# Simple Recipe Works: Vision-Language-Action Models are Natural Continual Learners with Reinforcement Learning

*Abstract*—**Continual Reinforcement Learning (CRL) for Vision-Language-Action (VLA) models is a promising direction toward self-improving embodied agents that can adapt in open-ended, evolving environments. However, conventional wisdom from continual learning suggests that naive Sequential Fine-Tuning (Seq. FT) leads to catastrophic forgetting, necessitating complex CRL strategies. In this work, we take a step back and conduct a systematic study of CRL for large pretrained VLAs across three models and five challenging lifelong RL benchmarks. We find that, contrary to established belief, simple Seq. FT with low-rank adaptation (LoRA) is remarkably strong: it achieves high plasticity, exhibits little to no forgetting, and retains strong zero-shot generalization, frequently outperforming more sophisticated CRL methods. Through detailed analysis, we show that this robustness arises from a synergy between the large pretrained model, parameter-efficient adaptation, and on-policy RL. Together, these components reshape the stability–plasticity trade-off, making continual adaptation both stable and scalable. Our results position Sequential Fine-Tuning as a powerful method for continual RL with VLAs and provide new insights into lifelong learning in the large model era.**

## I. Introduction

Vision-Language-Action (VLA) models represent an emerging paradigm toward building general-purpose embodied agents. By fine-tuning VLMs for decision-making, these systems have demonstrated strong generalization across diverse scenarios (O'Neill et al., 2024; Kim et al., 2024a; Black et al., 2024). However, despite their broad competence, current VLA models remain brittle when deployed in evolving or out-of-distribution settings, where reliability and sustained adaptation become critical. This gap highlights the need for continual learning mechanisms that enable VLAs to incrementally refine and extend their capabilities through ongoing interaction, thereby transforming strong initial generalization into self-sustained, lifelong competence.

Such incremental self-improvement, where an agent needs to learn from a non-stationary stream of tasks and experiences, can be formalized as Continual Reinforcement Learning (CRL). The simplest approach to tackle CRL is through *Sequential Fine-Tuning* (Seq. FT), where the model is directly finetuned on each new task or environments as it arrives. However, much prior work has shown that Seq. FT is prone to **catastrophic forgetting**, where the model's performance on previously learned tasks degrades substantially as it adapts to new ones (French, 1999; Kirkpatrick et al., 2017; Goodfellow et al., 2013). To mitigate this effect, existing CRL methods introduce mechanisms such as regularization (Kirkpatrick et al., 2017), replay (Rolnick et al., 2019; Buzzega et al., 2020), or parameter isolation (Mallya & Lazebnik, 2018; Yu et al., 2025b) to constrain parameter updates. While these approaches are effective at preserving performance on previously learned tasks, they often come at the cost of **plasticity loss**, where the model's ability to adapt to new tasks gradually diminishes. This trade-off between retaining past knowledge and remaining adaptable is known as the stability–plasticity dilemma, which poses a fundamental challenge for continual learning.

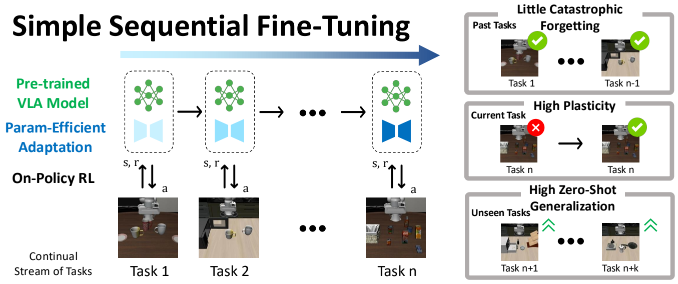



Fig. 1: **VLAs as Natural Continual Learners**. We show that the synergy between pre-trained VLA, on-policy RL, and LoRA is enough to overcome catastrophic forgetting while maintaining plasticity, enabling simple Sequential Fine-Tuning to achieve surprisingly good performance.

The application to VLA models appears to make things even more difficult: on the one hand, modern VLA models contain billions of parameters and result in extremely computationally costly training. Therefore, efficient VLA post-training requires *parameter-efficient fine-tuning* (PEFT) methods, such as LoRA (Hu et al., 2021), which in turn raises new questions about how PEFT interacts and potentially synergizes with CRL strategies. On the other hand, these VLA models come with valuable pre-trained knowledge and strong zero-shot performance. As a result, we desire CRL algorithms that not only maintain the performance of trained tasks, but also preserve (and possibly enhance) these valuable **zero-shot generalization capabilities**.

How do existing CRL methods handle these aforementioned challenges? Does the interplay between large pretrained VLAs, PEFT adaptation, and RL introduce new technical difficulties? In this paper, we seek to answer these questions, by conducting a thorough empirical study of existing CRL methods across 3 different VLA models and 5 challenging lifelong RL benchmarks. Our findings are striking. Across a wide range of CRL methods, the simple strategy of **Sequential Fine-Tuning** under standard low-rank adaptation (LoRA) consistently achieves **high plasticity and performances**, while exhibiting **little to no forgetting** and strong **zero-shot generalization** performance that often surpasses the multi-task oracle. In contrast, existing CRL methods, despite often making additional assumptions such as access to previous data and/or weights, consistently suffer from reduced plasticity due to their added constraints, leading to inferior adaptation to new tasks.

These findings are exciting because they reveal an unexpectedly simple yet highly effective path toward scalable lifelong adaptation in large VLAs. However, they are also quite puzzling, since they stand in stark contrast to previous results from the continual learning community, where Sequential Fine-Tuning typically leads to severe forgetting and thus low performance. Upon further investigation, we find that the robustness of naive finetuning emerges from the interplay between large pre-trained VLAs, LoRA-based parameter-efficient adaptation, and on-policy reinforcement learning. Rather than exacerbating instability, these components collectively make continual adaptation more stable, while synergistically preserving the learning plasticity. More specifically, our analysis finds that each of these three components mitigates catastrophic forgetting from a complementary perspective, and removing any single one of them causes a significant increase in forgetting. Taken together, our results and analysis establish parameter-efficient Sequential Fine-Tuning as a simple but effective method for continual reinforcement learning with VLA models. These results, supported by our open-source implementation, offer a principled starting point for future work on scalable lifelong embodied intelligence.

## II. Problem Formulation

### A. Language-Conditioned MDP for VLA Post-Training

We formulate each task in VLA post-training as a finite-horizon, language-conditioned Markov Decision Process (MDP):

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, H, \mu_0, \ell, r),$$

where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function, $H$ is the horizon, $\mu_0$ is the initial state distribution, $\ell \in \mathcal{L}$ is a natural-language instruction specifying the task, and $r : \mathcal{S} \times \mathcal{A} \times \mathcal{L} \to \{0, 1\}$ is a **sparse reward function**. For each task, the VLA policy $\pi_\theta(a_t \mid s_t, \ell)$ is trained to maximize the cumulative reward.

In our work, all tasks share the same state and action space, where the state space consists of camera images, and the action space consists of robot end-effector pose and gripper command.

### B. Continual Reinforcement Learning in Language-Conditioned MDPs

In the continual setting, the agent learns sequentially over $T$ tasks $\{\mathcal{T}_1, \dots, \mathcal{T}_T\}$ in fixed order that is beyond the control of the agent, where each task $\mathcal{T}^k$ is represented by a language instruction $\ell^k$ and its corresponding sparse reward function $r^k$. Up to task $k$, the CRL objective is to optimize the average return over all seen tasks:

$$\max_\theta J_{\mathrm{CRL}}(\theta) = \frac{1}{k} \sum_{j=1}^{k} \mathbb{E}_{\pi_\theta} \left[ \sum_{t=1}^{H} r^j \right].$$

The agent learns each task purely through interacting with the environment, **without access to any demonstrations**. A defining characteristic of the CRL setting is that, when learning task $\mathcal{T}^k$, the agent cannot access data or interact with the environments of previous tasks $\{\mathcal{T}^1, \dots, \mathcal{T}^{k-1}\}$.

### C. Evaluation Metrics

Following the existing literature (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Zheng et al., 2023; Abel et al., 2023), we adopt standard continual learning metrics for performance evaluation, including *Average Success (AVG)*, which measures overall performance at the end of training, *Negative Backward Transfer (NBT)*, which measures forgetting, and *Forward Transfer (FWT)*, which measures generalization. In addition, we introduce *Zero-Shot Success (ZS)* as a new metric to measure the ability of the algorithm to retain pre-trained capabilities in the VLA. We describe these metrics in detail in the supplementary material (Appendix D).

## III. An Empirical Study of Continual RL for VLAs

In this section, we empirically evaluate continual reinforcement learning (CRL) methods for post-training large Vision-Language-Action (VLA) models. In Sec. III-A, we describe the experiment setup and algorithms. In Sec. III-B and Sec. III-C, we present our results and findings.

