoRA rank from 64 to 8 has a graceful effect on T2 (-4pp) but a sharp effect on T4 (-12pp). Adapter capacity, not adapter form alone, limits the harder tasks.

Table 1: LIBERO expert performance across adapter regimes (best-val success rate over 50 episodes; checkpoint step in parentheses).

Task	FFT (457M)	LoRA $r = 64$ (3.0M)	Tied-A (1.5M)	LoRA $r = 8$ (0.37M)
T2 (moka pot)	96% (3500)	76% (3500)	76% (5000)	72% (9500)
T4 (mugs on plates)	60% (3500)	26% (3500)	18% (3000)	14% (6000)
T7 (soup in basket)	62% (6000)	32% (9500)	22% (3000)	28% (4000)

Cross-task evaluation: experts are true specialists. To verify that the trained experts are not incidentally generalizing across tasks, we evaluate each best-val expert on all three target tasks (Table 2). Every off-diagonal cell is exactly 0% under both LoRA $r = 64$ and FFT. Even with $150\times$ more trainable parameters, FFT produces no incidental cross-task capability. The implication is unambiguous: any merged policy that succeeds on more than one task must come from the merging operation itself, not from latent capability shared across the experts.

Table 2: Cross-evaluation of LoRA $r = 64$ and FFT experts on all three tasks. All off-diagonal cells are 0%.

Regime	Expert	Eval T2	Eval T4	Eval T7
LoRA $r = 64$	Expert T2 (3500)	**76%**	0%	0%
	Expert T4 (3500)	0%	**26%**	0%
	Expert T7 (9500)	0%	0%	**32%**
FFT	Expert T2 (3500)	**96%**	0%	0%
	Expert T4 (3500)	0%	**60%**	0%
	Expert T7 (6000)	0%	0%	**62%**

Sequential fine-tuning: positive forward transfer, complete forgetting. Sequential fine-tuning starts from the strongest single-task expert (T2 best-val), continues on T4, then on T7. Table 3 reports per-task success at each stage. The forgetting pattern is identical under LoRA $r = 64$ and FFT: training on T4 immediately erases T2 ($76\%/96\% \rightarrow 0\%$) despite T2 being the initialization, and the next stage erases both T2 and T4. The forward-transfer pattern, however, differs across regimes. Under LoRA, the T2$\rightarrow$T4 stage exceeds the in-domain T4 expert (42% vs. 26%), indicating that initialization from a related task provides a stronger starting point than the pretrained base. Under FFT, the same effect is not observed (44% vs. 60%), likely because the FFT T4 expert already operates near the task ceiling and leaves little room for forward-transfer improvement. That the catastrophic forgetting holds even with FFT's $150\times$ larger parameter budget rules out limited adapter capacity as the cause of forgetting; it is a property of sequential adaptation in this setting and motivates parameter merging as an alternative.

Table 3: Sequential fine-tuning. Each row reports per-task success rate at the end of that stage.

Regime	Stage	T2	T4	T7
LoRA $r = 64$	T2 only (= Expert T2)	76%	0%	0%
	T2 $\rightarrow$ T4 (best-val 3500)	0%	42%	0%
	T2 $\rightarrow$ T4 $\rightarrow$ T7 (8000)	0%	0%	24%
FFT	T2 only	96%	0%	0%
	T2 $\rightarrow$ T4	0%	44%	0%
	T2 $\rightarrow$ T4 $\rightarrow$ T7	0%	0%	24%

Triple-merge: every operator returns 0% on every task. For each of the three LoRA regimes, the best-val checkpoints of all three experts were combined under each of the six operators (21 configurations total) and evaluated on T2, T4, T7 (Table 4). Every cell is 0%. This holds for the linear-average baseline, for selection-based operators (magnitude, TIES), for redundancy reduction (DARE), and for importance-weighted Fisher merging. It also holds for the lower-capacity tied-A and $r = 8$ regimes designed to reduce inter-expert interference. To rule out training-time drift as the cause, we additionally merged the standard $r = 64$ experts at step 1500 (well below best-val); this earlier merge also returned 0% on all tasks, indicating that the merge difficulty is structural to the independently-trained adapters rather than the result of accumulated divergence.

Table 4: Triple-merge results. Six operators applied to all three LoRA adapter regimes, evaluated on each task. Every cell is 0%.

