PRESERVE WHAT MATTERS: COUNTERFACTUAL TASK-VECTOR DECOMPOSITION FOR REPLAY-FREE CONTINUAL VISION-LANGUAGE-ACTION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models have demonstrated strong capabilities in language-conditioned robotic manipulation. Yet operating in open-ended environments requires continual acquisition of new skills, which can severely overwrite previously learned behaviors and lead to catastrophic forgetting. Existing replay-based methods mitigate forgetting by revisiting stored demonstrations, incurring persistent storage and repeated training costs, while replay-free approaches typically preserve prior capabilities through allocating additional task-specific modules, leading to growing model capacity as new skills accumulate. Inspired by task arithmetic, we view the parameter change induced by downstream fine-tuning as a proxy for newly acquired capability. This recasts implicit skill retention as explicit parameter-space protection, avoiding both historical demonstration replay and task-specific model expansion. Building on this view, we further identify that not every direction in a historical task vector deserves equal preservation: transferable task functionality can be entangled with environment-specific correlations, so indiscriminately protecting the full update may unnecessarily restrict future adaptation. We therefore introduce CSD-VLA, a replay-free framework that uses task-preserving counterfactual interventions to identify an environment-sensitive subspace, decomposes each historical task vector into environment-sensitive and counterfactually invariant components, and selectively protects only the latter. Experiments on OpenVLA-OFT-7B and QwenOFT-3B across LIBERO and RoboTwin 2.0 demonstrate that CSD-VLA outperforms representative replay-based and replay-free baselines in overall continual-learning performance, strengthens retention under both task-level and cross-capability shifts, and better preserves historical skills under unseen visual changes, without replaying historical demonstrations or expanding the policy architecture.

1 INTRODUCTION

Vision-Language-Action (VLA) models have demonstrated strong capabilities in language-conditioned robotic manipulation by integrating visual perception, language understanding, and action generation within a unified policy (Kim et al., 2024; Black et al., 2024). Nevertheless, adapting a pretrained VLA to new manipulation tasks and environments still commonly requires downstream fine-tuning. In long-lived robotic systems, such adaptation is inherently continual: the same policy must acquire new skills while retaining previously learned ones. Naively fine-tuning the model on each incoming task can interfere with earlier behaviors, creating a fundamental tension between historical-skill retention and plasticity for future adaptation.

Existing continual VLA methods mainly preserve past skills through historical data or additional model capacity. Replay-based approaches store and revisit previous demonstrations; PHASER, for example, improves replay by allocating memory according to manipulation phases and interference risk (Chen et al., 2026). Despite their effectiveness, these methods require persistent demonstration storage and repeated training on historical data. Representative replay-free methods instead isolate skills in task- or skill-specific components: CLARE dynamically expands lightweight adapters, AtomicVLA constructs an expandable skill-expert library, and CORAL assigns a LoRA expert to each task (Römer et al., 2026; Zhang et al., 2026; Luo et al., 2026). These designs avoid demon-

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

stration replay, but continually introduce additional modules or parameters as new tasks and skills are added. This raises a natural question: can a shared VLA preserve historical capabilities using a representation already produced by the learning process itself?

Inspired by task arithmetic, we treat the parameter change induced by downstream fine-tuning as a proxy for newly acquired capability. Task Arithmetic showed that the difference between pretrained and fine-tuned parameters forms a functionally meaningful direction in weight space (Ilharco et al., 2023). Accordingly, we represent the capability acquired from task t by its task vector $\tau_t = \phi_t - \phi_{t-1}$. This view recasts implicit skill preservation as explicit parameter-space protection, avoiding both historical demonstration replay and task-specific model expansion.

However, a historical task vector should not be preserved indiscriminately. Because a task vector is induced by fine-tuning on a finite set of demonstrations collected in a particular environment, it may encode task-relevant functionality together with incidental correlations involving backgrounds, textures, or other visual conditions. Prior studies show that robot policies can rely on task-irrelevant visual features and that VLAs can exhibit counterfactual failures driven by visual shortcuts (Xing et al., 2025; Fang et al., 2026). These observations suggest that directions within a historical task vector can have different preservation value: environment-sensitive directions should remain adaptable, because constraining them may prevent the model from correcting environment-specific correlations under domain shifts and reduce the parameter-space freedom needed to learn new tasks.

Motivated by this insight, we introduce **CSD-VLA**, a replay-free continual learning framework based on counterfactual selective preservation. After learning a task, CSD-VLA combines its low-rank historical update space with task-preserving counterfactual interventions to identify directions that are sensitive to environmental changes. It then decomposes the historical task vector into an environment-sensitive component and a counterfactually invariant component. During subsequent adaptation, CSD-VLA selectively protects only the invariant component while leaving the environment-sensitive complement available for revision.

We evaluate CSD-VLA with OpenVLA-OFT-7B and QwenOFT-3B on LIBERO (Liu et al., 2023) and RoboTwin 2.0 (Chen et al., 2025), covering both task-level and cross-capability continual learning. CSD-VLA consistently improves task- and capability-level retention while preserving new-task plasticity, and further strengthens retention under unseen visual changes, without historical replay or task-specific architectural expansion.