### A. Experimental Setup

We follow a consistent training protocol across all methods to ensure fair comparison. As explained in Appendix A, all of our experiments are conducted with GRPO and LoRA unless noted otherwise. Specifically, all methods share the same core hyperparameters, including network architecture, learning rate, batch size, optimizer config, LoRA rank, and GRPO hyperparameters, which we directly inherit from the default configuration of Yu et al. (2025a). For method-specific hyperparameters (e.g., EWC coefficient, Replay coefficient), we perform a local sweep within one order of magnitude of the values reported in the original papers and select the best-performing setting. Notably, we do not do any hyperparameter tuning for *Sequential Fine-Tuning*. We provide additional details in the supplementary material, including details regarding the base VLA, pretraining datasets, train/heldout splits, and training durations (Appendix G), as well as shared and method-specific hyperparameters (Appendix I-J). We aggregate results across 3 independent random seeds for each experiment and report mean $\pm$ standard error.

*a) CRL Algorithms:* We focus our evaluation on eight algorithms spanning the dominant paradigms in Continual Reinforcement Learning. As reference points, Sequential Fine-Tuning (often used in prior work as lower bound) trains tasks sequentially without any forgetting-prevention mechanism, while Multi-Task Training (upper bound oracle) breaks the non-stationary assumption and trains jointly on all tasks simultaneously. Next, we evaluate representatives of the three principal CRL paradigms (Pan et al., 2025): Elastic Weight Consolidation (Kirkpatrick et al., 2017) (regularization-based), Expert Replay (Rolnick et al., 2019) and Dark Experience Replay (Buzzega et al., 2020) (replay-based), and Dynamic Weight Expansion (parameter isolation). We additionally evaluate two methods motivated by large pretrained model adaptation: SLCA (Zhang et al., 2023), which applies layerwise learning-rate decoupling to preserve pretrained representations, and RETAIN (Yadav et al., 2025), which uses discounted weight merging to balance adaptation and retention. Full descriptions of each algorithm are provided in Appendix E.

## B. Results: A Study of CRL Methods on VLAs

*a) Evaluation Domains:* For the first set of experiments, we evaluate on three benchmarks: libero-object, libero-spatial, and libero-long-horizon. All three benchmarks consist of challenging robot manipulation tasks, with each focusing on different aspects of knowledge transfer[1]. Although the LIBERO benchmarks provide expert demonstrations, we do not use these demonstrations during continual post-training, except in the ER method, where they are used for replay. In each of these tasks, the VLA model takes in an RGB image and a natural-language instruction, and outputs a sequence of 7-dimensional actions that controls the end-effector poses and the gripper state. We visualize these benchmarks in Appendix K, and refer the reader to Liu et al. (2023a) for a more detailed description of these tasks. We present the results in Table I.

Across the three benchmarks, Sequential Fine-Tuning (Seq. FT) consistently achieves strong performance (Fig. 2). In terms of **Average Success on the training tasks (AVG)**, Seq. FT achieves performance similar to replay-based and parameter isolation methods, and surpasses the rest of the CRL methods. While the average training success of Seq. FT is often slightly lower than the multi-task oracle, this gap is generally quite small and can be closed under modest modifications to the training setup, as we will demonstrate in Sec. F.

In the meantime, Sequential Fine-Tuning consistently preserves **strong zero-shot generalization capabilities**, and often outperforms the multi-task oracle. This observation indicates that Seq. FT does not degrade, and often enhances, the pretrained model's generalization capabilities.

Such surprisingly strong performance stems from the fact that naive Sequential Fine-Tuning exhibits almost **no forgetting** in these experiments (Fig. 3). Contrary to the conventional expectation that Sequential Fine-Tuning suffers from severe catastrophic forgetting, we observe little performance degradation on previously learned tasks, with the NBT metric consistently showing less than 2% of (and sometimes even negative) forgetting. Given the absence of significant forgetting, it is therefore reasonable that Sequential Fine-Tuning performs competitively. Since it imposes no constraints or regularization on parameter updates, the optimization process can focus entirely on fitting the current task without incurring stability–plasticity trade-offs.

[1]Note that while some recent papers claimed high success rate on the "libero benchmarks", they are often ignoring the continual learning assumptions, training on the test tasks, training without considering the epoch limits, and/or training with expert demonstrations, which makes those results inapplicable to our problem setup.

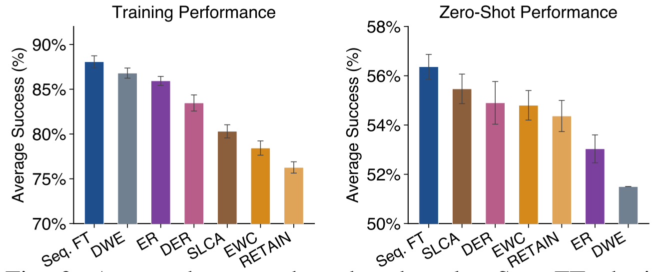


Fig. 2: Averaged across three benchmarks, Seq, FT obtains strong performance in both performance (AVG) and generalization (ZS).

By contrast, **the addition of CRL techniques does not provide much added benefit** and often hurt the performance. EWC, SLCA, and RETAIN all suffer a significant loss in plasticity, as illustrated by their lower average success rate due to constrained parameter updates. DWE cannot benefit from positive transfer due to parameter isolation. Replay-based methods require access to expert demonstrations and storage that grows with the number of tasks, yet do not improve performance.

Together, these results suggest that Sequential Fine-Tuning could be a strong minimal-assumption approach for continual post-training of large VLA models. The observation that Sequential Fine-Tuning simultaneously exhibits minimal forgetting, good plasticity, and preserved generalization challenges conventional expectations in continual learning. A natural question is whether this behavior is specific to the three evaluated benchmarks, or whether it reflects a more general property of large pretrained models trained with on-policy RL.

To examine the robustness of this phenomenon, we next introduce a series of controlled variations to the training setup, including environmental perturbations, changes of physical engine and VLA models, and task-order modifications. As we will show, the favorable properties of Sequential Fine-Tuning persist under these variations. Finally, in Sec. IV, we provide mechanistic analysis and additional empirical evidence to better understand the source of this unexpected stability.

## C. Robustness Under Controlled Perturbations

To assess whether the strong performance of Sequential Fine-Tuning depends on specific benchmark configurations, we conduct additional experiments under controlled perturbations. We examine three axes of variation: (1) environmental perturbations that alter visual and state conditions, (2) changes in domain and model architecture, and (3) modifications to

TABLE I: Comparison of performance across CRL algorithms. Each number represent success rate of tasks (%). In addition to the metrics discussed in Sec. II-C, we report Δ between the initial checkpoint and the final checkpoint to indicate performance change during training. We **bold** the highest-performing method for each metric, not including the multitask oracle.

| **Domain** / Method | **Metrics (%)** | | | | | |
|---|---|---|---|---|---|---|
| | **AVG ↑** | **ΔAVG ↑** | **NBT ↓** | **FWT ↑** | **ZS ↑** | **ΔZS ↑** |
| **libero-spatial** | | | | | | |
| Sequential Fine-Tuning | **81.2±0.4** | **+24.3** | 0.3±0.5 | **3.9±1.5** | **57.1±1.1** | **+5.6** |
| Elastic Weight Consolidation | 66.1±0.9 | +9.3 | 0.7±1.7 | 1.5±0.3 | 52.6±0.9 | +1.1 |
| Expert Replay | 80.2±0.5 | +23.3 | 0.6±1.1 | -2.3±0.1 | 49.2±1.0 | -2.3 |
| Dark Experience Replay | 73.4±1.3 | +16.6 | 4.7±1.3 | 0.7±0.9 | 55.2±0.7 | +3.7 |
| Dynamic Weight Expansion | 79.6±0.9 | +22.7 | 0.0±0.0 | 0.0±0.0 | 51.5±0.0 | +0.0 |
| SLCA (Layered LR) | 69.9±0.7 | +13.0 | **-0.6±2.0** | 1.5±0.3 | 56.1±0.9 | +4.6 |
| RETAIN (Weight Merging) | 66.0±0.7 | +9.1 | 2.9±1.4 | 1.4±1.4 | 53.7±0.8 | +2.2 |
| Multitask (Oracle) | 85.8±0.2 | +28.9 | – | – | 51.2±0.7 | -0.3 |
| **libero-object** | | | | | | |
| Sequential Fine-Tuning | **93.2±0.7** | **+37.6** | 1.0±0.7 | 7.1±0.8 | 25.4±0.2 | +5.8 |
| Elastic Weight Consolidation | 82.6±1.2 | +26.9 | 0.1±0.8 | **10.0±0.4** | 25.3±0.8 | +5.6 |
| Expert Replay | 88.8±0.2 | +33.1 | 4.5±0.6 | 6.4±1.1 | **26.7±0.5** | **+7.1** |
| Dark Experience Replay | 89.1±0.2 | +33.4 | 0.8±1.1 | 6.8±0.8 | 24.8±1.7 | +5.2 |
| Dynamic Weight Expansion | 92.4±0.3 | +36.7 | 0.0±0.0 | 0.0±0.0 | 19.6±0.0 | +0.0 |
| SLCA (Layered LR) | 84.1±0.7 | +28.4 | **-1.6±0.5** | +5.2±1.4 | 24.2±0.2 | +4.6 |
| RETAIN (Weight Merging) | 76.6±0.3 | +20.9 | 0.8±1.0 | 1.8±1.5 | 22.5±0.9 | +2.9 |
| Multitask (Oracle) | 95.7±0.7 | +40.1 | – | – | 27.6±1.3 | +8.0 |
| **libero-long-horizon** | | | | | | |
| Sequential Fine-Tuning | **89.8±0.9** | **+6.8** | **-2.4±1.0** | 0.5±0.1 | 86.6±0.2 | +3.3 |
| Elastic Weight Consolidation | 86.6±0.3 | +3.6 | 0.8±1.3 | **3.0±1.3** | 86.5±0.1 | +3.1 |
| Expert Replay | 88.8±0.8 | +5.8 | -0.2±1.7 | -1.1±0.6 | 83.2±0.2 | -0.1 |
| Dark Experience Replay | 87.6±0.4 | +4.6 | 0.7±0.8 | 0.7±0.2 | 84.7±0.2 | +1.3 |
| Dynamic Weight Expansion | 88.4±0.5 | +5.4 | 0.0±0.0 | 0.0±0.0 | 83.4±0.0 | +0.0 |
| SLCA (Layered LR) | 86.9±0.6 | +3.9 | -1.3±1.0 | -0.2±0.3 | 86.1±0.7 | +2.7 |
| RETAIN (Weight Merging) | 86.2±0.9 | +3.2 | 1.6±1.0 | 1.0±1.2 | **86.9±0.2** | **+3.6** |
| Multitask (Oracle) | 90.5±0.8 | +7.5 | – | – | 85.2±0.5 | +1.8 |