	LoRA $r = 64$			Tied-A			LoRA $r = 8$		
Strategy	T2	T4	T7	T2	T4	T7	T2	T4	T7
Linear averaging	0	0	0	0	0	0	0	0	0
Magnitude (Max-Weight)	0	0	0	0	0	0	0	0	0
Task arithmetic	0	0	0	0	0	0	0	0	0
DARE	0	0	0	0	0	0	0	0	0
TIES	0	0	0	0	0	0	0	0	0
Fisher	0	0	0	0	0	0	0	0	0

Pair-merge: only T2 ever recovers, and only when its expert is included. To probe whether the failure stems from combining all three experts at once, we additionally merged each pair under each of the six operators on the tied-A and $r = 8$ adapter sets, plus six linear-average pair merges on $r = 64$ (42 configurations total). The full pair-merge sweep is reported in Appendix B (Table 6); Table 5 lists the seven non-zero cells. The pattern is starkly asymmetric: *every* non-zero cell appears only on T2, only when T2's expert is one of the two adapters merged. T4 and T7 never appear non-zero in any merged configuration. The strongest cell, tied-A magnitude pruning of $\{T2, T7\}$, recovers 8% on T2 (4 of 50 episodes); the remaining cells range from 2% to 4%.

Table 5: All non-zero pair-merge cells (out of 42). Every non-zero result is on T2 and only when T2's expert is one of the merged adapters.

Strategy	Pair / Adapter set	Result
Magnitude pruning	{T2, T7} / Tied-A	8% on T2
TIES	{T2, T7} / $r = 8$	4% on T2
Linear averaging	{T2, T4} / Tied-A	2% on T2
Linear averaging	{T2, T7} / Tied-A	2% on T2
Linear averaging	{T2, T7} / $r = 8$	2% on T2
Task arithmetic ($\alpha = 0.4$)	{T2, T7} / Tied-A	2% on T2
Task arithmetic ($\alpha = 0.7$)	{T2, T7} / Tied-A	2% on T2

Qualitative analysis of the strongest non-zero cell. Because non-zero cells could in principle reflect random success from initial-state alignment rather than retained task structure, we inspected the success episodes of the two strongest cells frame-by-frame. Figure 2 shows representative frames from the 8% cell (tied-A magnitude pruning of $\{T2, T7\}$): the gripper deliberately closes on the stove knob and rotates it (igniting the burner), then crosses the workspace, grasps the moka pot, and places it on the lit burner — both subtasks of T2 are executed with coordinated motion. The 4% cell (LoRA $r = 8$ TIES of $\{T2, T7\}$) shows a partially degraded version of the same behavior: the pick-and-place subtask is intact, but the knob is no longer turned with a deliberate grasp — the burner happens to ignite as the arm sweeps through the knob area en route to the pot (Appendix Figure 5). This is consistent with merging preserving a subset of the dominant expert's capability, with coarser motor primitives (pick-and-place) surviving while finer primitives (rotational grasp-and-turn) degrade first. No analogous recovery is observed for T4 or T7.

(a) Initial (b) Knob turn (c) Approach pot (d) Grasp pot (e) Pot placed

Figure 2: Strongest non-zero pair-merge cell: tied-A magnitude pruning of $\{T2, T7\}$, 8% on T2. The gripper deliberately closes on the stove knob and rotates it before grasping and placing the moka pot — both subtasks of T2 are executed with coordinated motion.

4.4 Discussion

Three observations together rule out the most common explanations for the merge failure observed above.

(1) Training-time drift is not the cause. The early-merge diagnostic at step 1500 — well before either expert has converged or substantially diverged from the base — also returns 0%. If the failure were due to experts moving too far apart in parameter space, this earlier merge would have at least partially succeeded.

(2) Adapter-side conflict and capacity are not the cause. The tied-A regime removes A-matrix conflict by construction (every expert's A is identical), so any remaining merge difficulty must come from the B-side alone. The $r = 8$ regime forces each expert into a $8\times$ smaller subspace, which should encourage shared low-dimensional structure. Both fail. This rules out interpretations in which the failure stems from A-side conflict, excess capacity, or insufficiently shared structure.