Our contributions are:

- We formulate acquired VLA capabilities as fine-tuning-induced *task vectors*, enabling historical knowledge to be explicitly preserved in parameter space without replay or task-specific model expansion.
- We introduce *counterfactual selective preservation*, which identifies environment-sensitive directions within historical updates and protects only their counterfactually invariant components during future adaptation.
- We validate CSD-VLA across two VLA backbones on LIBERO and RoboTwin 2.0, demonstrating improved task- and capability-level retention while preserving new-task plasticity, together with stronger retention under unseen visual changes.

2 Related Work

2.1 Continual Learning for VLA Policies

Continual VLA studies show that sequential adaptation can cause forgetting, with replay remaining a strong baseline in both simulated and real-world settings (Liu et al., 2026; Zhu et al., 2026). PHASER improves replay through phase-aware memory allocation and interference-based sampling (Chen et al., 2026), whereas replay-free methods such as CLARE, AtomicVLA, and CORAL allocate expandable adapters or experts (Römer et al., 2026; Zhang et al., 2026; Luo et al., 2026). Related LoRA-based methods also reduce interference during parameter-efficient continual learning (Liang & Li, 2024; He et al., 2025; Wei et al., 2025). However, these approaches require either historical data or additional task-dependent components. CSD-VLA instead preserves historical capabilities in one shared parameterization without replay or task-specific expansion.

2.2 Parameter-Space Representations and Preservation

Task Arithmetic showed that fine-tuning updates encode functionally meaningful directions in weight space (Ilharco et al., 2023). Subsequent methods improve their composition through sign resolution, sparsification, adaptive coefficients, or parameter-importance weighting (Yadav et al., 2023; Yu et al., 2024; Yang et al., 2024; Matena & Raffel, 2021). Related VLA methods manipulate parameter-space representations to retain or transfer capabilities (Yadav et al., 2025; Song et al., 2026; Kang et al., 2026), while continual-merging methods exploit task-vector geometry to reduce interference (Tang et al., 2025; Yang et al., 2025; Qiu et al., 2026). Unlike methods that treat each update as a single unit, CSD-VLA identifies which directions *within* a historical task vector warrant protection during future sequential adaptation.

2.3 Visual Shortcuts and Counterfactual Invariance

Visual randomization and counterfactual augmentation improve robotic generalization by weakening environment-specific correlations (Tobin et al., 2017; Armengol Urpí et al., 2024; Ameperosa et al., 2025). VLA evaluations nevertheless reveal reliance on task-irrelevant visual cues and failures under controlled environmental or counterfactual changes (Xing et al., 2025; Fei et al., 2026; Fang et al., 2026). Such correlations are especially problematic in continual learning because protecting them may restrict later adaptation and propagate spurious dependencies across tasks (Lee et al., 2024). Rather than augmenting policy training data, CSD-VLA uses task-preserving counterfactuals to identify environment-sensitive task-vector directions and exclude them from future protection.

3 Method

CSD-VLA operates at task boundaries. As shown in Fig. 1(a), after learning a task, it uses controlled visual counterfactuals to identify parameter directions whose action-learning signal changes with the environment. It then separates the task's parameter update into an environment-sensitive component and an orthogonal remainder. Only the remainder contributes to the protection constraint used when learning subsequent tasks.

3.1 Continual Adaptation and Historical Task Vectors

We consider a sequence of manipulation tasks, each providing demonstrations of visual observations, robot states, language instructions, and target actions. The VLA retains a fixed architecture and a shared set of trainable parameters throughout the sequence. In parameter-efficient implementations, we continually update the same adapter rather than creating an independent adapter for each task. As illustrated in Fig. 1(a), after learning task t, we record its induced parameter displacement as the historical task vector:

$$\tau_t = \phi_t - \phi_{t-1}. \tag{1}$$

where t indexes the current task, $\phi_{t-1} \in \mathbb{R}^d$ and $\phi_t \in \mathbb{R}^d$ are the shared trainable parameters immediately before and after learning that task, d is the number of trainable parameter coordinates, and τ_t is the resulting parameter change. All checkpoints and gradients are expressed in the same continuously maintained parameter coordinates. We use τ_t as a proxy for acquired capability, without assuming that every part of the update contributes equally to task performance.

3.2 Identifying an Environment-Sensitive Subspace

The counterfactual direction-identification procedure is illustrated in Fig. 1(b). At the end of task t, we select a small probe set from its current demonstrations, sampling states across the progress of each manipulation trajectory. For each probe state, we retain the robot and object configuration, camera geometry, instruction, and target action, but render alternative visual environments. We use $K = 6$ environment variants, consisting of two illumination variants, two render-only support-surface texture variants, and two distal background variants. These interventions are restricted to visual properties that do not alter the demonstrated action. Each variant specifies one fixed rendering intervention that is applied consistently to every state in the probe set.

