# RARA: ROBUST POLICY FINE-TUNING THROUGH RETRIEVAL-ANCHORED REPRESENTATION ALIGNMENT FOR ROBOTIC MANIPULATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Pre-trained robot policies, such as Vision-Language-Action (VLA) models, provide a strong foundation for generalist robotic manipulation, but adapting them to new tasks with only a few demonstrations remains challenging. Directly fine-tuning a pre-trained policy often achieves strong in-distribution performance, yet it can distort the representations learned during pre-training and weaken the model's generalization ability under out-of-distribution shifts. We propose Retrieval-Anchored Representation Alignment (RARA), a two-stage adaptation method, to address this issue. First, RARA retrieves representation features from pretraining with similar task intent and motion, and aligns the policy's representations of the target demonstrations with the pre-trained representations of the retrieved demonstrations. Second, RARA fine-tunes the policy on the target demonstrations while limiting how much its representations of the retrieved demonstrations change during fine-tuning. By reducing the representation mismatch before action learning and preserving pre-trained knowledge during adaptation, RARA limits unnecessary representation drift during fine-tuning. Experiments with Diffusion Policy and GR00T on 13 tasks in RoboTwin, RoboCasa, and on a real robot show that RARA improves the average success of fine-tuning by 9.5 points in-distribution and 13.6 points out-of-distribution, and outperforms all compared baselines.

## 1 INTRODUCTION

Pre-training on large-scale robot demonstrations followed by fine-tuning on a small target dataset has become a standard practice in learning robot manipulation policies (O'Neill et al., 2024; Khazatsky et al., 2024; Black et al., 2025; Bjorck et al., 2025). By leveraging supervision from large and diverse robot demonstration datasets, pre-training provides useful initial representations that help adaptation to a new task. However, fine-tuning on limited target data can still overfit the policy to the narrow distribution seen during adaptation: A policy may succeed on the target task under the in-distribution conditions, but fail when the camera viewpoint, background, or object properties change (Fei et al., 2026). This vulnerability arises because standard fine-tuning on limited data can severely distort the representations learned during pre-training (Kumar et al., 2022; Kachaev et al., 2026; Yadav et al., 2026). Therefore, *robust policy fine-tuning*, which requires learning a target task while maintaining generalization to out-of-distribution shifts, remains a critical challenge when adapting pre-trained robotic policies to new tasks.

Existing methods have explored several ways to preserve generalizable representations when adapting policies to new tasks. One line of work constrains the adapted policy toward its pre-trained initialization, for example by regularizing fine-tuned weights toward the pre-trained weights (Huang et al., 2026), merging pre-trained and fine-tuned weights (Yadav et al., 2026), or aligning policy representations with those of a frozen pre-trained visual encoder (Kachaev et al., 2026). Another line of work distills knowledge from multiple pre-trained models into a shared representation for downstream robot learning (Shang et al., 2024). While these approaches preserve broad pre-trained knowledge, they do not provide a principled way to focus alignment to pre-training demonstrations that are most relevant to the target task.



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.

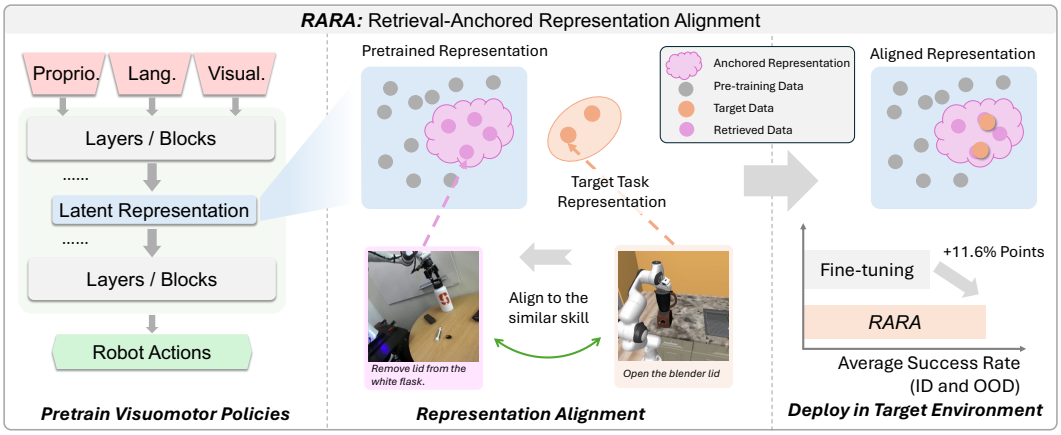



Figure 1: **Overview of RARA.** (Left) A policy is first pre-trained on diverse robot data. (Middle) To adapt it to a target task, RARA first retrieves pre-training demonstrations of a similar skill, such as removing a lid from a flask for the target task of opening a blender lid. It then aligns the target representation with their pre-trained representation, which it keeps as an anchor during fine-tuning. (Right) Across 13 simulated and real-world tasks, RARA improves the average ID and OOD success over fine-tuning by 11.6 points.

We propose **R**etrieval-**A**nchored **R**epresentation **A**lignment (**RARA**), which uses task-relevant pre-training demonstrations as anchors for fine-tuning (Fig. 1). RARA retrieves relevant demonstrations from the pre-training data by matching both task instructions and robot motions to the target demonstrations. For example, demonstrations of removing a pot lid may be retrieved for a target task of opening a blender lid. The retrieved demonstrations support adaptation in two complementary ways. Before fine-tuning, RARA aligns each target frame with retrieved frames at a corresponding stage of the motion (e.g., when the gripper is closing around the lid). The pre-trained policy has seen these stages under many visual conditions, so alignment lets the target frames reuse this robust representation instead of fitting the few target demonstrations from scratch. During fine-tuning, an anchor loss keeps the representations of the retrieved demonstrations close to their pre-trained values, so the small target dataset does not overwrite this robust structure. Because the anchor is applied to retrieved demonstrations rather than to the target observations, it does not restrict task-specific adaptation.