(3) Importance weighting does not help. Fisher merging — the strongest theoretical candidate among the six operators, since it weights each expert's parameters by task-specific importance — returns 0% on every triple-merge and every pair-merge cell. If different tasks used genuinely disjoint parameter subsets, Fisher's importance weighting should preserve each task's critical structure. The uniform null suggests that the LoRA adapters in this VLA setting do not localize task-specific information into combinable parameter subspaces in the first place.

The qualitative T2-only recovery fits this view: merging does not interleave structure from multiple experts, it preserves a (decaying) subset of whichever expert dominates the merge. Because T2 is the strongest individual expert at 76%, the dominance falls to T2's expert in every non-zero cell, and the recovered behavior is a strict subset of T2's behavior. T4 and T7 never become dominant and so are never recovered. This is consistent with sequential fine-tuning's behavior: in our setting, the trained representations seem to be *positionally consumable* (a later task can build on an earlier one as initialization, with positive forward transfer) but not *post-hoc combinable* (the same representations cannot be re-aggregated from independently-trained copies). A similar continual-fine-tuning phenomenon is observed on RoboTwin in Appendix A.

5 Limitations

Our findings are negative for post-hoc parameter merging, and that conclusion comes with several caveats. (i) *Backbone scope.* All experiments use SmolVLA at $\sim$450M parameters; whether merging behavior changes for larger VLAs such as OpenVLA-7B or π_0 is left open. Larger backbones may concentrate task-specific information in different parameter subsets, where merging operators could interact differently. (ii) *Task and benchmark scope.* Our LIBERO study covers three long-horizon tasks (T2, T4, T7) selected for their multi-object structure, and our RoboTwin study covers two structurally similar bimanual tasks. The behavior of merging on a much wider task suite, on tasks that share more visual or motor structure (e.g., a family of pick-and-place variants), or on cross-embodiment settings remains untested. (iii) *Operator coverage.* We evaluate six operators that span the main families used in the literature, but more recent merging strategies — e.g., methods that explicitly account for representation alignment, layer-wise interpolation schedules, or

merging-aware fine-tuning — could produce different outcomes. (iv) *Evaluation budget.* LIBERO success rates are computed over 50 episodes per task and RoboTwin over 10 simulation runs per setting due to compute constraints; small non-zero merges (2–8%) carry meaningful Monte-Carlo noise, although the qualitative trajectory analysis confirms that the 8% T2 result reflects retained task structure rather than chance. (v) *Adaptation regime scope.* We compare LoRA at three ranks and FFT, but parameter-efficient methods that change the geometry of the update (e.g., adapters with shared bottlenecks, prefix tuning) might lead to merge-friendlier experts.

6 CONCLUSION

We presented a controlled comparison of sequential fine-tuning and post-hoc parameter merging as continual-adaptation strategies for a vision-language-action model, using SmolVLA as a common backbone, three long-horizon LIBERO tasks as the primary benchmark, and an additional RoboTwin study reported in the appendix. Across 21 triple-merge and 42 pair-merge configurations spanning six merging operators (linear averaging, magnitude-based merging, task arithmetic, DARE, TIES, and Fisher merging) and four adapter regimes (FFT, LoRA $r = 64$, tied-A LoRA, LoRA $r = 8$), no triple-merge produced a working multi-task policy and only seven pair-merge cells were non-zero — all of them recovering only the strongest single task and only when that task's expert participated in the merge. Sequential fine-tuning, by contrast, succeeded at the most recent task at every stage but completely lost prior tasks. Diagnostics — early-checkpoint merging, capacity-reduced and tied-A LoRA, and qualitative trajectory inspection — localize the failure to the structure of the trained adapters rather than to operator choice or accumulated training-time drift. Independently fine-tuned VLA experts in this setting therefore do not appear to occupy parameter subspaces that any of the six tested operators can recombine. We see this as motivation for training-time interventions that explicitly target merge-compatibility — shared adapter structures, joint regularization across experts, and merging-aware fine-tuning — as a more promising route to continual adaptation in VLAs than purely post-hoc operators.

7 CONTRIBUTIONS

Ardalan Aryashad: I conducted training on the RoboTwin benchmark and analyzed the effects of continual learning, particularly catastrophic forgetting. In addition, I implemented and executed training of SmolVLA on the LIBERO benchmark using the LeRobot framework. I developed and documented a complete training pipeline in Google Colab and provided trained model checkpoints, which were used by other team members for evaluation and parameter merging experiments.