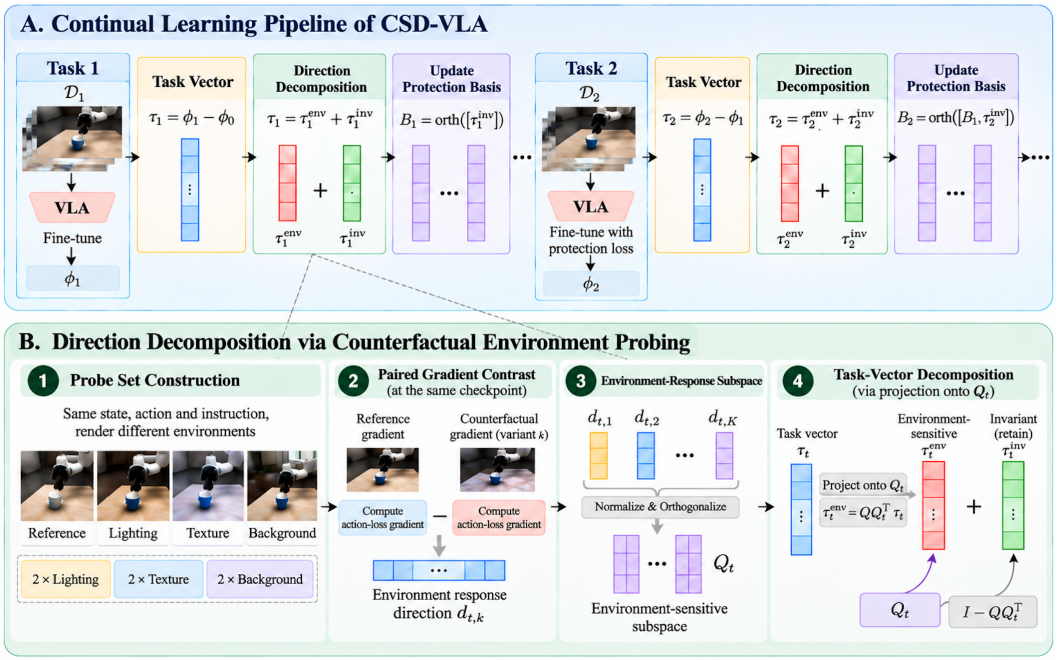


Figure 1: **CSD-VLA pipeline overview.** At each task boundary, the fine-tuning-induced parameter change is recorded as a historical task vector. Controlled counterfactual environments produce paired gradient contrasts whose span defines an environment-response subspace Q_t. Projecting the task vector onto Q_t separates its environment-sensitive component from the invariant remainder. During subsequent task learning, only the accumulated invariant directions are protected, while environment-sensitive directions remain plastic for adaptation.

For each variant, we compare its action-loss gradient with that of the original observation at the same post-task checkpoint:

$$d_{t,k} = \frac{1}{N_t} \sum_{i=1}^{N_t} \Big[\nabla_\phi \ell_{\text{act}}(\phi_t; x_i^{(k)}, a_i) - \nabla_\phi \ell_{\text{act}}(\phi_t; x_i^{(0)}, a_i) \Big], \qquad k = 1, \ldots, K. \tag{2}$$

where N_t is the number of probe states for task t, i indexes those states, K is the number of environment variants, $x_i^{(0)}$ contains the original observation, robot state, and instruction, $x_i^{(k)}$ contains the corresponding observation under variant k with the same robot state and instruction, a_i is their shared target action or action chunk, ℓ_{act} is the VLA's per-example action-training loss, and $d_{t,k}$ is the mean paired gradient contrast for variant k. Every gradient is evaluated at the fixed parameter point ϕ_t; probing performs no optimizer step. For stochastic action losses, paired evaluations use the same sampled noise and time step so that their difference reflects the visual intervention rather than loss randomness.

For each environment variant, we average the paired gradient contrasts across probe states to estimate its intervention-specific direction while suppressing state-dependent noise. In contrast, different variants capture distinct modes of environmental sensitivity, we therefore keep the resulting K directions separate and use their span to form the environment-response subspace, rather than averaging them into a single direction. Specifically, after discarding numerically negligible contrasts, we normalize and orthogonalize the remaining directions:

$$Q_t = \text{orth}\big([\hat{d}_{t,1}, \ldots, \hat{d}_{t,K}]\big), \qquad \hat{d}_{t,k} = \frac{d_{t,k}}{\|d_{t,k}\|_2}. \tag{3}$$

where $\hat{d}_{t,k}$ is a non-negligible contrast normalized to unit length, $\|\cdot\|_2$ is the Euclidean norm, $\text{orth}(\cdot)$ denotes rank-revealing orthogonalization that removes redundant directions, and the columns of

$Q_t \in \mathbb{R}^{d\times\rho_t}$ form an orthonormal basis with rank $\rho_t \le K$. This basis describes the local parameter directions whose action-learning signal responds to the tested environment changes.