We fine-tune a pre-trained Diffusion Policy on four RoboTwin tasks, and a DROID-pre-trained GR00T policy on five RoboCasa tasks and four real-robot tasks. We evaluate each policy in the conditions covered by its demonstrations (in-distribution, ID) and under new camera poses, backgrounds, distractors, objects or object positions (out-of-distribution, OOD). Across these 13 tasks, RARA improves the average success rates of fine-tuned policies by 9.5 % points in ID and 13.6 % points in OOD settings, consistently outperforming the baseline methods.

Our contributions are threefold: (1) we introduce RARA, a framework for robust policy fine-tuning that uses task-relevant pre-training experience to guide representation learning; (2) we evaluate RARA across different policies, simulated environments, and real-world manipulation tasks, measuring performance of target tasks and robustness to unseen conditions; and (3) we conduct extensive studies that separate initial alignment from representation constraints during fine-tuning, and examine how the choice of source data and network layer affects adaptation.

## 2 Related Work

**Robust fine-tuning.** The goal of robust policy fine-tuning is to learn a downstream task without erasing the generalization acquired during pre-training (Yadav et al., 2026; Grover et al., 2025; Hancock et al., 2026; Yang et al., 2026). In transfer learning, Kumar et al. (2022) show that full fine-tuning can distort pre-trained features and hurt OOD robustness, motivating staged or constrained updates. In robot learning, one line of work freezes parts of the model and adds lightweight trainable modules, such as adapters or task-specific heads, to preserve existing capabilities while specializing to new tasks (Liang et al., 2022; Sharma et al., 2023; Liu et al., 2024; Xu et al., 2023).

Another line preserves the capabilities of pre-trained policy more directly: RETAIN merges pre-trained and fine-tuned policy weights (Yadav et al., 2026), and ReVLA gradually merges the visual backbone back toward its original vision foundation model (Dey et al., 2025); GenVLA keeps a frozen visual encoder with an additional trainable branch (Grover et al., 2025); MAPS constrains module-wise parameter drift (Huang et al., 2026); AMF ensembles a fine-tuned policy with a frozen prior (Bagatella et al., 2025); and PriorVLA uses a frozen prior expert to guide an adaptation expert (Guo et al., 2026). These methods preserve the pre-trained policy through its architecture or weights. RARA instead acts on the representation: it aligns the target representation with that of retrieved pre-training demonstrations, and then limits how much the latter changes during fine-tuning, without restricting the architecture or which weights are updated.

**Domain adaptation.** Robot policies often face domain gaps between simulation and the real world, different embodiments, or pre-training and downstream deployment environments. Domain adaptation methods therefore seek task-relevant representations or data that transfer across such shifts. For VLA models, Kachaev et al. (2026) identify degradation of visual representations after action fine-tuning and preserve visual grounding through alignment with a frozen vision teacher, while Luo et al. (2026) show that action generation can become insufficiently sensitive to relevant visual tokens in deeper layers. In sim-and-real co-training, Cheng et al. (2025) align joint observation-action distributions with optimal-transport-inspired objectives, while Lei et al. (2026) use adversarial training to maintain an appropriate separation between sim and real domains in the learned representation space. Data selection offers a complementary route: Dass et al. (2026) select useful data from a pre-training set for a target imitation-learning task, improving the adapted policy with task-relevant source examples. Similarly, RARA selects task-relevant data from the pre-training set, but it uses the selected demonstrations as anchors for representation alignment rather than as additional training data: it aligns the target representation with that of the selected demonstrations and limits how much their representation changes during fine-tuning.

## 3 Problem Definition

We consider adapting a visuomotor policy $\pi_{\theta_{\text{pre}}}$, pre-trained on a large and diverse set of robot demonstrations $D_{\text{pre}}$, to a downstream target task with a limited target demonstration set $D_{\text{tgt}} = \{(o_i, a_i)\}_{i=1}^{N}$. Here $o_i$ denotes the full observation available to the policy, including visual inputs and any other language or state information, and $a_i$ denotes the demonstrated action. Standard target-only fine-tuning updates the policy by behavior cloning,

$$\theta_{\text{ft}} = \arg\min_{\theta} \mathbb{E}_{(o,a)\sim D_{\text{tgt}}} \left[ \mathcal{L}_{BC}\big(\pi_\theta(o), a\big) \right], \tag{1}$$

where $\mathcal{L}_{BC}$ is the behavior cloning loss. While this objective directly optimizes performance on the in-distribution (ID) target data, a robust policy must also remain reliable under out-of-distribution (OOD) shifts after adaptation. Therefore, robust fine-tuning aims to learn a policy that achieves high expected task success under both ID and OOD evaluation distributions:

$$\max_{\theta} \; J_{\text{ID}}(\pi_\theta) + J_{\text{OOD}}(\pi_\theta), \qquad J_{\mathcal{E}}(\pi_\theta) = \mathbb{E}_{\tau\sim p_{\mathcal{E}}(\pi_\theta)} \left[ S(\tau) \right], \quad \mathcal{E} \in \{\text{ID}, \text{OOD}\}, \tag{2}$$

where $S(\tau) \in \{0, 1\}$ indicates task success for a trajectory $\tau$, and $p_{\mathcal{E}}$ denotes the trajectory distribution induced by the policy $\pi_\theta$ in evaluation setting $\mathcal{E}$.

## 4 RARA: Retrieval-Anchored Representation Alignment

RARA adapts a pre-trained policy to a target task in two stages. It first aligns the target representation to the pre-trained representation to obtain a good initial representation, and then fine-tunes the policy on the target task while constraining representation shifts of pretraining tasks. Both stages use demonstrations retrieved from the pre-training data as a reference (§4.1).

**Stage 1: Aligning to the pre-trained representation.** Before fully fine-tuning the policy, Stage 1 obtains an initial representation of the target observations by aligning it to the pre-trained representation. Specifically, it aligns the representation of each target frame with the pre-trained representation of retrieved frames at the matched steps of the task (Fig. 2). An anchor loss limits how much this alignment changes the representation of the retrieved demonstrations. Only the layers that produce the representation and a projector that maps pooled representation into an embedding space are trained in this stage.