Louison Lu: I contributed to the implementation and experimental design of the SmolVLA training pipeline on LIBERO. I ran fine-tuning experiments, uploaded trained checkpoints to Hugging Face for team use, and verified checkpoint accessibility and output correctness for downstream evaluation. I also contributed to the early research plan and proposal structure, reviewed training code and commands, and identified checkpoint-selection issues that could affect fair comparison in sequential training. In addition, I investigated OpenVLA as an alternative VLA model and evaluated its feasibility under our available compute constraints, which helped inform the team's decision to focus on SmolVLA.

Danie Craig Kulandai: I owned the LoRA fine-tuning track of the project on the LIBERO-10 benchmark, including expert training under three adapter regimes (standard $r = 64$, tied-A, and $r = 8$ LoRA), specialist cross-evaluation, and sequential fine-tuning across the T2→T4→T7 task sequence. I also conducted the investigation on the merged models received from Wenwen: a triple-merge sweep across all six merge operators on all three adapter regimes, a pair-merge sweep on the tied-A and $r = 8$ adapter sets, an early-merge diagnostic ruling out training-time drift, and qualitative analysis of the two strongest non-zero pair-merge configurations via per-episode video inspection. I prepared the LIBERO dataset conversion to LeRobot v3.0 format, set up the LoRA training and evaluation pipelines on the server and provided best-validation LoRA related checkpoints to the team for downstream merging experiments.

Raja Kumar: My contribution to the project focused on designing and executing a structured training and evaluation pipeline on the LIBERO benchmark. I proposed and implemented an approach that leverages a pretrained base model trained on LIBERO_90, followed by targeted validation

on LIBERO_10 tasks to assess performance under both in-distribution (ID) and out-of-distribution (OOD) settings. I set up the experimental framework using the SmolVLA model within the LeRobot ecosystem. I conducted large-scale pretraining on LIBERO_90, followed by fine-tuning on downstream tasks, and performed evaluations to analyze generalization and task-specific performance. I also compiled everyone's progress and results into the final report. In addition, I provided GPU resources for the team, enabling faster experimentation and efficient iteration on model training and evaluation.

Wenwen Han: I designed and implemented the model merging pipeline for both LoRA adapter and full fine-tuning (FFT) settings across different expert tasks. I developed six parameter merging strategies — Linear Average, Magnitude-based, Task Arithmetic, DARE, TIES-Merging, and Fisher-weighted Merging — implemented them in PyTorch, validated all merged checkpoints, and uploaded them to HuggingFace Hub for downstream evaluation. I also wrote the Merging Strategies section of the final report, describing the theoretical motivation and implementation details for each method.

Kevin Ngo: During the initial stages of the project, I developed the data-format conversion scripts to make RoboTwin 2.0 training data compatible with the LeRobot robotics training framework (LeRobot v3.0 format). For the final experiment layout where LIBERO was used, I owned the full fine-tuned expert policy generation and evaluation track. Specifically, I trained the expert policies for the 3 LIBERO-10 tasks chosen by the team. Then, I identified the best checkpoint for each expert policy by evaluating local rollouts. Finally, I uploaded the best checkpoints to Hugging Face to enable the team to use the checkpoints in the sequential policy generation and merged policy generation. I also helped generate and evaluate sequential policies built from these full fine-tuned experts, which served as a comparison point against the merged-policy results.

Shravan Shenoy: I contributed to the project's experimental setup, evaluation infrastructure, and analysis workflow. Early on, I helped prepare the team's background material to align on VLA foundations, sequential fine-tuning, and parameter-merging approaches, and I developed the project blueprint framework to keep training/evaluation protocols and reporting consistent across contributors with respect to iterations, hyperparameters and model checkpoints. For experimentation, I implemented and iterated on the MuJoCo-based evaluation harness used in our LIBERO studies, including reproducible run orchestration, rollout generation, and OOD scenario support (e.g., lighting, camera, and observation perturbations) to evaluate policy overfitting and generalisability. I also supported the comparative analysis of expert, sequential, and merged policies by running evaluations, organizing outputs, and synthesizing both metric-based and rollout-based findings for project discussions and presentation material.