3.3 Task-Vector Decomposition and Selective Preservation

We project the historical update onto the identified environment-response subspace and assign its orthogonal remainder to the preservation component:

$$\tau_t^{\text{env}} = Q_t Q_t^\top \tau_t, \qquad \tau_t^{\text{inv}} = \tau_t - \tau_t^{\text{env}}. \tag{4}$$

where τ_t^{env} is the component of the task vector aligned with the tested environment responses, τ_t^{inv} is its counterfactually invariant remainder, and $Q_t Q_t^\top$ is the orthogonal projector onto the span of Q_t. The two components sum exactly to τ_t. Here, "invariant" has a local operational meaning: τ_t^{inv} is orthogonal to the responses elicited by our selected interventions at ϕ_t. We do not assume that it is a uniquely identifiable semantic skill vector, or that every direction in τ_t^{env} is a harmful shortcut. The environment-sensitive component is not removed from the model; it is simply omitted from this task's explicit protection constraint.

We maintain an orthonormal protection basis B_t that summarizes the invariant directions selected from all previously learned tasks. The initial basis is empty, i.e., $B_0 = \varnothing$. After task t, we first remove from τ_t^{inv} the component already covered by the historical protection basis:

$$r_t = \left(I - B_{t-1}B_{t-1}^\top\right)\tau_t^{\text{inv}}. \tag{5}$$

Here, r_t represents the novel preservation direction contributed by task t after excluding directions already protected by earlier tasks. If $\|r_t\|_2$ is larger than a small numerical threshold ϵ_B, we normalize the residual and append it to the existing basis:

$$b_t = \frac{r_t}{\|r_t\|_2}, \qquad B_t = [B_{t-1},\, b_t]\,. \tag{6}$$

Otherwise, when $\|r_t\|_2 \le \epsilon_B$, the new invariant vector is considered redundant with the existing protected span and we simply set

$$B_t = B_{t-1}. \tag{7}$$

Since B_{t-1} is orthonormal and r_t is constructed in its orthogonal complement, the updated basis satisfies $B_t^\top B_t = I$. Equivalently, this incremental update corresponds to orthogonalizing $[B_{t-1}, \tau_t^{\text{inv}}]$, while avoiding the need to recompute the entire basis from scratch. When task $t+1$ arrives, training starts from ϕ_t and minimizes

$$\mathcal{J}_{t+1}(\phi) = \mathcal{L}_{\text{act}}(\phi; \mathcal{D}_{t+1}) + \frac{\lambda}{2}\left\|B_t^\top(\phi - \phi_t)\right\|_2^2. \tag{8}$$

where ϕ denotes the parameters being optimized, $\mathcal{D}_{t+1}$ is the new task's demonstration set, $\mathcal{L}_{\text{act}}$ is its empirical action-training loss, $\lambda \ge 0$ controls preservation strength, B_t contains the historical invariant directions selected through task t, and ϕ_t is the checkpoint used as the new task's anchor. The second term softly penalizes movement along the aggregate protected span while leaving its orthogonal complement available for adaptation.

After updating B_t, we discard the task's probe observations and gradient contrasts. Later tasks use only their own demonstrations and the stored protection basis. This basis participates in training but adds no task-specific module to the deployed policy. CSD-VLA therefore avoids historical demonstration replay and inference-time model expansion, while requiring only a one-time counterfactual probing stage at the end of each task and a lightweight subspace regularization during subsequent training.

4 Experiment

4.1 Experimental Setup

Environments and Tasks. We evaluate CSD-VLA on two robotic manipulation benchmarks, LIBERO Liu et al. (2023) and RoboTwin 2.0 Chen et al. (2025). Following PHASER Chen et al.

(2026), our main continual-learning evaluation uses LIBERO-Goal and LIBERO-Long, each containing 10 tasks learned sequentially. LIBERO-Goal consists of tasks with distinct manipulation objectives in a shared workspace, while LIBERO-Long contains compositional multi-stage tasks with longer action sequences. We follow the task ordering and per-task training protocol of PHASER to enable direct comparison.

We further evaluate on RoboTwin 2.0 Chen et al. (2025) under task-level and cross-capability continual-learning settings. The task-level setting uses a fixed 10-task RoboTwin-Single-Arm sequence, whereas the cross-capability setting spans Single-Arm, Coordinated Bimanual, and Independent Bimanual manipulation to evaluate retention across progressively different control structures (detailed in Section 4.3). All methods use the same task order, demonstrations, and per-task training budget, and are evaluated with 100 rollouts per task.

Counterfactual Probe Configuration. For each task, we sample $N_t = 128$ probe states approximately uniformly over trajectory progress and compute probe gradients in mini-batches of 16. We use the $K = 6$ counterfactual variants from Section 3 (two lighting, two texture, and two background interventions). All gradients are evaluated at the fixed post-task checkpoint without optimizer updates.

VLA Backbones. We evaluate OpenVLA-OFT-7B Kim et al. (2025) and QwenOFT-3B Chen et al. (2026), which differ in model scale and action-generation architecture. Following PHASER, both use rank-32 LoRA with a continuously maintained parameterization across tasks, and we adopt the same LIBERO task order and per-task training budget.