the task order in the continual sequence. Across all settings, we evaluate whether the three key properties observed earlier, namely minimal forgetting, good plasticity, and preserved zero-shot generalization, continue to hold. We elaborate on the setups and results of these experiments in Appendix C.

Taken together, these robustness experiments indicate that the unexpected stability of Sequential Fine-Tuning is not a fragile artifact of benchmark design, but a consistent pattern across environmental, architectural, and sequential variations. We therefore turn to a mechanistic analysis to better understand the source of this behavior.

## IV. Analysis: What Makes Sequential Fine-Tuning So Effective?

Given the experimental results in Sec. III, we conduct further analysis and additional experiments in this section towards better understanding the surprising effectiveness of Sequential Fine-Tuning. We focus our analysis from the following three properties of Sequential Fine-Tuning in our experiments: little catastrophic forgetting, strong plasticity, and good zero-shot generalization. Finally, in Appendix F, we discuss how we can potentially close the already small gap on Average Success between the Sequential Fine-Tuning method and the Multitask Oracle.

### A. Why Little Catastrophic Forgetting?

Most previous CRL methods are designed to mitigate catastrophic forgetting, with results showing that Sequential Fine-Tuning leads to significant unlearning of previous tasks. This mismatch raises a key question: why can simple Sequential Fine-Tuning avoid catastrophic forgetting in our experiments in the VLA domain? To investigate this phenomenon, we start by conducting ablation studies by (1) removing the RL objective (reducing to SFT), (2) replacing the large VLA model with a smaller neural network with 12 Million parameters, pre-trained to a similar (but inevitably slightly different) initial performance, and (3) removing LoRA. We describe the detailed setup of these experiments in Appendix H, and show the results in Tab. III.

The results here are revealing: all three components play a crucial role. Removing any one of them leads to a significant drop in both AVG performance and zero-shot generalization, where the model quickly loses all pre-trained capabilities during RL finetuning. In the following paragraphs, we analyze how each factor contributes to mitigating forgetting.

**Effect of On-Policy RL:** The observation that on-policy RL helps prevent forgetting has been noted in several recent papers in the LLM domain (Shenfeld et al., 2026; Chen et al., 2025; Lai et al., 2026). While no previous work has demonstrated this phenomenon in the VLA domain, it is perhaps not surprising that a similar conclusion holds.

As pointed out in Shenfeld et al. (2026), this effect can largely be attributed to the use of on-policy data. Specifically, let $\pi_0(a \mid s)$ denote the base policy and $\pi_\theta(a \mid s)$ the adapted policy. Supervised fine-tuning (SFT) learns with

$$\nabla_\theta \mathcal{L}_{\mathrm{SFT}} = -\mathbb{E}_{(s,a)\sim D_{\text{task}}}\big[\nabla_\theta \log \pi_\theta(a \mid s)\big].$$

Thus, SFT increases the log-probability of dataset actions regardless of how small $\pi_0(a \mid s)$ was. If the dataset contains actions outside the high-probability region of $\pi_0$, probability mass must be shifted into regions where $\pi_0(a \mid s)$ is small. This necessarily increases the forward KL divergence

$$\mathrm{KL}(\pi_\theta \,\|\, \pi_0) = \mathbb{E}_s \; \mathbb{E}_{a\sim\pi_\theta(\cdot|s)} \left[\log \frac{\pi_\theta(a \mid s)}{\pi_0(a \mid s)}\right],$$

which grows when $\pi_\theta$ allocates mass to actions unlikely under $\pi_0$.

The policy gradient update, by contrast, results in

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim d_{\pi_\theta},\, a\sim\pi_\theta}\big[A^{\pi_\theta}(s,a)\, \nabla_\theta \log \pi_\theta(a \mid s)\big].$$

where $d_{\pi_\theta}$ is the on-policy state distribution and $A^{\pi_\theta}(s,a)$ is the advantage function. Crucially, both the objective and its gradient are weighted by samples $(s,a) \sim d_{\pi_\theta}(s)\pi_\theta(a \mid s)$. In other words, policy gradient updates only reweight probability mass where $\pi_\theta$ already has support, and cannot suddenly assign high probability to actions with near-zero probability. As a result, the probability mass can only move gradually outward from the support of $\pi_0$, creating an implicit objective that minimizes KL drift from $\pi_0$. Since forgetting empirically correlates with forward KL from $\pi_0$ (Shenfeld et al., 2026), such an implicit regularization helps the model retain its learning capability and mitigate catastrophic forgetting.

While it is impressive that RL helps alleviate catastrophic forgetting, it is equally worth noticing that, unlike in previous work (Shenfeld et al., 2026; Chen et al., 2025; Lai et al., 2026), our results on the VLA domains suggest that *on-policy RL alone is not sufficient for avoiding catastrophic forgetting*, and both the large pretrained model and parameter-efficient adaptation (i.e., LoRA) are also critical for maintaining performance.

**Effect of Large Pretrained Models:** The effect of large pretrained models for mitigating forgetting can be largely attributed to the curse (or rather "blessing" in our case) of dimensionality (Mirzadeh et al., 2022). Specifically, for two random unit vectors $u, v \in \mathrm{Unif}(\mathcal{S}^{d-1})$, it is well known that $\sqrt{d}\langle u, v\rangle \to \mathcal{N}(0,1)$ as $d \to \infty$. In other words, in high-dimensional space, almost all random vectors are nearly orthogonal. As a result, overparametrized models inherently create a vast "Null Space" where gradient updates in most directions barely affect the pre-trained knowledge, as also noted in concurrent work (Liu et al., 2026).

We empirically validate this analysis via examining the Fisher Information (Kirkpatrick et al., 2017). Let $\boldsymbol{\theta} \in \mathbb{R}^D$ denote the model parameters, $\mathbf{g} = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$ the gradient of the loss of the *current training task*, and $\mathbf{F} \in \mathbb{R}^{D\times D}$ denote the Fisher Information Matrix (FIM) with respect to the *pre-training tasks*. Using a local second-order approximation, the increase in the pre-training loss under a parameter update $\Delta$ can be written as

$$L_{\text{old}}(\boldsymbol{\theta} + \Delta) \approx L_{\text{old}}(\boldsymbol{\theta}) + \tfrac{1}{2}\Delta^\top \mathbf{F}\Delta.$$

Thus, if the current task updates parameters along direction $\mathbf{g}$, the resulting increase in the old-task loss is governed by $\mathbf{g}^\top\mathbf{F}\mathbf{g}$. We therefore compute the Rayleigh quotient of the Fisher Information Matrix along the gradient direction as

$$E_F(\mathbf{g}) = \frac{\mathbf{g}^\top\mathbf{F}\mathbf{g}}{\mathbf{g}^\top\mathbf{g}} = \frac{\sum_{d=1}^{D} f_d\, g_d^2}{\sum_{d=1}^{D} g_d^2}.$$

We define $E_F(\mathbf{g})$ as the *Fisher energy*, which measures the average curvature of the pre-training tasks along the gradient direction of the current task, and therefore quantifies how strongly the new task will interfere with the pretrained knowledge, where a high value indicates more interference.