A RoboTwin Continual-Learning Study

This appendix reports the RoboTwin (Mu et al., 2025; Chen et al., 2025) study referenced in Section 4.3. RoboTwin 2.0 is a benchmark for bimanual manipulation consisting of 50 dual-arm tasks and over 100,000 trajectories generated through an automated expert pipeline with structured domain randomization across clutter, lighting, background, and language instructions. It is therefore particularly well-suited to studying robustness and ID/OOD generalization in addition to continual adaptation.

A.1 Tasks and Embodiment

We selected two structurally similar tasks: *Click Bell* and *Click Alarm Clock*. Both require the robot to reach a target object and apply a pressing action; their structural similarity makes them suitable for studying transfer learning, catastrophic forgetting, and parameter merging. Experiments use the Aloha AgileX dual-arm robot (14 DoF). Each task has 50 expert trajectories collected in a clean environment (white background, no clutter), defining the ID setting. The OOD setting introduces cluttered tables and varying backgrounds (Figure 3), substantially increasing difficulty.

Figure 3: Example of In-Distribution (ID, left) and Out-of-Distribution (OOD, right) RoboTwin scenes. The OOD condition introduces cluttered tables and varied backgrounds.

A.2 Dataset Conversion Pipeline

Since RoboTwin is not natively compatible with the LeRobot format, we developed a conversion pipeline that downloads RoboTwin trajectories, extracts RGB observations, robot states, and actions, re-formats trajectories into LeRobotDataset v3.0, and stores results locally or uploads them to the Hugging Face Hub. The same SmolVLA training pipeline is then used as for LIBERO, with the action dimension increased from 7 (single-arm) to 14 (dual-arm).

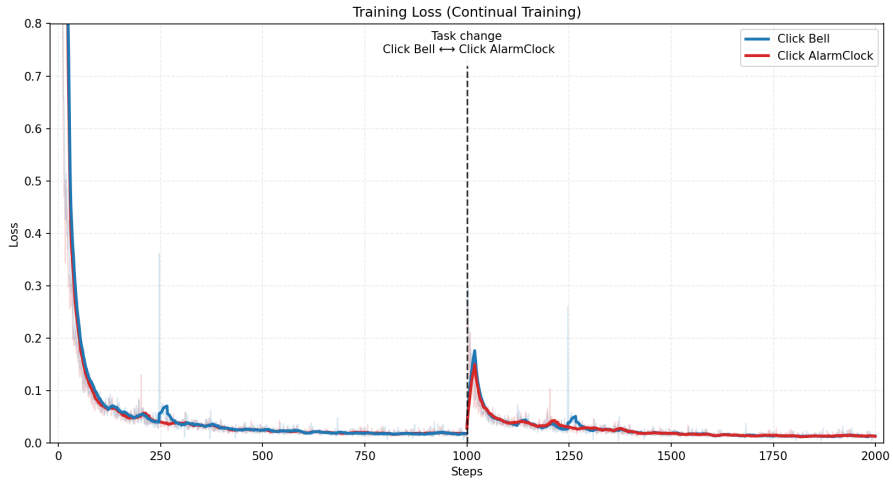
A.3 Continual-Training Protocol

To study catastrophic forgetting in this setting, we adopt a sequential fine-tuning protocol. The model is first trained on a source task for 1,000 steps, then continued on the target task in increments of 250 steps, with intermediate checkpoints saved to analyze performance transitions. At the task switch boundary, the loss exhibits a sharp increase due to the mismatch between previously-learned representations and the new task distribution; subsequent training shows slower convergence than the initial phase, indicating that the model must partially overwrite existing representations while adapting to new data.