Baselines. We compare against Sequential FT, Experience Replay (ER) Rolnick et al. (2019), MIR Aljundi et al. (2019), iCaRL Rebuffi et al. (2017), PHASER Chen et al. (2026), and CLARE Römer et al. (2026). Sequential FT provides the replay-free fine-tuning baseline; ER, MIR, iCaRL, and PHASER cover representative replay-based strategies. CLARE represents replay-free model expansion through adaptive adapters and routing. As its original experiments use a different VLA architecture, we port CLARE to OpenVLA-OFT-7B and QwenOFT-3B under our unified training and evaluation protocol. For replay-based methods, we follow the replay budgets of PHASER for each backbone and suite, while CLARE and CSD-VLA use no historical demonstrations.

Evaluation Metrics. We report Final Average Success Rate (ASR) and Negative Backward Transfer (NBT) as the primary metrics, measuring final performance and forgetting, respectively. Lower NBT indicates better retention. Following PHASER, LIBERO results use 50 rollouts per task. We additionally analyze new-task plasticity, environment-robust retention, and efficiency.

4.2 Main Results on LIBERO

Table 1 compares continual-learning performance across two VLA backbones and two LIBERO suites. Following PHASER, final ASR is evaluated with 50 rollouts per task. For replay-based methods, we use the matched per-task replay buffers of PHASER: $B = 110/180$ for OpenVLA-OFT-7B on Goal/Long and $B = 1000$ for QwenOFT-3B on both suites.

CSD-VLA achieves the highest final ASR in three of the four settings; on QwenOFT-3B/Goal, it remains within 0.5 points of PHASER while attaining lower NBT. The gains are also consistent against the replay-free CLARE baseline, with CSD-VLA improving both ASR and NBT across all backbone–suite combinations.

Notably, CSD-VLA remains competitive with or surpasses replay-based methods despite never revisiting historical demonstrations. Compared with PHASER, it improves both retention and final performance on OpenVLA-OFT-7B and QwenOFT-3B/Long, while maintaining comparable performance on QwenOFT-3B/Goal. The consistency across model scales and action heads suggests that the benefit does not depend on a particular VLA architecture. Overall, these results show that selective parameter-space preservation can surpass both replay-based and task-specific expansion approaches in overall continual-learning performance, while requiring neither historical replay nor architectural growth.

Table 1: **Continual learning across two VLA backbones and two LIBERO suites.** Results marked with † are reported by PHASER under the matched protocol; all remaining results are obtained from our own implementation.

	OpenVLA-OFT-7B				QwenOFT-3B			
	GOAL		LONG		GOAL		LONG	
Method	ASR↑	NBT↓	ASR↑	NBT↓	ASR↑	NBT↓	ASR↑	NBT↓
Sequential FT†	10.0	95.3	5.0	37.1	8.0	67.2	9.0	62.9
ER†	77.6	22.2	54.6	**-5.6**	39.4	47.8	33.0	33.3
MIR†	81.6	15.6	82.2	10.0	73.6	9.3	41.0	19.6
iCaRL†	50.2	42.0	46.4	45.0	65.0	13.0	33.0	**6.0**
PHASER†	87.8	7.8	85.8	10.0	**79.0**	6.4	51.6	12.4
CLARE	89.0	10.5	84.8	13.7	76.4	15.8	51.5	9.2
CSD-VLA (Ours)	**90.3**	**6.5**	**87.0**	8.0	78.5	**6.1**	**54.0**	7.4

4.3 MAIN RESULTS ON ROBOTWIN 2.0

We next evaluate whether CSD-VLA generalizes beyond LIBERO under both task-level and capability-level continual adaptation.

Task-Level Continual Learning. Table 2(a) reports the 10-task RoboTwin-Single-Arm sequence, testing whether the gains observed on LIBERO transfer to a different simulator and manipulation domain. CSD-VLA achieves the highest final ASR on both backbones, reaching 65.0% on OpenVLA-OFT-7B and 54.0% on QwenOFT-3B. Compared with ER, it improves ASR by 6.8 and 7.2 points while substantially reducing NBT. Against the replay-free CLARE baseline, CSD-VLA achieves higher final ASR on both backbones, with lower NBT on OpenVLA-OFT-7B and a modest NBT trade-off on QwenOFT-3B. Overall, CSD-VLA provides a favorable stability–plasticity balance without replay.

Cross-Capability Continual Learning. Table 2(b) evaluates cross-capability continual learning from Single-Arm to coordinated and independent bimanual manipulation, introducing substantial shifts in action structure while requiring previously acquired capabilities to be retained. We organize RoboTwin tasks into three capability families: Single-Arm, Coordinated Bimanual (Coord.), and Independent Bimanual (Indep.), and learn an equal number of tasks sequentially in the order Single-Arm $\rightarrow$ Coord. $\rightarrow$ Indep. Tasks are learned individually, while the grouping defines capability boundaries for retention evaluation. In addition to final ASR, we report Capability NBT (Cap-NBT), which measures forgetting of a completed capability family after subsequent families are learned. Unlike task-level NBT, Cap-NBT uses the end of each capability stage as its reference checkpoint, thereby focusing on cross-capability forgetting.