Since the full FIM scales quadratically with the number of parameters, we use a diagonal empirical approximation for the FIM: $\mathbf{F} \approx \mathrm{diag}(f_1, \ldots, f_D), \quad \text{where} \quad f_d = \mathbb{E}\big[g_d^2\big]$, and normalize it by $\max_d(f_d)$ so that the value is in $[0,1]$. We examine $E_F(\mathbf{g})$ for both the small neural network policy from ablation study, and the large OpenVLA-OFT model on the libero-spatial task suite. On the large OpenVLA-OFT model, the average $E_F$ is only 0.02, indicating very little interference between the task gradient and pretrained knowledge. However, on the small policy, $E_F$ jumps to 0.16, which likely explains the catastrophic forgetting that occurs with small models.

**Effect of Low-Rank Adaptation:** LoRA constrains fine-tuning updates to a low-rank subspace, restricting the gradient update $\Delta W$ to a rank-$r$ subspace around the pretrained weight $W_0$. By concentrating task-specific changes within this narrow, low-dimensional subspace ($r \ll d$), LoRA limits the degrees of freedom of the update, preventing simultaneous alterations of the high-energy principal directions of the model. Therefore, it is perhaps not very surprising that LoRA can alleviate catastrophic forgetting and preserve pre-trained knowledge. However, our empirical analysis suggests that the effect of LoRA may be deeper than this simple interpretation. Rather than merely reducing the total capacity of the update, LoRA appears to *prevent a small subset of layers from undergoing disproportionately large structural changes* during fine-tuning.

To examine this hypothesis, we conduct empirical analysis comparing the weight update $\Delta W$ with and without using LoRA. In full fine-tuning, the mean effective rank per layer is 208.6, with a very large standard deviation of 148.5 across different layers. This result indicates that full fine-tuning caused a subset of layers to undergo extremely high-rank updates, indicating uneven adaptation and potential overwriting of pretrained representations in those layers. By contrast, LoRA (with rank 32) produces a nearly uniform pattern across layers: the mean effective rank is 29.3, with a tiny standard deviation of 2.16. The mean nuclear norm per layer is also lower for LoRA (0.259 vs. 0.609), indicating that LoRA not only limits rank but also reduces the total magnitude of directional modification per layer. These statistics support the interpretation that LoRA reduces catastrophic forgetting primarily by constraining the per-layer update geometry: it prevents any single layer from undergoing uncontrolled, high-rank structural modification.

To summarize, **RL, LoRA, and the VLA itself alleviate catastrophic forgetting from three complementary perspectives: objective, constraints, and capacity**. As a result, their synergistic combination leads to stable learning without forgetting in a way that no two of them alone exhibit, as we empirically observe in our experiments.

### *B. Why Good Plasticity?*

The ability of Sequential Fine-Tuning to learn new tasks effectively is well-known (Liu et al., 2023a), but it is more surprising that this good plasticity is preserved even when LoRA is applied. In particular, previous studies have noted that "LoRA often underperforms in supervised pre-training" (Biderman et al., 2024), where the constrained gradient update reduces the plasticity of the model. This contrast raises the question of why our model, with LoRA applied, is still able to learn effectively and maintain high plasticity in continual Reinforcement Learning.

Upon further investigation, we found that such a result is tightly coupled with the nature of policy gradient RL, and more specifically to its low-capacity requirements. We follow Schulman & Lab (2025), and illustrate this phenomenon from an information-theoretic perspective. Specifically, policy gradient methods such as GRPO learn based on the advantage function, which only provides O(1) bits of information for each episode under a sparse reward setup. For example, in our experiments on OpenVLA-OFT with 7B parameters, the rank-32 LoRA weights contain around 100M parameters, which is more than enough to absorb the information obtained from the 50k training rollout episodes. By contrast, in supervised learning, the information contained in each episode scales linearly with the length of the episode, and therefore often leads to per-episode information that is thousands of time richer than in RL. Such a discrepancy likely leads to the performance loss of LoRA when applied to supervised learning in previous work. This perspective highlights the synergy between on-policy RL and LoRA, as their combination effectively reduces catastrophic forgetting without sacrificing much plasticity.

### *C. Why Good Zero-shot Generalization?*

Finally, we observe that Sequential Fine-Tuning consistently preserves strong zero-shot generalization. Since maintaining zero-shot capability can be viewed as a form of preventing forgetting, this behavior can largely be understood through the same mechanisms discussed in Sec. IV-A.

What is more intriguing is that Sequential Fine-Tuning often maintains a slight edge over oracle multi-task training on the generalization capabilities. Although this gap is generally small on the benchmarks we evaluate, the trend is consistent across settings and therefore noteworthy. We do not yet have a definitive explanation for this phenomenon. One plausible hypothesis is that task sequencing acts as a form of implicit regularization. Rather than jointly optimizing over all tasks and potentially overfitting to the aggregated objective, sequential training exposes the model to a shifting objective over time (Abel et al., 2023). Such non-stationary optimization dynamics may encourage more robust representations and improved generalization. Investigating this implicit regularization effect more rigorously remains an exciting direction for future work.

## V. Conclusion

In this work, we conducted a systematic study of Continual Reinforcement Learning for large Vision-Language-Action (VLA) models. Our investigation yielded a surprising and significant result: the simple approach of Sequential Fine-Tuning with Low-Rank Adaptation achieves strong plasticity, minimal forgetting, enhanced zero-shot generalization, and frequently outperforms more sophisticated CRL methods. Further analysis reveals that this stability is not accidental but emerges from a synergy between the large pretrained model, parameter-efficient fine-tuning (LoRA), and the stable nature of on-policy RL post-training. These components collectively reshape the stability-plasticity dilemma, allowing the model to adapt to new tasks without overriding previous knowledge. Together, these findings offer us a simple but scalable recipe of how RL can be used as a powerful continual post-training paradigm for large pre-trained VLA models.

One natural future direction is to apply these findings to empower physical robotic systems, either via sim-to-real transfer (Tobin et al., 2017; Zhao et al., 2020) or real-world reinforcement learning (Hu et al., 2025b; Zhu et al., 2020). More generally, our results suggest that, as pre-trained models become larger and more capable, the traditional focus on catastrophic forgetting may no longer be the primary bottleneck in continual RL. Instead, future work may benefit from designing algorithms that emphasize efficient adaptation and improved zero-shot generalization. Ultimately, our findings and open-source codebase provide a principled starting point for the community to build more capable and adaptable lifelong embodied agents.

## REFERENCES


Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on lifelong learning agents*, pp. 620–636. PMLR, 2023.

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado P van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. *Advances in Neural Information Processing Systems*, 36:50377–50407, 2023.

Nishanth Anand and Doina Precup. Prediction and control in continual reinforcement learning, 2023. URL https://arxiv.org/abs/2312.11669.

Glen Berseth, Zhiwei Zhang, Grace Zhang, Chelsea Finn, and Sergey Levine. Comps: Continual meta policy search. *arXiv preprint arXiv:2112.04467*, 2021.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan C. Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Anand Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Ho Vuong, F. Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *ArXiv*, abs/2212.06817, 2022. URL https://api.semanticscholar.org/CorpusID:254591260.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Krzysztof Choromanski, Tianli Ding, Danny Driess, Kumar Avinava Dubey, Chelsea Finn, Peter R. Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil J. Joshi, Ryan C. Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Sergey Levine, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Ho Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Ted Xiao, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *ArXiv*, abs/2307.15818, 2023. URL https://api.semanticscholar.org/CorpusID:260293142.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*, 2025.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Xiang Li, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, Tiejun Huang, Yu Wang, and Chao Yu. $\pi_{\texttt{RL}}$: Online rl fine-tuning for flow-based vision-language-action models, 2026. URL https://arxiv.org/abs/2510.25889.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44:1684 – 1704, 2023. URL https://api.semanticscholar.org/CorpusID:257378658.

Haoyuan Deng, Zhenyu Wu, Haichao Liu, Wenkai Guo, Yuquan Xue, Ziyu Shan, Chuanrui Zhang, Bofang Jia, Yuan Ling, Guanxing Lu, et al. A survey on reinforcement learning of vision-language-action models for robotic manipulation. *Authorea Preprints*, 2025.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024.

Esraa Elelimy, David Szepesvari, Martha White, and Michael Bowling. Rethinking the foundations for continual reinforcement learning. *arXiv preprint arXiv:2504.08161*, 2025.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 12799–12807, 2023.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv* *preprint arXiv:1312.6211*, 2013.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Jiaheng Hu, Rose Hendrix, Ali Farhadi, Aniruddha Kembhavi, Roberto Martín-Martín, Peter Stone, Kuo-Hao Zeng, and Kiana Ehsani. Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3617–3624. IEEE, 2025a.

Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. Slac: Simulation-pretrained latent action space for whole-body real-world rl. In *Proceedings of The 9th Conference on Robot Learning*, pp. 2966–2982, 2025b.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

Christos Kaplanis, Murray Shanahan, and Claudia Clopath. Policy consolidation for continual reinforcement learning. In *International Conference on Machine Learning*, pp. 3242–3251. PMLR, 2019.

Samuel Kessler, Mateusz Ostaszewski, MichałPaweł Bortkiewicz, Mateusz Żarski, Maciej Wolczyk, Jack Parker-Holder, Stephen J Roberts, Piotr Mi, et al. The effectiveness of world models for continual reinforcement learning. In *Conference on Lifelong Learning Agents*, pp. 184–204. PMLR, 2023.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Grace Lam, Pannag R. 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*, abs/2406.09246, 2024b. URL https://api.semanticscholar.org/CorpusID:270440391.

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

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.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Qingfu Zhang, Hongbin Liu, Gaofeng Meng, and Fei Zhu. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training, 2026. URL https://arxiv.org/abs/2507.05386.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

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

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

Zuxin Liu, Jesse Zhang, Kavosh Asadi, Yao Liu, Ding Zhao, Shoham Sabach, and Rasool Fakoor. Tail: Task-specific adapters for imitation learning with large pretrained models. *arXiv preprint arXiv:2310.05905*, 2023b.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Yuan Meng, Zhenshan Bing, Xiangtong Yao, Kejia Chen, Kai Huang, Yang Gao, Fuchun Sun, and Alois Knoll. Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nature Machine Intelligence*, 7(2):256–269, 2025.

Golnaz Mesbahi, Parham Mohammad Panahi, Olya Mastikhina, Steven Tang, Martha White, and Adam White. Position: Lifetime tuning is incompatible with continual reinforcement learning, 2025. URL https://arxiv.org/abs/2404.02113.

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Huiyi Hu, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Wide neural networks forget less catastrophically. In *International conference on machine learning*, pp. 15699–15717. PMLR, 2022.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Chaofan Pan, Xin Yang, Yanhua Li, Wei Wei, Tianrui Li, Bo An, and Jiye Liang. A survey of continual reinforcement learning. *arXiv preprint arXiv:2506.21872*, 2025.

Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: Parameter efficient continual learning. volume 37, pp. 97476–97498, 2024.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

John Schulman and Thinking Machines Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. DOI: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *ICLR*, 2026.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):1–42, 2025.

Hongyao Tang, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, and Glen Berseth. Mitigating plasticity loss in continual reinforcement learning by reducing churn, 2025. URL https://arxiv.org/abs/2506.00592.

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 (IROS)*, pp. 23–30. IEEE, 2017.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*, 2025.

Maciej Wolczyk and et al. Disentangling transfer in continual reinforcement learning. *Advances in Neural Information Processing Systems*, 35:6304–6317, 2022.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11097–11107, 2020.

Annie Xie and Chelsea Finn. Lifelong robotic reinforcement learning by retaining experiences. In *Conference on Lifelong Learning Agents*, pp. 838–855. PMLR, 2022.

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.

Chao Yu, Yuanqing Wang, Zhen Guo, Hao Lin, Si Xu, Hongzhi Zang, Quanlu Zhang, Yongji Wu, Chunyang Zhu, Junhao Hu, et al. Rlinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation. *arXiv preprint arXiv:2509.15965*, 2025a.

Jiazuo Yu, Zichen Huang, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Moe-adapters++: Towards more efficient continual learning of vision-language models via dynamic mixture-of-experts adapters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19148–19158, 2023.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744. IEEE, 2020.

Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 19125–19136, 2023.

Henry Zhu, Justin Yu, Abhishek Gupta, Dhruv Shah, Kristian Hartikainen, Avi Singh, Vikash Kumar, and Sergey Levine. The ingredients of real-world robotic reinforcement learning. *arXiv preprint arXiv:2004.12570*, 2020.

## Appendix A
## Background & Related Work

*a) Vision-Language-Action Models.:* VLA models unify visual perception, natural-language conditioning, and action generation in a single policy. They are typically trained on large-scale robot datasets by imitation learning which results in generalization capability across tasks and environments. A major family of models adopts *autoregressive* action generation: RT-1, RT-2, and OpenVLA (Brohan et al., 2022, 2023; Kim et al., 2024b) discretize actions into tokens and decode them auto-regressively conditioned on images and task instructions. A closely related variant uses *action chunking*, where the policy predicts short action horizons at each decision step rather than a single action, with OpenVLA-OFT as a representative example (Kim et al., 2025). Another family of approaches uses *continuous generative* action heads: diffusion-based policies generate actions through iterative denoising (Chi et al., 2023), while Pi-0 adopts a flow-matching head built on a vision-language backbone as an alternative continuous-action VLA design (Black et al., 2024).

*b) Reinforcement Learning Post-Training of VLA Models.:* RL post-training recently emerged as an effective methodology to refine and improve large pretrained Vision-Language-Action (VLA) models (Deng et al., 2025; Lu et al., 2025; Hu et al., 2025a; Yu et al., 2025a; Intelligence et al., 2025; Chen et al., 2026; Wagenmaker et al., 2025). The pretrained generalization capabilities of VLAs allow for effective exploration and open up exciting possibilities for learning from sparse rewards on challenging tasks. A key challenge in RL post-training of VLA Models is maintaining training stability and avoiding performance collapse. Prior work has shown that stable adaptation requires carefully controlled on-policy updates, small learning rates, and well-behaved policy objectives (Hu et al., 2025a; Yu et al., 2025a).

Following this established recipe, we adopt on-policy reinforcement learning throughout this work. In particular, we use Group Relative Policy Optimization (GRPO) (Guo et al., 2025), a stable policy-gradient method that has achieved strong empirical performance in large-scale post-training. We provide a detailed description of GRPO and its application for training autoregressive and flow-based VLAs in Appendix. B.

*c) Continual Reinforcement Learning.:* Continual reinforcement learning (CRL) (Pan et al., 2025; Abbas et al., 2023; Dohare et al., 2024; Tang et al., 2025; Khetarpal et al., 2022; Meng et al., 2025; Mesbahi et al., 2025; Abel et al., 2023; Elelimy et al., 2025) studies RL agents that must adapt continually to non-stationary tasks or environments while retaining competence on previously encountered ones. A common categorization is *what* is transferred across changes (Wolczyk & et al, 2022; Pan et al., 2025), such as value functions (Anand & Precup, 2023), policies (Kaplanis et al., 2019; Berseth et al., 2021), experiences (Xie & Finn, 2022), or learned dynamics models (Kessler et al., 2023), and *how* transfer is implemented (Pan et al., 2025; Khetarpal et al., 2022), which can be grouped into: (i) *regularization-based* methods that constrain parameter updates to reduce interference (Kirkpatrick et al., 2017), (ii) *replay-based* methods that preserve and reuse past experience (Rolnick et al., 2019; Buzzega et al., 2020), and (iii) *parameter-isolation* methods that allocate additional state or parameters to isolate or store knowledge (Rusu et al., 2016). Most of these works only consider small models trained from scratch. By contrast, we focus on CRL applied to large pre-trained VLA models and the intriguing properties that arise from such a setup.

*d) Parameter-Efficient Fine-Tuning.:* Given the scale of modern generative models such as VLAs, full-parameter fine-tuning is often prohibitively expensive, especially in continual learning settings (Shi et al., 2025). This has motivated parameter-efficient fine-tuning (PEFT) (Fu et al., 2023; Ding et al., 2023; Li & Liang, 2021; Hu et al., 2021), which adapts a pretrained network by updating only a small subset of parameters while keeping the backbone weights frozen. Among various PEFT methods, the predominant approach is Low-Rank Adaptation (LoRA) (Hu et al., 2021; Liu et al., 2023b; Qiao & Mahdavi, 2024). LoRA adapts a pretrained model by parameterizing weight updates as low-rank matrices while keeping the original pretrained weights frozen. Concretely, for a pretrained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA parametrizes the adapted weight as

$$W = W_0 + BA,$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ are trainable matrices with rank $r \ll \min(d, k)$. After training, the LoRA weight can be easily merged into the original weight via $W_{\text{new}} \leftarrow W_0 + BA$.

This formulation significantly reduces the number of trainable parameters while preserving the expressive capacity of the pretrained model. Given its strong empirical performance and widespread adoption in large-scale model adaptation, we adopt LoRA as our parameter-efficient fine-tuning method throughout this work.