**Stage 2: Fine-tuning on the target task.** Stage 2 fine-tunes the policy on the target demonstrations with its behavior cloning loss. The anchor loss stays active, so fine-tuning preserves the pre-trained representation on the retrieved demonstrations while the policy adapts to the target task. The projector is discarded, and only the fine-tuned policy is used at inference.

Formally, the two stages solve

$$\text{Stage 1:} \quad \min_{\theta,\,\psi} \; \mathcal{L}_{\text{contrastive}} + \lambda_1 \, \mathcal{L}_{\text{anchor}}, \tag{3}$$

$$\text{Stage 2:} \quad \min_{\theta} \; \mathcal{L}_{\text{BC}} + \lambda_2 \, \mathcal{L}_{\text{anchor}}, \tag{4}$$

where $\mathcal{L}_{\text{BC}}$ is the behavior-cloning loss on the target demonstrations (Eq. 1), $\mathcal{L}_{\text{contrastive}}$ aligns target and retrieved representations, $\mathcal{L}_{\text{anchor}}$ limits the change of the retrieved representations, and $\psi$ are the projector parameters. In Stage 1, $\theta$ includes only the layers that produce the anchored representation; in Stage 2, the trainable parameters follow the standard fine-tuning recipe of the policy. The rest of this section describes retrieval (§4.1) and the two losses (§4.2).

### 4.1 Retrieving Relevant Pre-training Demonstrations

RARA uses pre-training demonstrations relevant to the target task as a reference for the policy's representation update. We define relevance at two levels. A relevant *demonstration* performs the same type of task as the target, even though the objects and scenes may differ; we find such demonstrations by instruction similarity. A relevant *frame* is at the matched step of the task as a target frame; we find such frames by comparing action chunks. Both levels are necessary. Fig. 3 shows an example. Removing a pot lid and unstacking tissue rolls both lift an object upward, so their action chunks are similar even though the tasks are different. Action similarity alone would pair a target frame with a frame of a different task.




Figure 3: Pre-training demonstration retrieval for the target task: Open Blender Lid. Selecting by action similarity alone can return similar action chunks from a different task (red); retrieving by the task instruction returns the same kind of task (green).

**Retrieving demonstrations.** We embed task instructions with a frozen CLIP text encoder (Radford et al., 2021). Because a target task can have several instructions, we average their normalized embeddings. We then rank the pre-training demonstrations by the cosine similarity between their instruction embedding and this average, and keep the top $K$ demonstrations as $D_{\text{ret}} \subseteq D_{\text{pre}}$.

**Pairing frames by action chunks.** We split every demonstration in $D_{\text{ret}}$ and $D_{\text{tgt}}$ into overlapping action chunks $A_t = [a_t, \ldots, a_{t+H-1}]$ and represent each chunk by the observation $o_t$ at its first step. Let $\delta(A, A')$ be the dynamic-time-warping (DTW) distance between two normalized action chunks, which matches similar actions executed at different speeds. For a target frame $o_i^{\text{tgt}}$ with action chunk $A_i^{\text{tgt}}$, the positives $\mathcal{P}(i)$ are the frames of the $m_+$ retrieved chunks closest to $A_i^{\text{tgt}}$ under $\delta$, and the negatives $\mathcal{N}(i)$ are frames of retrieved chunks sampled from those farther from $A_i^{\text{tgt}}$ than the median distance. RARA uses the actions of the retrieved demonstrations only to form these pairs, not as behavior-cloning targets.

### 4.2 Aligning an Intermediate Representation

Let $h_\theta^\ell(o)$ denote the representation at layer $\ell$ of policy $\pi_\theta$, and let $\bar{h}^\ell(o) = h_{\theta_{\text{pre}}}^\ell(o)$ be the corresponding representation from a frozen copy of the pre-trained policy. The observation $o$ includes the policy's image, language and state inputs, as applicable.

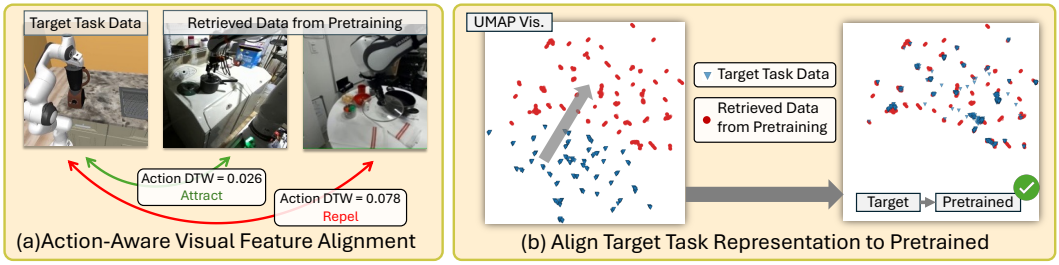



Figure 2: **Stage 1 of RARA.** (a) For a target frame, frames of the retrieved demonstrations with similar action chunks (small DTW distance) are pulled closer, and frames with different action chunks are pushed away. (b) UMAP of the representations before (left) and after (right) Stage 1: the target representations (blue) move toward those of the retrieved pre-training data (red).

**Choosing the anchored layer.** We call $\ell$ the *anchored layer* and $h_\theta^\ell$ the *anchored representation*; both the contrastive loss and the anchor loss act on this representation. The anchored layer can be any layer of the policy, from an early perception layer to a layer inside the action network, so RARA does not depend on a particular policy architecture. In practice, the target data usually differ from the pre-training data mainly in visual appearance. In this case, any layer that follows the visual input can serve as the anchored layer. We keep $\ell$ general below; §5.1 gives the layer used in our experiments, and RQ3 (§5.4) shows that the gain is not tied to one layer.