CSD-VLA exhibits substantially lower Cap-NBT and the highest final ASR on both backbones. The advantage is most pronounced on OpenVLA-OFT-7B, where it reduces Cap-NBT from 16.6 with ER to 7.7 while increasing final ASR to 57.3%. More importantly, these gains persist when the learning stream crosses control regimes—from single-arm manipulation to coordinated and independent bimanual control—rather than merely varying tasks within a fixed capability family. The stronger degradation of ER and CLARE under the same transitions suggests that effective task-level retention does not automatically translate into capability-level retention. By decomposing each historical task vector and selectively protecting its invariant component, CSD-VLA preserves the reusable structure of an acquired capability while leaving environment-sensitive directions adaptable. This compact parameter-space mechanism therefore extends preservation from individual tasks to capability families, improving the retention frontier without historical replay or task-specific architectural growth.

5 ANALYSIS

5.1 EFFECT OF SELECTIVE PRESERVATION

Experimental design. We evaluate the effect of selective preservation using OpenVLA-OFT-7B on the 10-task LIBERO-Goal sequence, following the same training and evaluation protocol as the

Table 2: **Continual-learning results on RoboTwin 2.0 across task-level and capability-level settings.**

(a) Task-level continual learning on RoboTwin-Single-Arm

		OpenVLA-OFT-7B		QwenOFT-3B	
Method	**Replay-Free**	ASR↑	NBT↓	ASR↑	NBT↓
Sequential FT	✓	38.0	35.9	29.7	33.7
ER	×	58.2	13.2	46.8	14.3
CLARE	✓	63.6	8.2	52.4	**4.5**
CSD-VLA (Ours)	✓	**65.0**	**5.5**	**54.0**	6.6

(b) Cross-capability continual learning: Single-Arm → Coord. → Indep.

		OpenVLA-OFT-7B		QwenOFT-3B	
Method	**Replay-Free**	Cap-NBT↓	Final ASR↑	Cap-NBT↓	Final ASR↑
Sequential FT	✓	38.9	35.1	35.5	28.1
ER	×	16.6	51.0	17.0	41.5
CLARE	✓	15.0	53.5	14.6	46.2
CSD-VLA (Ours)	✓	**7.7**	**57.3**	**8.5**	**47.5**

Table 3: **Selective-preservation ablation on LIBERO-Goal with OpenVLA-OFT-7B.** † denotes reference values reported by PHASER.

		New-task Acquisition		Continual Performance	
Method	Protected Vector	FWT↑	Adapt. AUC↑	NBT↓	Final ASR↑
NoProtect	None	**95.8**	**79.0**	95.3†	10.0†
FullProtect	τ_t	94.8	76.1	**5.8**	87.2
EnvProtect	τ_t^{env}	95.1	78.2	81.8	15.9
CSD-VLA	τ_t^{inv}	95.5	78.7	6.5	**90.3**

main experiment. We compare four variants: NoProtect, which applies no historical preservation; FullProtect, which protects the complete task vector τ_t; EnvProtect, which protects only τ_t^{env}; and CSD-VLA, which protects only τ_t^{inv}. All protection variants use the same regularization strength and basis-update rule. In addition to final ASR and NBT, we report FWT to measure new-task performance at the end of each adaptation stage and Adaptation AUC to capture learning efficiency throughout current-task training.

Stability–plasticity pattern analysis. Table 3 suggests that selective preservation retains nearly the same new-task plasticity as unconstrained fine-tuning. CSD-VLA differs from NoProtect by only 0.3 points in both FWT and Adaptation AUC, while reducing NBT from 95.3 to 6.5, indicating that protecting the selected invariant directions introduces little interference with subsequent learning.

Compared with FullProtect, CSD-VLA improves FWT and Adaptation AUC while incurring only a small increase in NBT (6.5 vs. 5.8), suggesting that protecting the full historical update unnecessarily constrains adaptation. EnvProtect provides much weaker retention despite similar acquisition, further indicating that the key is not how much of the historical update is protected, but *which* directions are preserved. We hypothesize that FullProtect's slight retention advantage partly reflects environment-specific fitting, which we examine under held-out visual conditions in Section 5.2. Overall, CSD-VLA achieves a favorable stability–plasticity balance, retaining previously learned capabilities while maintaining nearly the same new-task acquisition performance as unconstrained fine-tuning.

5.2 Environment-Robust Retention

Held-out evaluation. We further compare FullProtect and CSD-VLA under unseen visual changes using the same OpenVLA-OFT-7B checkpoints from the LIBERO-Goal decomposition study. We construct four held-out categories: Lighting, support-surface Texture, Background, and Composition, with two variants per category. The Composition variants jointly alter lighting, texture, and background. Except for the visual factors being varied, all other evaluation conditions, including

Table 4: **Environment-robust retention on LIBERO-Goal with OpenVLA-OFT-7B.**

	Lighting		Texture		Background		Composition	
Method	ASR↑	NBT↓	ASR↑	NBT↓	ASR↑	NBT↓	ASR↑	NBT↓
FullProtect	64.6	14.2	60.4	11.1	76.8	8.3	68.9	11.0
CSD-VLA	**71.1**	**9.4**	**69.9**	**7.6**	**80.6**	**6.8**	**75.6**	**8.8**

Table 5: **CSD-VLA training overhead on LIBERO with OpenVLA-OFT-7B.** Additional FLOPs introduced by each component over the complete 10-task sequential training sequence. Percentages in parentheses are relative to Sequential FT under the same backbone and optimization budget.