## Appendix B
## GRPO Training Formulation

In GRPO, at each update, trajectories are sampled from the previous policy $\pi_{\theta_{\text{old}}}$, and the policy is optimized using

$$\max_{\theta} \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}} \Big[\min\Big(\rho_t(\theta)\,\hat{A},\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}\Big)\Big],$$

where

$$\rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t, \ell)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t, \ell)}, \quad \hat{A} = \frac{R - \mu_R}{\sigma_R}.$$

Here $R$ denotes the episodic return of the sampled trajectory, and $\mu_R, \sigma_R$ are the mean and standard deviation of returns within the sampled group.

For VLA models that generate actions via autoregressive tokens (Kim et al., 2024a, 2025), GRPO can be applied directly by treating the sequence of action tokens as the policy output and computing the likelihood ratios over tokens. For VLA models with continuous flow or diffusion action heads (Black et al., 2024; Intelligence et al., 2025), actions

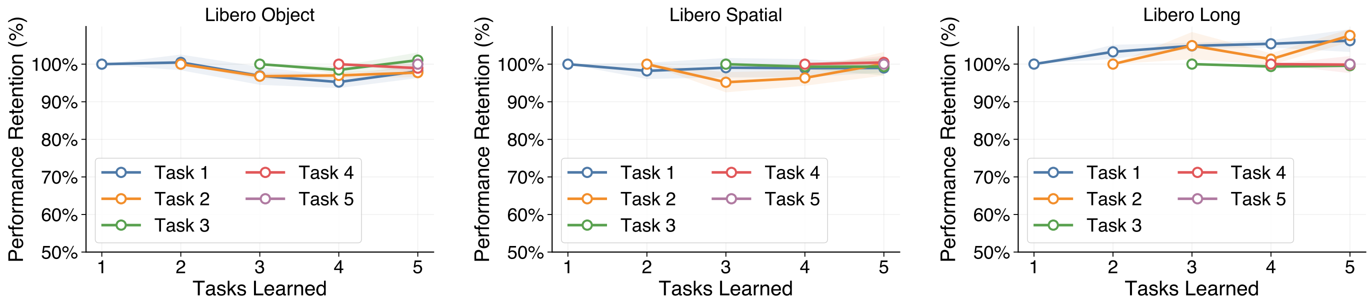



Fig. 3: Each line tracks a single training task's success rate, normalized to 100% at the point it was first learned. Subsequent x-values show how that task's performance changes as additional tasks are learned. Sequential Fine-Tuning shows little forgetting throughout the entire training.

TABLE II: Examining the consistency of Seq. FT performance across different perturbations. We **bold** the metrics for which Seq. FT outperforms the multitask oracle.

| **Domain** / Method | Metrics (%) | | | | | |
|---|---|---|---|---|---|---|
| | AVG ↑ | ΔAVG ↑ | NBT ↓ | FWT ↑ | ZS ↑ | ΔZS ↑ |
| **Camera Perturbation** | | | | | | |
| Seq. FT | **75.5±0.2** | **+18.9** | -0.5±0.5 | 3.7±1.1 | **46.7±0.2** | **-0.6** |
| Multitask (Oracle) | 75.2±0.1 | +18.6 | – | – | 43.8±0.5 | -3.6 |
| **Lighting Perturbation** | | | | | | |
| Seq. FT | 82.4±0.5 | +26.7 | 0.2±0.4 | 5.7±0.1 | **54.9±1.0** | **+1.9** |
| Multitask (Oracle) | 87.0±0.3 | +31.3 | – | – | 54.1±0.3 | +1.2 |
| **Robot State Perturbation** | | | | | | |
| Seq. FT | 81.2±0.9 | +23.4 | 0.6±0.5 | 0.2±0.3 | **42.7±0.7** | **+2.4** |
| Multitask (Oracle) | 86.1±0.3 | +28.3 | – | – | 42.2±0.7 | +1.9 |
| **Pi-0 on RoboCasa** | | | | | | |
| Seq. FT | 29.5±3.0 | +10.6 | -0.1±2.1 | 1.2±1.7 | **21.5±1.9** | **+2.7** |
| Multitask (Oracle) | 31.4±2.3 | +12.5 | – | – | 20.8±1.2 | +2.0 |
| **OpenVLA on ManiSkill** | | | | | | |
| Seq. FT | 70.9±1.5 | +19.4 | -1.0±1.5 | 0.5±0.6 | **51.0±0.8** | **+11.0** |
| Multitask (Oracle) | 72.8±0.2 | +21.2 | - | - | 50.7±0.8 | +10.7 |
| **Task Order Perturbation** | | | | | | |
| Seq. FT (Re-order 1) | 79.8±0.5 | +22.9 | 1.4±1.4 | 3.5±0.3 | **54.4±0.8** | **+3.9** |
| Seq. FT (Re-order 2) | 81.2±1.0 | +24.4 | 1.6±1.7 | 0.8±1.3 | **55.7±0.5** | **+4.2** |
| Seq. FT (Re-order 3) | 80.2±1.0 | +23.3 | -0.3±0.5 | 2.4±1.3 | **57.6±1.0** | **+6.1** |
| Multitask (Oracle) | 85.8±0.2 | +28.9 | – | – | 51.2±0.7 | -0.3 |

are generated by integrating a learned velocity field defined by a deterministic ordinary differential equation (ODE):

$$\frac{dx_t}{dt} = v_\theta(x_t, t).$$

Since deterministic flows do not provide stochastic exploration required by policy gradients, we adopt the Flow-SDE formulation (Chen et al., 2026) and introduce controlled Gaussian noise into the dynamics:

$$dx_t = v_\theta(x_t, t)\, dt + \sigma_t\, dW_t,$$

where $\sigma_t$ is a noise schedule and $dW_t$ is a Wiener process increment. This converts the deterministic sampler into a stochastic policy that defines a distribution over actions. Standard policy gradient objectives (e.g., PPO or GRPO) can then be applied by optimizing the advantage-weighted likelihood over the resulting action trajectories.

## Appendix C
## Robustness Under Controlled Perturbations

**Environmental Perturbations.** First, we assess the robustness of our result to changes in environment parameters across tasks. Specifically, we introduce three types of perturbation: *camera perturbation*, where the camera position and orientation of each task is set to different values; *lighting*

TABLE III: Ablation studies on the libero-spatial benchmark

| Ablation | Metrics | | | | | |
|---|---|---|---|---|---|---|
| | **AVG ↑** | **ΔAVG ↑** | **NBT ↓** | **FWT ↑** | **ZS ↑** | **ΔZS ↑** |
| Seq. FT (Original) | **81.2±0.4** | **+24.3** | **0.3±0.5** | **0.3±0.5** | **57.1±1.1** | **+5.6** |
| SFT instead of RL | 29.9±2.3 | -27.0 | 78.7±1.9 | -53.8±0.0 | 1.1±0.9 | -50.4 |
| Smaller Policy | 13.1±0.9 | -53.7 | 11.4±3.7 | -63.4±0.5 | 0.0±0.0 | -56.2 |
| Without LoRA | 7.3±5.2 | -49.6 | 40.9±11.8 | -50.4±1.3 | 0.0±0.0 | -51.5 |

*perturbation*, where the lighting intensity of each task is different; and *robot state perturbation*, where the location of the robot base is different for each task. These experiments evaluate whether the strong performance of Sequential Fine-Tuning is attributable to the environment parameters remaining constant in the original LIBERO benchmark.

**Domain and Model Variations.** Next, we examine whether our conclusion still holds on different VLAs and in different benchmarks. In particular, besides the OpenVLA-OFT (Kim et al., 2025) model that we used for experiments in Sec. III-B, we additionally evaluate Pi-0 (Black et al., 2024), a flow-matching VLA built on PaliGemma, and OpenVLA (Kim et al., 2024a), an auto-regressive VLA based on Llama 2 that, unlike OpenVLA-OFT, does not use action chunking. We evaluate these models on the RoboCasa (Nasiriany et al., 2024), a benchmark with diverse scenes and many non-pick-and-place tasks, and Maniskill (Gu et al., 2023), a benchmark based on the SAPIEN (Xiang et al., 2020) physical engine, respectively.

**Task Order Sensitivity.** Finally, we investigate the sensitivity of Sequential Fine-Tuning to task ordering. Classical continual learning methods often exhibit strong dependence on the order in which tasks are presented, particularly when tasks differ in difficulty or similarity. We construct alternative task sequences by permuting the order of tasks within the libero-spatial benchmark and repeat the continual training procedure.

We evaluate Sequential Fine-Tuning and the multi-task oracle under these perturbations, and report the results for these experiments in Table II. Across all conditions, Seq. FT maintains strong performance. Specifically, the **AVG** of Seq. FT consistently show a big increase from the base model, and maintains a $< 5\%$ gap with the multi-task oracle (which, as discussed in Sec. F, can be bridged). The **NBT** stays below $2\%$ for all experiments, with frequent negative values, indicating the same absence of catastrophic forgetting that we noticed earlier. Finally, the **ZS** performance maintains a consistent edge over the multitask oracle, demonstrating the surprising ability of Seq. FT to boost generalization.