**Matching target representations to source references.** For a target observation $o_i^{\text{tgt}}$, let $\mathcal{P}(i)$ and $\mathcal{N}(i)$ be its positive and negative frames from $D_{\text{ret}}$ (§4.1). We compare representations in a contrastive embedding space. The representation is pooled over tokens or cameras as needed and mapped to an $\ell_2$-normalized embedding by a trainable two-layer MLP projector $g_\psi$. Following common practice in contrastive learning, we compute the loss after this projector rather than directly on the representation, which has been shown to improve the learned representation and to reduce its dimensional collapse (Chen et al., 2020; Jing et al., 2022) (see Appx. E for details). The embedding of the target frame is computed with the adapting policy, $z_i^{\text{tgt}} = g_\psi(h_\theta^\ell(o_i^{\text{tgt}}))$, and the embedding of a retrieved frame $o_j$ is computed with the frozen pre-trained policy, $\bar{z}_j = g_\psi(\bar{h}^\ell(o_j))$. A multi-positive contrastive loss (Oord et al., 2018; Khosla et al., 2020; Miech et al., 2020) then aligns each target embedding with the embeddings of its positive frames (Fig. 2):

$$\mathcal{L}_{\text{contrastive}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\sum_{p\in\mathcal{P}(i)}\exp\left(\text{sim}(z_i^{\text{tgt}},\bar{z}_p)/\tau\right)}{\sum_{p\in\mathcal{P}(i)}\exp\left(\text{sim}(z_i^{\text{tgt}},\bar{z}_p)/\tau\right)+\sum_{n\in\mathcal{N}(i)}\exp\left(\text{sim}(z_i^{\text{tgt}},\bar{z}_n)/\tau\right)}, \tag{5}$$

where $B$ is the target batch size, $\text{sim}(u,v) = u^\top v/(\|u\|\,\|v\|)$ is cosine similarity, and $\tau$ is the temperature. The pre-trained policy is frozen, so the representations $\bar{h}^\ell$ of the retrieved frames stay fixed and serve as the reference for the anchor loss below; only their embeddings $\bar{z}_j$ change as the projector is trained. The projector is used only for the contrastive loss and is discarded after Stage 1.

**Anchoring source representations.** Updating the policy to represent the target also changes how it represents source observations. We constrain these changes with

$$\mathcal{L}_{\text{anchor}}^\ell(\theta) = \mathbb{E}_{o\sim D_{\text{ret}}}\left[d\big(h_\theta^\ell(o),\bar{h}^\ell(o)\big)\right], \tag{6}$$

where $d$ measures the discrepancy between the two representations (see Appx. C for details). The loss compares the adapting and the pre-trained policy on the same frame of a retrieved demonstration but uses no action labels. It is computed only on the retrieved demonstrations, so it leaves the target representation free to change under the contrastive loss and, later, the task loss; it only limits how much these changes also alter the representation of the retrieved demonstrations. Together with the contrastive loss, it forms the Stage-1 objective in Eq. 3, and it stays active in Stage 2 (Eq. 4).

## 5 Experiments

We evaluate whether RARA learns a target task while generalizing beyond the conditions covered by its demonstrations, by fine-tuning two pre-trained policies on 13 tasks in simulation and on a real robot (§5.1). We aim to answer the following four research questions:

- **RQ1:** How does RARA change the representation compared with standard fine-tuning? (§5.2)
- **RQ2:** Does RARA improve ID and OOD success over fine-tuning and prior methods? (§5.3)
- **RQ3:** Does the gain depend on the choice of the anchored layer? (§5.4)
- **RQ4:** Which components and data contribute to the gain? (§5.5)

### 5.1 SETUP

We evaluate RARA in two simulated environments and on a real robot (Fig. 4). In each setting, a pre-trained policy is fine-tuned on a small set of target demonstrations and evaluated both in-distribution (ID) and out-of-distribution (OOD). See Appx. D for the full details of setup.

- **RoboTwin 2.0** (Chen et al., 2026). We use four single-arm target tasks, such as Place Container and Move Pill Bottle. ID uses the narrow layout of the target demonstrations; OOD widens the object layout, the backgrounds and the distractors. For pre-training, we collect about 2k demonstrations of various skills in RoboTwin, such as pick-and-place and pressing a bell, holding out the four target tasks. The pre-training data cover a larger workspace and more distractors, objects and background textures than the target demonstrations. Each target task has 50 demonstrations.
- **RoboCasa** (Nasiriany et al., 2026). We use five target tasks, such as Open Blender Lid and Close Fridge Drawer. ID uses the demonstration kitchen with randomized object poses; OOD changes the camera pose, background, distractors and object instances. Pre-training uses DROID (Khazatsky et al., 2024), which contains 76k demonstrations, and each target task has 100 demonstrations collected in one kitchen.
- **Real Franka robot**. We use four target tasks, such as Remove Pot Lid and Load Bread in Toaster. ID and OOD are defined as in RoboCasa. Pre-training uses DROID, and each target task has 50 to 80 demonstrations collected by VR teleoperation.

**Policies.** We evaluate two common visuomotor policies:

- **Diffusion Policy** (Chi et al., 2023), a widely used visuomotor policy, which we pre-train on the RoboTwin pre-training data.
- **GR00T N1.7** (Bjorck et al., 2025), a large vision-language-action (VLA) model pre-trained on DROID, which we use for RoboCasa and the real robot.

Since the target tasks differ from the pre-training data mainly in visual appearance, the anchored layer of both policies is the output of the layers that encode the visual input. All methods in a setting use the same optimizer settings and training budget. See Appx. C for details.

**Evaluation Protocol.** We evaluate each policy on 150 rollouts per task and setting in RoboTwin, 50 in RoboCasa and 10 trials on the real robot, and repeat each experiment three times (see Appx. D for details). On the real robot, all methods are tested on the same recorded scene randomizations. We report the mean and standard deviation of the success rate.