CF Probe	Vector Decomp.	Basis Update	Protection	**Total**
$148.18\,\mathrm{P}$ (1.008%)	$0.457\,\mathrm{T}$ $(< 0.001\%)$	$0.0879\,\mathrm{T}$ $(< 0.001\%)$	$2.21\,\mathrm{P}$ (0.015%)	$\mathbf{150.39}\,\mathrm{P}$ $(\mathbf{1.023}\%)$

task definitions, camera setup, and rollout budgets, are kept identical between FullProtect and CSD-VLA, so that the evaluation differs only in the held-out visual conditions.

For each visual category, we evaluate every task both immediately after its acquisition and at the final checkpoint. We report OOD ASR, which measures the final average success rate under the corresponding held-out visual shift, and OOD NBT, which measures the degradation from task acquisition to the final checkpoint under the same OOD condition.

Robust-retention pattern analysis. As shown in Table 4, CSD-VLA consistently outperforms FullProtect under all four held-out visual shifts, achieving both higher OOD ASR and lower OOD NBT. Averaged across lighting, texture, background, and composition changes, CSD-VLA improves OOD ASR by 6.6 points and reduces OOD NBT by 3.0 points.

This result contrasts with the ID setting in Table 3, where FullProtect attains slightly lower NBT than CSD-VLA (5.8 vs. 6.5). The reversal suggests that the additional ID retention of FullProtect partly comes from preserving environment-specific correlations that fit the original visual conditions but transfer poorly to unseen environments. By protecting the full task vector, these environment-sensitive directions are also constrained during later adaptation, leading to larger degradation of cross-environment performance. CSD-VLA instead leaves them revisable and selectively protects the invariant component, resulting in more robust retention of historical capabilities under visual shifts.

5.3 Training Efficiency

CSD-VLA requires neither historical replay data nor task-specific model expansion, and leaves the deployed policy unchanged, introducing no additional inference FLOPs. As shown in Table 5, over the complete 10-task LIBERO sequence, CSD-VLA adds only 1.023% training FLOPs relative to standard sequential fine-tuning. Nearly all of this overhead comes from the task-boundary counterfactual gradient probes (1.008%). In contrast, the subsequent task-vector decomposition and protection-basis update consist only of low-rank parameter-space operations and each contribute less than 0.001%, while the per-step protection constraint adds only 0.015%. These results show that selective parameter-space protection introduces only a small computational overhead during continual adaptation.

6 Conclusion

We introduced CSD-VLA, a replay-free continual-learning framework that represents acquired capabilities through fine-tuning-induced task vectors and selectively preserves their counterfactually invariant components. By separating environment-sensitive historical directions from those worth protecting, CSD-VLA improves task- and capability-level retention while maintaining strong new-task plasticity across LIBERO and RoboTwin 2.0. It further provides more robust retention under unseen visual shifts, while requiring no historical replay, task-specific model expansion, or additional inference cost. These results suggest that effective continual VLA learning depends not only on how strongly historical knowledge is preserved, but on identifying which parts of that knowledge should remain protected.

AI USE STATEMENT

In this work, we used generative AI tools to review the manuscript for clarity and internal consistency, including the presentation of the methodology and experimental results, and to polish its language.

REFERENCES

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1908.04742.

Ezra Ameperosa, Jeremy A. Collins, Mrinal Jain, and Animesh Garg. Rocoda: Counterfactual data augmentation for data-efficient robot learning from demonstrations. In *IEEE International Conference on Robotics and Automation*, pp. 13250–13256, 2025. URL https://arxiv.org/abs/2411.16959.

Núria Armengol Urpí, Marco Bagatella, Marin Vlastelica, and Georg Martius. Causal action influence aware counterfactual data augmentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1709–1729. PMLR, 2024. URL https://proceedings.mlr.press/v235/armengol-urpi-24a.html.

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. doi: 10.48550/arXiv.2410.24164. URL https://arxiv.org/abs/2410.24164.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. doi: 10.48550/arXiv.2506.18088. URL https://arxiv.org/abs/2506.18088.

Ziyang Chen, Shaoguang Wang, Weiyu Guo, Qianyi Cai, He Zhang, Pengteng Li, Yiren Zhao, and Yandong Guo. Phaser: Phase-aware and semantic experience replay for vision-language-action models. *arXiv preprint arXiv:2606.03598*, 2026. doi: 10.48550/arXiv.2606.03598. URL https://arxiv.org/abs/2606.03598.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in vlas. *arXiv preprint arXiv:2602.17659*, 2026. doi: 10.48550/arXiv.2602.17659. URL https://arxiv.org/abs/2602.17659.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: A progressive robustness benchmark for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2510.13626.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30534–30544, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/He_CL-LoRA_Continual_Low-Rank_Adaptation_for_Rehearsal-Free_Class-Incremental_Learning_CVPR_2025_paper.html.