## APPENDIX D
## EVALUATION METRICS

Suppose tasks arrive sequentially in the order $\{\mathcal{T}_1, \dots, \mathcal{T}_T\}$. After completing training on task $\mathcal{T}_i$, we evaluate the policy on all tasks $\mathcal{T}_j$ and record the success rate $S_{i,j} \in [0,1]$. This produces a success matrix $S \in \mathbb{R}^{T\times T}$, where $S_{i,j}$ denotes the success rate on task $j$ after training up to task $i$. Additionally, we denote the initial performance of the base model on task $j$ as $S_{0,j}$.

*a) Training Average Final Success (AVG).:* The overall performance after learning all tasks is defined as the average final success rate:

$$\text{AVG} = \frac{1}{T}\sum_{j=1}^{T} S_{T,j}. \tag{1}$$

This measures how well the final policy performs across the entire training task sequence.

*b) Negative Backward Transfer (NBT).:* Negative Backward Transfer (a.k.a Forgetting) measures the degradation in performance on previous tasks after learning subsequent ones. Since each task is trained once in sequence, we define forgetting relative to the performance immediately after completing training on that task:

$$\text{NBT} = \frac{1}{T-1}\sum_{j=1}^{T-1} \left(S_{j,j} - S_{T,j}\right). \tag{2}$$

Lower values indicate better retention of previously acquired skills, where a value of 0 indicate that there is no forgetting on average.

*c) Forward Transfer (FWT).:* Forward transfer quantifies whether learning previous tasks improves performance on future tasks before they are trained. Let $S_{0,j}$ denote the zero-shot success rate on task $j$ before any task-specific training. Then

$$\text{FWT} = \frac{1}{T-1}\sum_{j=2}^{T} \left(S_{j-1,j} - S_{0,j}\right). \tag{3}$$

Positive values indicate beneficial transfer to unseen tasks. Importantly, FWT is strongly influenced by the task ordering. To better measure transfer capabilities, we propose an additional metric called the held-out performance, as explained below.

*d) Held-Out Tasks Performance (ZS).:* Unlike in classic continual RL, VLA contain strong zero-shot performance on unseen tasks even before any training occur. To evaluate the ability to retain and potentially enhance these zero-shot capabilities, we assess the final policy on a set of held-out tasks $\mathcal{H}$ not encountered during continual training. Held-out performance is defined as

$$\text{ZS} = \frac{1}{|\mathcal{H}|}\sum_{h\in\mathcal{H}} S^{\text{held}}_{T,h}, \tag{4}$$

where $S^{\text{held}}_{T,h}$ denotes the success rate on held-out task $h$ after completing training on all tasks.

## Appendix E
## Evaluation Algorithms

In this section, we describe the algorithms we evaluated in our study in detail, as well as the reasoning for choosing these algorithms. We begin by establishing two reference points that anchor our evaluation.

**Sequential Fine-Tuning:** The most direct approach to continual learning is to train tasks sequentially without any additional mechanism to prevent forgetting. At each stage, the model is fine-tuned solely on the current task via interaction. Sequential Fine-Tuning requires no replay buffer, parameter isolation, or task-specific regularization. It is commonly treated as a lower-bound baseline in continual RL, as it is expected to suffer from catastrophic forgetting under non-stationary task sequences.

**Multi-Task Training (Oracle):** As an upper-bound reference, we train a model jointly on all tasks, assuming simultaneous access to experiences from the entire task set. This setting violates the sequential and non-stationary assumptions of continual learning and therefore serves as an oracle baseline. Its performance is often used to represent the best achievable performance when task order constraints are removed.

Beyond these reference points, we evaluate a diverse set of continual learning algorithms spanning the principal methodological paradigms in the literature. Continual reinforcement learning (CRL) methods are commonly categorized into three principal paradigms (Pan et al., 2025): (i) *regularization-based methods*, which constrain parameter updates to preserve prior knowledge; (ii) *replay-based methods*, which reuse data or model outputs from previous tasks; and (iii) *parameter-isolation methods*, which allocate task-specific capacity to avoid interference. To systematically evaluate these paradigms, we evaluate the following representative approaches.

- **Elastic Weight Consolidation (Kirkpatrick et al., 2017)** (regularization-based): penalizes parameter updates directly in the weight space, in proportion to their estimated importance to previous tasks using a Fisher-based quadratic constraint.
- **Expert Replay (Rolnick et al., 2019)** (replay-based): stores expert demonstrations for all tasks and replay them during training as an additional Behavior Cloning loss term. Note that this approach requires access to the expert demonstrations, as well as space to store the demonstration data which grows linearly with the number of tasks.
- **Dark Experience Replay (Buzzega et al., 2020)** (replay-based): Instead of replaying labels, DER matches the logits of the previous model, preserving functional behavior while avoiding the use of expert data. Note that this approach requires storing previous interactions and logit values which grow linearly with the number of tasks.
- **Dynamic Weight Expansion** (parameter isolation): We allocate an isolated task-specific LoRA adapter (Hu et al., 2021) for each task that is only activated when facing the corresponding task, thereby preventing interference in gradient updates. The number of adapter weights grows linearly with the number of tasks.

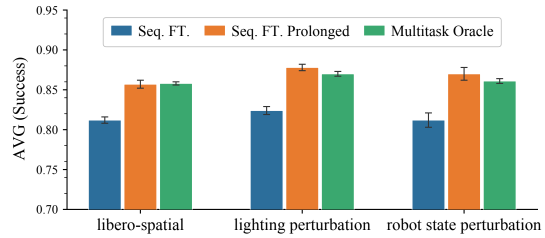



Fig. 4: Final training success rates: by simply prolonging the Seq. FT training steps, we can obtain on-par performance with multitask oracle.

In addition to classical CRL methods, we evaluate two additional methods motivated by recent advances in large pretrained models:

- **SLCA (Zhang et al., 2023)**: a method for layerwise learning-rate decoupling, by applying higher learning rates to action head and lower rates to the VLM trunk, in an effort to preserve the pre-trained representations of the base VLA model.
- **RETAIN (Yadav et al., 2025)**: after training on each task, RETAIN merges the delta weight update back into the base model with a discount coefficient, instead of fully accepting it. RETAIN represents model-merging approaches designed to balance adaptation and retention in weight space without explicit replay or importance estimation.

Together, these methods span the dominant CRL paradigms as well as emerging large-model adaptation strategies.

## Appendix F
## Closing the Training Gap Between the Multi-task Oracle and Sequential Fine-Tuning

In our experiments, we noted that there is a small but consistent gap between multi-task training and continual learning on the training task average success. While it is understandable that CRL methods would under-perform the oracle, in this section we seek to investigate whether this gap is introduced by fundamental limitations of the CRL setup that caused the agent to converge to sub-optimal local optima. Specifically, we examine this question in the three domains where **the gap between the Seq. FT and the multitask oracle is largest** (around 5%). We test whether we can bridge this gap by simply doubling the number of training episodes on the lowest performing task in each of these benchmarks, and report the results in Fig. 4.

As shown by these results, **we can close this gap and reach on-par AVG with the multitask oracle simply by training for more episodes**. These results indicate that the AVG gap is not due to Seq. FT getting stuck at sub-optimal solutions. Instead, they highlight two insights: first, multi-task training may introduce synergies that improve sample efficiency, which is an intriguing direction for future study; second, if the goal is to match multi-task performance, Sequential Fine-Tuning can achieve it by simply training for more episodes on the lower-performing tasks.

## APPENDIX G
## EXPERIMENT SETUP

Each of our base models are obtained by performing supervised finetuning (SFT) with a small amount of in-domain data, so that the model has non-zero initial success rate. This setup allows us to examine performance across a range of initial policy qualities and verify that our results are not specific to a single checkpoint. Here we provide the detailed experiment setup across different benchmarks.

For ManiSkill, we standardize the plate, background, and table color and restrict the variation in initial states to 40 discrete object positions and 4 object rotations. This reduces evaluation variance and ensures that all methods are evaluated on the same fixed set of task configurations, improving the comparability and reproducibility of results. We opt to use 4 training tasks which allows us to run multiple seeds and baselines while keeping the total experimental budget tractable.

We select training tasks whose initial success rates are neither near zero nor saturated. Tasks with non-zero initial performance ensure that the base model already possesses some relevant capabilities, allowing CRL to refine existing behaviors rather than learning entirely from scratch. Avoiding tasks with near-saturated performance leaves sufficient headroom for improvement, making it possible to meaningfully evaluate learning throughout training.

## APPENDIX H
## ABLATION SETUP

We provide additional details for the ablation experiments used in the libero-spatial benchmark. Unless otherwise specified, all ablations use the same environment setup, evaluation protocol, and shared hyperparameters described in Appendix I.