**Baselines.** We compare RARA with five baselines. They cover the two common ways to reuse pre-training knowledge during fine-tuning: training on pre-training data (Co-FT and STRAP) and keeping the policy close to the pre-trained one (RETAIN and BlindVLA). **FT** fine-tunes the policy on the target demonstrations only. **Co-FT** also trains on the retrieved pre-training demonstrations $D_{\mathrm{ret}}$ (§4.1) with the behavior-cloning loss. **STRAP** (Memmel et al., 2025) instead retrieves pre-training sub-trajectories that match the target demonstrations under dynamic time warping, and co-trains on them. **RETAIN** (Yadav et al., 2026) merges the weights of the pre-trained and fine-tuned policies. **BlindVLA** (Kachaev et al., 2026) aligns the policy's visual representations with those of a frozen pre-trained visual encoder during fine-tuning. See Appx. D.2 for details.

### 5.2 RQ1: HOW DOES RARA CHANGE THE REPRESENTATION?

Anchoring changes the representation of the retrieved pre-training frames as little as possible by design. RQ1 asks whether RARA also keeps the representation of target-task frames, on which the anchor is never applied, close to that of the pre-trained policy $P$, and whether it keeps the structure that $P$ learned in pre-training. We compare the anchored representation of each adapted policy with that of $P$ on Move Pill Bottle and Place Container in RoboTwin, whose objects do not appear in the pre-training data (Table 1; see Appx. D.3 for definitions). The left part of Table 1 is computed on

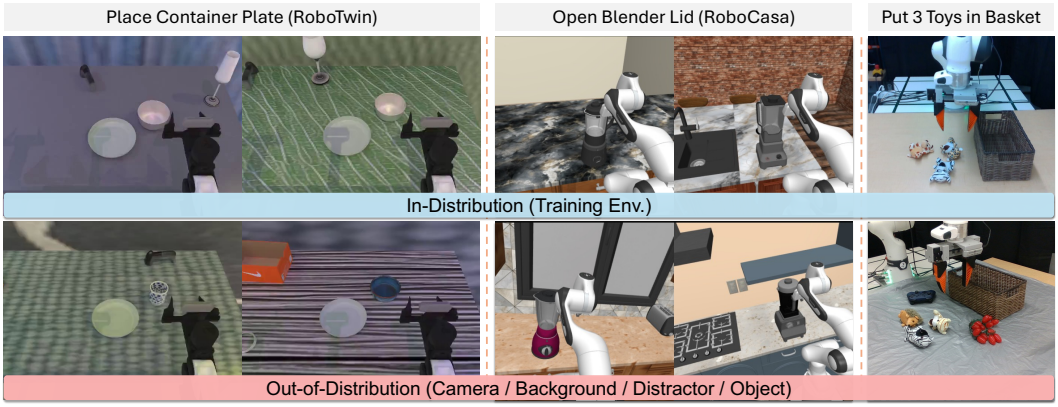



Figure 4: ID (top) and OOD (bottom) scenes for one RoboTwin, one RoboCasa and one real-robot task.

Table 1: Representation metrics at the anchored layer on RoboTwin (mean over five pre-training seeds). Left: computed on held-out pre-training frames; higher values mean that more of the pre-trained structure remains. Right: feature distortion on held-out frames of the target task in the ID and OOD conditions; lower values mean smaller changes from the pre-trained policy $P$. Bold marks the best adapted value in each column. "After Stage 1" is the policy before Stage-2 fine-tuning.

| | Pre-training feature retention | | | | | | Target-data feature distortion | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Move pill bottle | | | Place container | | | Move pill bottle | | Place container | |
| Recipe | Rank ↑ | CKA ↑ | $R^2$ ↑ | Rank ↑ | CKA ↑ | $R^2$ ↑ | ID ↓ | OOD ↓ | ID ↓ | OOD ↓ |
| Pre-trained $P$ | 58.8 | 1.00 | 0.71 | 58.8 | 1.00 | 0.71 | 0.00 | 0.00 | 0.00 | 0.00 |
| FT | 50.2 | 0.68 | 0.67 | 53.1 | 0.87 | 0.64 | 0.50 | 0.49 | 0.47 | 0.48 |
| Co-FT | 51.4 | **0.93** | 0.72 | 49.1 | **0.94** | 0.72 | 0.57 | 0.57 | 0.62 | 0.63 |
| RARA, after Stage 1 | **73.2** | 0.70 | **0.74** | **68.6** | 0.76 | **0.75** | 0.48 | 0.46 | 0.56 | 0.53 |
| RARA | 54.5 | 0.92 | 0.71 | 55.7 | 0.91 | 0.70 | **0.23** | **0.23** | **0.31** | **0.33** |

held-out pre-training frames, and higher values mean that more of the pre-trained structure remains: *effective rank* (Garrido et al., 2023) measures how many directions the representation uses, *CKA to* $P$ (Kornblith et al., 2019) measures its similarity to that of $P$, and the *position-probe* $R^2$ (Davari et al., 2022) measures how well object positions can be read from it. The right part reports *feature distortion* (Kumar et al., 2022) on frames from held-out episodes of the target task, in the ID and OOD conditions; it measures how far adaptation moves the representation from that of $P$, and lower values mean smaller changes.

**RARA keeps the representation of target-task frames close to that of $P$.** On both tasks and in both ID and OOD conditions, FT and Co-FT change the representation of target-task frames by 0.47–0.63, whereas RARA changes it by only 0.23–0.33, although the anchor is never applied to these frames. On pre-training frames, FT lowers the rank, CKA and probe $R^2$ on both tasks, whereas RARA stays close to $P$ (CKA 0.91–0.92, $R^2$ 0.70–0.71). Co-FT, which trains on the retrieved demonstrations with their actions, also keeps a high CKA (0.93–0.94), but it changes the representation of target-task frames the most, and its success is similar to or lower than that of FT (Table 2). Keeping the structure of the pre-training frames is therefore not sufficient; RARA also keeps the target-task representation close to that of $P$, while improving success on both tasks.

**Stage 1 aligns the representation and the Stage-2 anchor keeps it close to $P$.** Right after Stage 1, the representation of target-task frames is about as far from that of $P$ as after FT (distortion 0.46–0.56, row "RARA, after Stage 1"). Stage 1 does not keep the representation close to $P$; it aligns the target-task frames with the retrieved frames. Fine-tuning with the anchor in Stage 2 then reduces the distortion to 0.23–0.33. RQ4 shows that both stages are needed for the full gain.