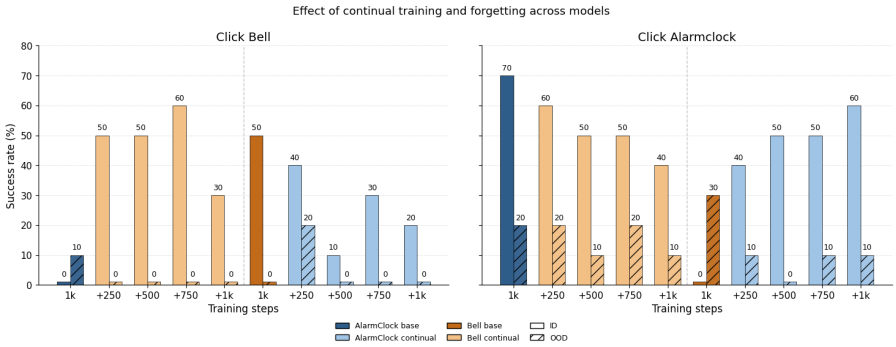
A.4 Results

Each model was evaluated over 10 simulation runs per setting (compute-bound). Figure 4 jointly visualizes loss evolution and the corresponding success-rate impact across tasks. Three observations emerge. (i) Single-task models perform strongly on their own task but poorly on the other, confirming highly specialized policies. (ii) Early continual-training stages (e.g., 250 steps) sometimes improve performance on the new task without fully destroying the original task, but as training continues, the original task's success rate declines significantly — catastrophic forgetting in the same form observed on LIBERO in Section 4.3. (iii) OOD success is consistently low across all models: cluttered backgrounds and visual perturbations substantially reduce success rates, echoing prior findings that VLA models can suffer severe drops under visual distribution shift. The qualitative

pattern is consistent with the LIBERO study: continued fine-tuning preserves recent skills at the expense of earlier ones.



(a) Training loss across the source → target task switch.



(b) Per-task success rate as continual training progresses.

Figure 4: RoboTwin continual-learning behavior of SmolVLA. The loss spike at the task switch and the subsequent decline of source-task success illustrate catastrophic forgetting in a bimanual setting.

B FULL PAIR-MERGE SWEEP

Table 6 reports the full per-cell pair-merge sweep referenced in Section 4.3: 6 strategies $\times$ 3 pairs $\times$ 2 adapter sets $=$ 36 cells, plus 6 plain-averaging cells from the standard $r = 64$ regime. Bold values mark the seven non-zero cells summarized in Table 5.

Table 6: Pair-merge results: per-task success rate (%). Bold values mark non-zero cells.

		Tied-A			LoRA $r = 8$		
Strategy	Pair	T2	T4	T7	T2	T4	T7
Linear averaging	{T2, T4}	**2**	0	0	0	0	0
	{T2, T7}	**2**	0	0	**2**	0	0
	{T4, T7}	0	0	0	0	0	0
Magnitude pruning	{T2, T4}	0	0	0	0	0	0
	{T2, T7}	**8**	0	0	0	0	0
	{T4, T7}	0	0	0	0	0	0
Task arithmetic ($\alpha = 0.4$)	{T2, T4}	0	0	0	0	0	0
	{T2, T7}	**2**	0	0	0	0	0
	{T4, T7}	0	0	0	0	0	0
Task arithmetic ($\alpha = 0.7$)	{T2, T4}	0	0	0	0	0	0
	{T2, T7}	**2**	0	0	0	0	0
	{T4, T7}	0	0	0	0	0	0
TIES (TRIM_K=0.20)	{T2, T4}	0	0	0	0	0	0
	{T2, T7}	0	0	0	**4**	0	0
	{T4, T7}	0	0	0	0	0	0
DARE ($p = 0.5$)	{T2, T4}	0	0	0	0	0	0
	{T2, T7}	0	0	0	0	0	0
	{T4, T7}	0	0	0	0	0	0
Fisher	{T2, T4}	0	0	0	0	0	0
	{T2, T7}	0	0	0	0	0	0
	{T4, T7}	0	0	0	0	0	0

C SECOND-STRONGEST PAIR-MERGE TRAJECTORY

Figure 5 shows representative frames from the second-strongest non-zero pair-merge cell (LoRA $r = 8$ TIES of $\{T2, T7\}$, 4% on T2). Top row: the arm approaches the stove with gripper open, sweeps past the knob, and the burner ignites as the knob is brushed by the side of the gripper rather than turned with a deliberate grasp. Bottom row: the arm crosses to the moka pot, grasps it, and places it on the lit burner. The merge retains the coarser pick-and-place primitive of T2 while losing the finer rotational manipulation primitive.