**SFT instead of RL** replaces online RL post-training with supervised fine-tuning on a dataset of demonstration trajectories collected from the environment. The policy is fine-tuned via behavior cloning on 432 demonstration trajectories using the same base model and input representation as the RL setup.

**Smaller Policy** replaces the OpenVLA-OFT model with a small CNN policy of around 12M parameters. The policy is initially supervised finetuned on 30 demonstrations to prime the model with non-zero success and RL finetuned using the same setup as Seq. FT.

**Without LoRA** performs RL post-training on the full OpenVLA model without parameter-efficient LoRA adapters, instead updating the base model parameters directly. All other RL hyperparameters remain identical to the main experimental setup.

## APPENDIX I
## SHARED HYPERPARAMETER

Here we present hyperparameters for the shared components of VLA post-training. These settings are used across all tasks unless otherwise specified. We adopt GRPO as the base algorithm and LoRA adapters with rank 32. Other hyperparameters follow the standard configuration listed below.

TABLE VI: Hyperparameters for RL post-training.

| Algorithm | Name | Value |
|---|---|---|
| GRPO | Optimizer | AdamW |
| | Learning rate | $2 \times 10^{-5}$ |
| | AdamW $\beta_1$ | 0.9 |
| | AdamW $\beta_2$ | 0.999 |
| | Adamw $\epsilon$ | $10^{-5}$ |
| | Gradient clip norm | 1.0 |
| | Global batch size | 8192 |
| | Discount $\gamma$ | 0.99 |
| | GAE $\lambda$ | 0.95 |
| | Clip ratio (low/high) | 0.20 / 0.28 |
| | KL coefficient $\beta$ | 0.0 |
| | Entropy bonus | 0.0 |
| | Rollout epochs | 16 |
| | Group size | 8 |
| | LoRA rank | 32 |

## APPENDIX J
## METHOD HYPERPARAMETERS

This table summarizes the method-specific hyperparameters used for each continual learning algorithm in our experiments. Sequential Fine-Tuning, Dynamic Weight Expansion, and multitask training are omitted, as they do not introduce any additional hyperparameters beyond those shared across all experiments.

TABLE VII: Algorithm-specific hyperparameters.

| Algorithm | Name | Value |
|---|---|---|
| EWC | regularization coefficient $\lambda$ | $1 \times 10^6$ |
| | fisher estimation samples | 65536 |
| ER | Replay loss weight $\lambda_{\text{replay}}$ | 0.03 |
| | Replay # trajectories | 10 |
| | Replay global batch size | 8192 |
| DER | Replay loss weight $\lambda_{\text{replay}}$ | 0.03 |
| | Replay # trajectories | 10 |
| | Replay global batch size | 8192 |
| SLCA | slow learning rate | $4 \times 10^{-6}$ |
| | fast learning rate | $4 \times 10^{-5}$ |
| RETAIN | merge coefficient $\lambda$ | 0.5 |

## APPENDIX K
## ENVIRONMENT DESCRIPTION

In this section, we describe the environments used in our experiments, including task visualizations, natural language instructions, and the corresponding train-test splits.

### A. *Libero-Spatial*

#### 1) *Training Tasks:*

1)  pick up the black bowl between the plate and the ramekin and place it on the plate

TABLE IV: Experiment setup across benchmarks.

| Parameter | libero-object | libero-spatial | libero-long-horizon | RoboCasa | maniskill |
|---|---|---|---|---|---|
| Base Model | OpenVLA-OFT | OpenVLA-OFT | OpenVLA-OFT | Pi-0 | OpenVLA |
| # of SFT Demos | 10 | 10 | 432 | 240 | 140 |
| Initial Training Success | 55.6 | 56.9 | 83.0 | 18.9 | 51.6 |
| # of Training Tasks | 5 | 5 | 5 | 4 | 4 |
| Episodes per Task | 10240 | 10240 | 5120 | 3840 | 10240 |
| Episode Length | 512 | 512 | 512 | 480 | 80 |

TABLE V: Ablation setup for libero-spatial benchmark.

| Parameter | SFT instead of RL | Smaller Policy | Without LoRA |
|---|---|---|---|
| Base Model | 7B OpenVLA-OFT | 12M CNN with MLP head | 7B OpenVLA-OFT |
| # of Training Tasks | 5 | 5 | 5 |
| Pre-training Demos | 10 | 30 | 10 |
| Initial Training Success | 56.9 | 66.8 | 56.9 |
| Batch Size | 256 | 8192 | 8192 |
| RL Episodes per Task | - | 10240 | 10240 |
| SFT Dataset Demos | 432 | - | - |
| SFT Training Steps | 600 | - | - |

2)  pick up the black bowl next to the ramekin and place it on the plate

3)  pick up the black bowl from table center and place it on the plate

4)  pick up the black bowl on the cookie box and place it on the plate

5)  pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate

### 2) *Held-Out Tasks:*

1)  pick up the black bowl on the ramekin and place it on the plate

2)  pick up the black bowl next to the cookie box and place it on the plate

3)  pick up the black bowl on the stove and place it on the plate

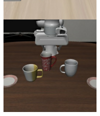
4)  pick up the black bowl next to the plate and place it on the plate

5)  pick up the black bowl on the wooden cabinet and place it on the plate

## B. *Libero-Long*

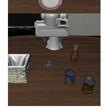
### 1) *Training Tasks:*

1)  put the black bowl in the bottom drawer of the cabinet and close it

2)  put the white mug on the left plate and put the yellow and white mug on the right plate

3)  pick up the book and place it in the back compartment of the caddy

4)  put the white mug on the plate and put the chocolate pudding to the right of the plate

5)  put both the alphabet soup and the cream cheese box in the basket

### 2) *Held-Out Tasks:*

1)  put both the alphabet soup and the tomato sauce in the basket

2)  put both the cream cheese box and the butter in the basket

3)  turn on the stove and put the moka pot on it

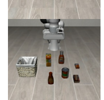
4)  put both moka pots on the stove

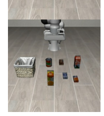
5)  put the yellow and white mug in the microwave and close it

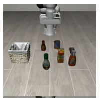
## C. *Libero-Object*

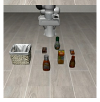
### 1) *Training Tasks:*

1)  pick up the tomato sauce and place it in the basket

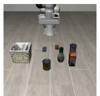
2)  pick up the butter and place it in the basket

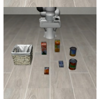
3)  pick up the milk and place it in the basket

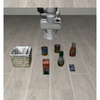
4)  pick up the chocolate pudding and place it in the basket

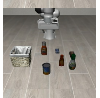
5)  pick up the orange juice and place it in the basket

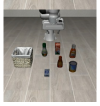
*2) Held-Out Tasks:*

1)  pick up the alphabet soup and place it in the basket

2)  pick up the cream cheese and place it in the basket

3)  pick up the salad dressing and place it in the basket

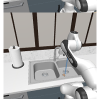
4)  pick up the bbq sauce and place it in the basket

5)  pick up the ketchup and place it in the basket

## D. RoboCasa

*1) Training Tasks:*

1)  turn sink spout

2)  turn on sink faucet

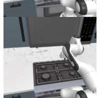
3)  close drawer

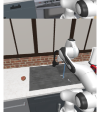
4)  press coffee machine button

*2) Held-Out Tasks:*

1)  close cabinet or microwave door

2)  turn on microwave

3)  turn off microwave

4)  turn off sink faucet

## E. Maniskill Put Plate On Scene 25 Main

*1) Training Tasks:*

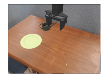
1)  put carrot on plate

2)  put bread on plate

3)  put ketchup bottle on plate

4)  put fast food cup on plate

*2) Held-Out Tasks:*

1)  put watering can on plate

2)  put pipe on plate

3)  put toy bear on plate

4)  put hamburger on plate

## F. Perturb Camera Angle

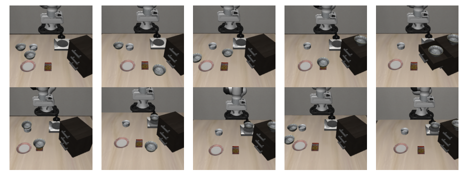


Fig. 5: Changing camera angles.

## G. Perturb Lighting Conditions

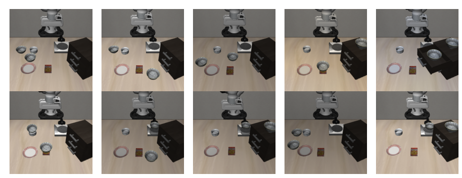


Fig. 6: Changing Lighting conditions.

## H. Perturb Robot Position

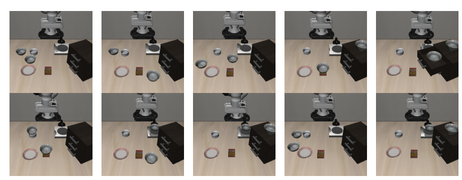


Fig. 7: Changing Robot initial position.