Table 2: Simulation results (success rate, %, mean ± std). RoboTwin uses the Diffusion Policy; RoboCasa uses GR00T N1.7 pre-trained on DROID. Bold: best; underline: second best.

| **RoboTwin** | Place Container | | Move Pill Bottle | | Click Alarm Clock | | Shake Bottle | | Task average | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | ID | OOD | ID | OOD | ID | OOD | ID | OOD | ID | OOD |
| FT | 82.0 ±0.7 | 60.7 ±2.4 | 32.9 ±3.7 | 9.6 ±2.3 | 80.4 ±6.3 | 52.4 ±4.1 | 74.2 ±2.1 | 66.4 ±5.7 | 67.4 ±1.4 | 47.3 ±2.4 |
| Co-FT | 62.7 ±5.8 | 43.3 ±3.3 | 32.4 ±8.1 | 10.0 ±2.4 | **81.8** ±3.4 | 61.3 ±1.8 | 80.7 ±6.8 | 75.6 ±5.7 | 64.4 ±2.8 | 47.6 ±1.3 |
| STRAP | 81.1 ±3.8 | 59.3 ±2.4 | 44.4 ±5.1 | 13.8 ±1.4 | 81.3 ±1.3 | 61.6 ±1.7 | 83.6 ±7.3 | 74.9 ±7.7 | 72.6 ±2.0 | 52.4 ±2.3 |
| RETAIN | 55.3 ±10.1 | 40.7 ±4.2 | 29.3 ±6.1 | 13.3 ±3.1 | 53.3 ±6.1 | 36.0 ±4.0 | 50.0 ±6.0 | 46.7 ±6.4 | 47.0 ±6.7 | 34.2 ±2.3 |
| BlindVLA | 69.3 ±3.1 | 52.9 ±2.0 | 47.6 ±1.4 | 15.8 ±0.8 | 74.0 ±6.7 | 46.9 ±1.0 | 80.7 ±4.8 | 73.3 ±2.9 | 67.9 ±1.0 | 47.2 ±0.8 |
| RARA | **86.7** ±2.0 | **63.3** ±4.2 | **51.6** ±2.5 | **23.8** ±1.4 | 80.4 ±3.3 | **64.2** ±2.7 | **85.3** ±0.7 | **82.0** ±2.4 | **76.0** ±1.0 | **58.3** ±1.2 |

| **RoboCasa** | Close Fridge Drawer | | Open Blender Lid | | Open Cabinet | | Turn On Toaster | | Pick-Place to Drawer | | Task average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | ID | OOD | ID | OOD | ID | OOD | ID | OOD | ID | OOD | ID | OOD |
| FT | 86.0 ±8.7 | 62.7 ±3.1 | 76.0 ±3.5 | 44.7 ±9.0 | 72.0 ±2.0 | 44.0 ±10.6 | 72.0 ±5.3 | 46.0 ±6.0 | 40.0 ±2.0 | 24.0 ±8.0 | 69.2 ±2.5 | 44.3 ±2.0 |
| Co-FT | 90.7 ±3.1 | 59.3 ±9.5 | 72.0 ±5.3 | 48.0 ±5.3 | 66.7 ±9.0 | 47.3 ±2.3 | 66.7 ±11.0 | 42.0 ±3.5 | 39.3 ±1.2 | 22.0 ±3.5 | 67.1 ±0.5 | 43.7 ±2.0 |
| STRAP | 84.0 ±8.7 | 66.7 ±3.1 | 73.3 ±1.2 | 46.7 ±3.1 | 68.7 ±1.2 | 52.0 ±5.3 | 66.7 ±6.1 | 47.3 ±7.0 | 38.0 ±5.3 | 28.0 ±7.2 | 66.1 ±3.7 | 48.1 ±3.2 |
| RETAIN | 33.3 ±5.8 | 22.0 ±6.0 | 13.3 ±7.6 | 9.3 ±9.5 | 46.7 ±3.1 | 19.3 ±4.2 | 46.0 ±2.0 | 20.0 ±2.0 | 16.0 ±3.5 | 3.3 ±3.1 | 31.1 ±1.8 | 14.8 ±2.9 |
| BlindVLA | 89.3 ±1.2 | 68.0 ±4.0 | 72.0 ±12.0 | 50.7 ±1.2 | 71.3 ±6.1 | 47.3 ±9.0 | 68.7 ±16.2 | 44.7 ±7.0 | 41.3 ±2.3 | 23.3 ±5.0 | 68.5 ±5.4 | 46.8 ±1.1 |
| RARA | **92.0** ±4.0 | **76.0** ±6.9 | **80.7** ±3.1 | **56.7** ±9.0 | **78.0** ±5.3 | **55.3** ±3.1 | **74.0** ±4.0 | **52.7** ±4.6 | **45.3** ±3.1 | **36.7** ±16.0 | **74.0** ±1.4 | **55.5** ±4.8 |

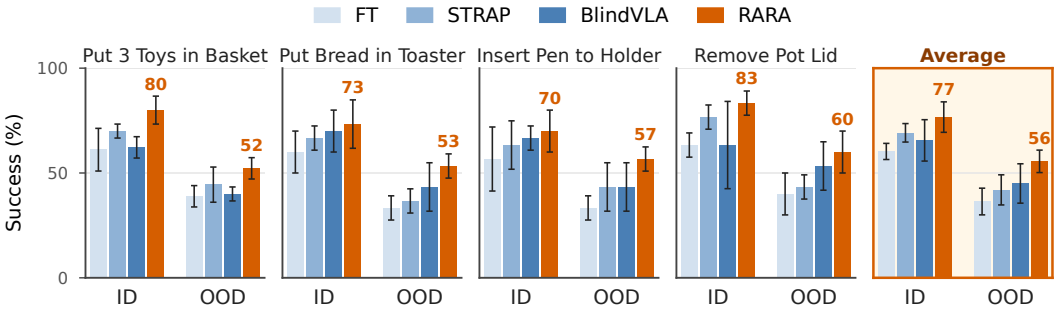



Figure 5: Real-robot results (success rate, %, mean ± std over three runs) with GR00T N1.7 pre-trained on DROID. RARA outperforms baselines in all target tasks, ID and OOD settings.