(a) Initial (b) Approach knob (c) Gripper open (d) Over knob (e) Knob turn

(f) Burner lit (g) Crossing (h) Reaching (i) Pot grasped (j) Pot placed

Figure 5: Second-strongest non-zero pair-merge cell: LoRA $r = 8$ TIES of $\{T2, T7\}$, 4% on T2. The pick-and-place primitive of T2 survives, while the deliberate rotational grasp-and-turn primitive does not.

D FFT Expert Checkpoint Selection

For full fine-tuned experts, the best-validation checkpoint was selected by sweeping local rollouts on the corresponding task. Figure 6 shows per-checkpoint success rates for the FFT experts of T2, T4, T7. The selected best-val steps used in Table 1 are the peaks of these curves (T2: step 3500, T4: step 3500, T7: step 6000). Selecting on best-val rather than the final checkpoint was important because late-training instability sometimes degraded performance below earlier peaks.

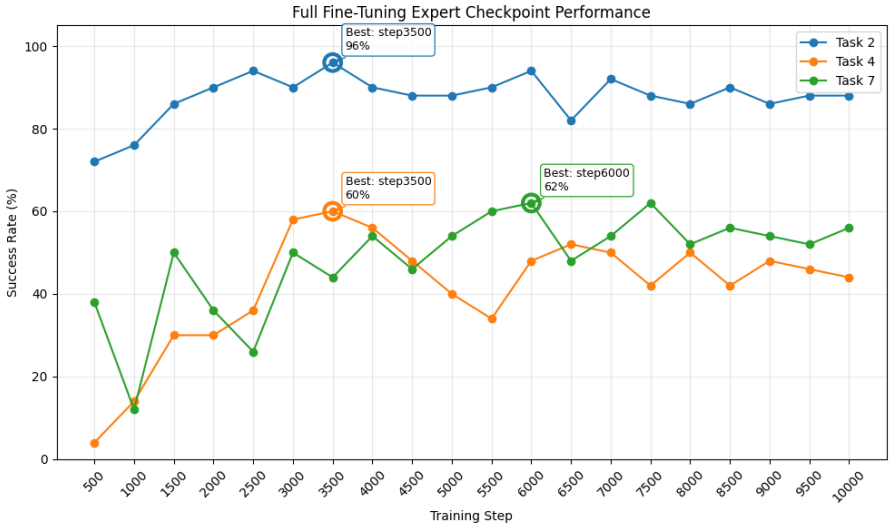


Figure 6: Full fine-tuned expert checkpoint performance across training steps for T2, T4, T7.

E Hyperparameters

Table 7: Training hyperparameters for SmolVLA fine-tuning on LIBERO and RoboTwin.

Parameter	RoboTwin	LIBERO
Policy base	SmolVLA-Pretrained	SmolVLA-LIBERO90
Action dimension	14	7
Batch size	64	64
Training steps	1,000	10,000
Action horizon	10	10
Action chunk size	50	50
Optimizer	Adam	Adam

Merging hyperparameters. Task arithmetic uses $\lambda = 0.3$ for triple merges and $\alpha \in \{0.4, 0.7\}$ for pair merges. DARE uses drop rate $p = 0.5$. TIES uses trim threshold $k = 0.20$. Fisher information is estimated using held-out demonstrations (50 trajectories per task).

References

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL `https://arxiv.org/abs/2410.24164`.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL https://arxiv.org/abs/2307.15818.

Tianxing Chen, Yao Mu, Zhixuan Liang, Zanxin Chen, Shijia Peng, Qiaojun Yu, Yude Zou, Mingkun Xu, Ruizhen Zhang, Mingyu Wang, 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. URL https://arxiv.org/abs/2506.18088.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2212.04089.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/abs/2406.09246. arXiv:2406.09246.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.03310.

Michael S. Matena and Colin Raffel. Merging models with Fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2111.09832.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2409.02920. arXiv:2409.02920.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. URL https://arxiv.org/abs/2405.12213.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023. URL https://arxiv.org/abs/2310.08864.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. URL https://arxiv.org/abs/2506.01844.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2203.05482.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2306.01708.

Yajat Yadav, Zhiyuan Zhou, Andrew Wagenmaker, Karl Pertsch, and Sergey Levine. Robust finetuning of vision-language-action robot policies via parameter merging. *arXiv preprint arXiv:2512.08333*, 2025.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2311.03099. arXiv:2311.03099.