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*, 2023. URL https://arxiv.org/abs/2212.04089.

Taewook Kang, Taeheon Kim, Donghyun Shin, and Jonghyun Choi. Domain arithmetic: One-shot vla adaptation under environmental shifts. *arXiv preprint arXiv:2607.00666*, 2026. doi: 10.48550/arXiv.2607.00666. URL https://arxiv.org/abs/2607.00666.

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. *arXiv preprint arXiv:2406.09246*, 2024. doi: 10.48550/arXiv.2406.09246. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025. doi: 10.48550/arXiv.2502.19645. URL https://arxiv.org/abs/2502.19645.

Donggyu Lee, Sangwon Jung, and Taesup Moon. Continual learning in the presence of spurious correlation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3Y7r6xueJJ.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liang_InfLoRA_Interference-Free_Low-Rank_Adaptation_for_Continual_Learning_CVPR_2024_paper.html.

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*, volume 36, 2023. URL https://arxiv.org/abs/2306.03310.

Huihan Liu, Changyeon Kim, Bo Liu, Minghuan Liu, and Yuke Zhu. Pretrained vision-language-action models are surprisingly resistant to forgetting in continual learning. *arXiv preprint arXiv:2603.03818*, 2026. doi: 10.48550/arXiv.2603.03818. URL https://arxiv.org/abs/2603.03818.

Yuankai Luo, Woping Chen, Tong Liang, and Zhenguo Li. Coral: Scalable multi-task robot learning via lora experts. *arXiv preprint arXiv:2603.09298*, 2026. doi: 10.48550/arXiv.2603.09298. URL https://arxiv.org/abs/2603.09298.

Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging. *arXiv preprint arXiv:2111.09832*, 2021. URL https://arxiv.org/abs/2111.09832.

Zihuan Qiu, Lei Wang, Yang Cao, Runtong Zhang, Bing Su, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Null-space filtering for data-free continual model merging: Preserving stability, promoting plasticity. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.21413.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017. doi: 10.1109/CVPR.2017.587.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1811.11682.

Ralf Römer, Yi Zhang, Yuming Li, and Angela P. Schoellig. Clare: Continual learning for vision-language-action models via autonomous adapter routing and expansion. *arXiv preprint arXiv:2601.09512*, 2026. doi: 10.48550/arXiv.2601.09512. URL https://arxiv.org/abs/2601.09512.

Wenxuan Song, Han Zhao, Fuhao Li, Ziyang Zhou, Xi Wang, Jing Lyu, Pengxiang Ding, Yan Wang, Donglin Wang, and Haoang Li. Capvector: Learning transferable capability vectors in parametric space for vision-language-action models. *arXiv preprint arXiv:2605.10903*, 2026. doi: 10.48550/arXiv.2605.10903. URL https://arxiv.org/abs/2605.10903.

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. Merging models on the fly without retraining: A sequential approach to scalable continual model merging. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2501.09522.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23–30. IEEE, 2017. doi: 10.1109/IROS.2017.8202133.

Xiwen Wei, Guihong Li, and Radu Marculescu. Online-lora: Task-free online continual learning via low rank adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6634–6645, 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Wei_Online-LoRA_Task-Free_Online_Continual_Learning_via_Low_Rank_Adaptation_WACV_2025_paper.html.

Youguang Xing, Xu Luo, Junlin Xie, Lianli Gao, Hengtao Shen, and Jingkuan Song. Shortcut learning in generalist robot policies: The role of dataset diversity and fragmentation. In *Conference on Robot Learning*, 2025. URL https://arxiv.org/abs/2508.06426.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

Yajat Yadav, Zhiyuan Zhou, Andrew Wagenmaker, Karl Pertsch, and Sergey Levine. Robust fine-tuning of vision-language-action robot policies via parameter merging. *arXiv preprint arXiv:2512.08333*, 2025. doi: 10.48550/arXiv.2512.08333. URL https://arxiv.org/abs/2512.08333.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nZP6NgD3QY.

Enneng Yang, Anke Tang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, and Jie M. Zhang. Continual model merging without data: Dual projections for balancing stability and plasticity. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/37d9f19150fce07bced2a81fc87d47a6-Abstract-Conference.html.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Likui Zhang, Tao Tang, Zhihao Zhan, Xiuwei Chen, Zisheng Chen, Jianhua Han, Jiangtong Zhu, Pei Xu, Hang Xu, Hefeng Wu, Liang Lin, and Xiaodan Liang. Atomicvla: Unlocking the potential of atomic skill learning in robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.07648.

Jiarun Zhu, Yijun Hong, Xiaoquan Sun, Zetian Xu, Qijun He, Haijier Chen, Zhiyong Wang, Mingqi Yuan, Wenjun Zeng, and Jiayu Chen. Can vision-language-action models learn from real-world data continually without forgetting? *arXiv preprint arXiv:2605.26820*, 2026. URL https://arxiv.org/abs/2605.26820.