### 5.3 RQ2: Does RARA Improve ID and OOD Success?

This experiment tests the main claim of the paper: keeping the representation close to retrieved pre-training data should improve success under distribution shift, not only on the configuration seen in the demonstrations. We therefore report ID and OOD success separately (Table 2 for simulation and Fig. 5 for the real robot).

RARA achieves the highest task-average success in ID and in OOD in all three settings. Over FT, it gains 8.6 (ID) and 11.0 (OOD) points on RoboTwin, 4.8 and 11.2 points on RoboCasa, and 16.4 and 19.2 points on the real robot. In all three settings, the gain is larger in OOD than in ID, which is the goal of robust fine-tuning. The baselines, which reuse pre-training knowledge in other ways, improve less. Co-FT and STRAP, which train on pre-training demonstrations with their actions, gain at most 5.5 points in OOD, and Co-FT does not improve over FT in simulation. RETAIN, which merges weights, falls well below FT in simulation, and BlindVLA, which aligns with a frozen visual encoder, gains at most 2.5 points in simulation and 8.6 points on the real robot. Since these baselines differ from RARA in more than one respect, we evaluate the contribution of each component of RARA separately in RQ4.

### 5.4 RQ3: Does the Choice of Anchored Layer Matter?

RARA aligns and anchors the representation at a single anchored layer, and the method does not fix which layer this is (§4.2). This experiment asks whether the choice matters. If only a few layers worked, a user would first have to search for the right layer before applying RARA.

We place the anchored layer at eight locations in the Diffusion Policy on Move Pill Bottle: the four ResNet stages, the encoder output (our default), and three places in the action head (the FiLM conditioning, the middle block and the output), with $\lambda_1 = \lambda_2 = 1$.

Every location improves over FT (Fig. 6): success rises from 21.2% to between 30.0% and 39.7%, a gain of 9 to 18 points. Some locations work better than others, but none of them comes close to FT. On this task, the gain therefore does not depend on one particular layer.

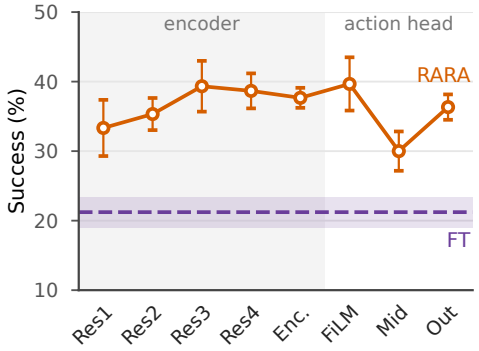



Figure 6: Choice of anchored layer (Move Pill Bottle, $\lambda_1 = \lambda_2 = 1$): mean of ID and OOD success. Bars and band: std over three checkpoints.

### 5.5 RQ4: Which Components and Which Data Produce the Gain?

We run an ablation study on RoboTwin, averaged over its four tasks, and on Open Blender Lid in RoboCasa (Table 3). We compare RARA with five variants. *Stage 1 only* fine-tunes without the anchor after Stage 1, and *Stage 2 only* skips Stage 1. *Co-FT in Stage 2* adds behavior cloning on the retrieved demonstrations to Stage 2. *Random pre-training data* replaces the retrieved demonstrations with a random subset of the pre-training data of the same size. *Random Stage-1 pairs* pairs target and retrieved frames at random in Stage 1, instead of by action-chunk similarity.

Both stages are needed. On RoboTwin, Stage 1 only is no better than FT, and Stage 2 only recovers most of the ID gain (73.0% vs. 76.0%) but none of the OOD gain (46.2%, vs. 47.3% for FT and 58.3% for RARA). On Open Blender Lid, each stage alone recovers part of the OOD gain, and only the full method reaches 56.7%. This is consistent with RQ1: Stage 1 aligns the representation, and the anchor in Stage 2 keeps it close to that of the pre-trained policy during fine-tuning. Adding behavior cloning on the retrieved demonstrations (Co-FT in Stage 2) lowers success in all four columns, so the retrieved demonstrations are most useful as an anchor for the representation.

Table 3: Ablations of RARA (success rate, %, mean ± std). RoboTwin: average over its four tasks. Bold: best; underline: second best.

| | RoboTwin (avg.) | | Open Blender Lid | |
|---|---|---|---|---|
| Method | ID | OOD | ID | OOD |
| FT | 67.4 ±1.4 | 47.3 ±2.4 | 76.0 ±3.5 | 44.7 ±9.0 |
| RARA | **76.0** ±1.0 | **58.3** ±1.2 | **80.7** ±3.1 | **56.7** ±9.0 |
| Random pre-training data | 50.3 ±1.7 | 34.9 ±1.7 | 67.7 ±1.3 | 44.2 ±1.2 |
| Random Stage-1 pairs | 65.3 ±3.1 | 40.6 ±1.2 | 76.7 ±2.3 | 45.5 ±4.2 |
| Co-FT in Stage 2 | 65.8 ±3.3 | <u>48.1</u> ±2.9 | 72.9 ±5.3 | <u>53.6</u> ±3.7 |
| Stage 1 only | 66.6 ±2.2 | 43.2 ±1.3 | <u>77.3</u> ±4.8 | 52.2 ±2.3 |
| Stage 2 only | <u>73.0</u> ±3.2 | 46.2 ±4.1 | 76.0 ±6.0 | 48.0 ±7.2 |

The choice of pre-training data and how its frames are paired also matter. With random pre-training data, RARA performs worse than FT on both RoboTwin and Open Blender Lid, by up to 17.1 points on RoboTwin. With random Stage-1 pairs, RARA falls below FT on RoboTwin (65.3%/40.6% vs. 67.4%/47.3%) and is only on par with FT on Open Blender Lid, so the action-aware pairing is needed. Appx. F further shows that the anchor helps when the pre-training data contain the same skill as the target task, and can hurt when the skill differs.

## 6 Conclusion

We presented RARA, a two-stage method for robust policy fine-tuning. RARA retrieves pre-training demonstrations of the same type of task as the target, first aligns the target representation to their pre-trained representation, and then fine-tunes the policy while anchoring the representation of these demonstrations. Across 13 tasks in simulation and on a real robot, RARA improves the average success of fine-tuning by 9.5 points in ID and 13.6 points in OOD. Our analysis shows that it also keeps the representation of target-task frames closer to that of the pre-trained policy, although the anchor is applied only to the retrieved demonstrations. While being effective, RARA relies on pre-training data that contain the same type of task as the target, and our experiments focus on visual changes with the same robot and action space. Future work includes retrieving useful references when such data are scarce, and extending RARA to new embodiments and action spaces.

### Reproducibility Statement

Section 4 describes the method, and Appx. B, C and D give the implementation and evaluation details. Section 5.1 states the evaluation protocol. The simulated results use the public RoboTwin 2.0 and RoboCasa benchmarks, and the GR00T policy starts from the DROID-pre-trained GR00T N1.7 checkpoint. Code for retrieval, alignment, and evaluation, the retrieved DROID episode lists, and the aligned checkpoints will be released.

### AI Use Statement

We used generative AI tools to write and run experiment code, to organize experimental results, and to edit the paper and format references. All code and results were checked by the authors. We did not use generative AI tools to generate datasets, to design the method or experiments, or to interpret results; theoretical and mathematical tasks are not applicable to this work. We take responsibility for the final content of this work.

## References

Marco Bagatella, Jonas Hübotter, Georg Martius, and Andreas Krause. Active fine-tuning of multi-task policies. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=hlyBdwHBeC.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_0$: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.010.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning (ICML)*, 2026.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607, 2020.

Shuo Cheng, Liqian Ma, Zhenyang Chen, Ajay Mandlekar, Caelan Reed Garrett, and Danfei Xu. Generalizable domain adaptation for sim-and-real policy co-training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ufKaXYJt1F.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Shivin Dass, Alaa Khaddaj, Logan Engstrom, Aleksander Madry, Andrew Ilyas, and Roberto Martín-Martín. DataMIL: Selecting data for robot imitation learning with datamodels. In *International Conference on Learning Representations (ICLR)*, 2026.

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. Reliability of CKA as a similarity measure in deep learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Sombit Dey, Jan-Nico Zaech, Nikolay Nikolov, Luc Van Gool, and Danda Pani Paudel. ReVLA: Reverting visual domain limitation of robotic foundation models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8679–8686. IEEE, 2025.

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 visual-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, 2026.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning (ICML)*, 2023.

Shresth Grover, Akshay Gopalkrishnan, Bo Ai, Henrik I Christensen, Hao Su, and Xuanlin Li. Enhancing generalization in vision-language-action models by preserving pretrained representations. *arXiv preprint arXiv:2509.11417*, 2025.

Xinyu Guo, Bin Xie, Wei Chai, Xianchi Deng, Tiancai Wang, Zhengxing Wu, and Xingyu Chen. PriorVLA: Prior-preserving adaptation for vision-language-action models, 2026. URL https://arxiv.org/abs/2605.10925.

Asher J Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning VLMs into VLAs without catastrophic forgetting. In *International Conference on Learning Representations (ICLR)*, 2026.

Chengyue Huang, Mellon M Zhang, Robert Azarcon, Glen Chou, and Zsolt Kira. MAPS: Preserving vision-language representations via module-wise proximity scheduling for better vision-language-action generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32451–32462, 2026.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Nikita Kachaev, Mikhail Kolosov, Daniil Zelezetsky, Alexey K. Kovalev, and Aleksandr I. Panov. Don't blind your VLA: Aligning visual representations for OOD generalization. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1864–1873. International Foundation for Autonomous Agents and Multiagent Systems, 2026. doi: 10.65109/PPER9186. URL https://doi.org/10.65109/PPER9186.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022.

Yu Lei, Minghuan Liu, Abhiram Maddukuri, Zhenyu Jiang, and Yuke Zhu. A mechanistic analysis of sim-and-real co-training in generative robot policies, 2026. URL https://arxiv.org/abs/2604.13645.

Anthony Liang, Ishika Singh, Karl Pertsch, and Jesse Thomason. Transformer adapters for robot learning. In *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022. URL https://openreview.net/forum?id=H--wvRYBmF.

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. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RRayv1ZPN3.

Yulin Luo, Hao Chen, Zhuangzhe Wu, Bowen Sui, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Qiuxuan Feng, Jiale Yu, Shuo Gu, Peng Jia, Pheng-Ann Heng, and Shanghang Zhang. Look before acting: Enhancing vision foundation representations for vision-language-action models, 2026. URL https://arxiv.org/abs/2603.15618.

Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. STRAP: Robot sub-trajectory retrieval for augmented policy learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9879–9889, 2020.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 2026.

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. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=ylZHvlwUcI.

Mohit Sharma, Claudio Fantacci, Yuxiang Zhou, Skanda Koppula, Nicolas Heess, Jon Scholz, and Yusuf Aytar. Lossless adaptation of pretrained vision models for robotic manipulation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=5IND3TXJRb-.

XPolicyLab Community, Tianxing Chen, et al. XPolicyLab: A unified standard and open ecosystem for robot policy evaluation and deployment. *arXiv preprint arXiv:2608.09892*, 2026.

Mengdi Xu, Yuchen Lu, Yikang Shen, Shun Zhang, Ding Zhao, and Chuang Gan. Hyper-decision transformer for efficient online policy adaptation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AatUEvC-Wjv.

Yajat Yadav, Zhiyuan Zhou, Andrew Wagenmaker, Karl Pertsch, and Sergey Levine. Robust fine-tuning of vision-language-action robot policies via parameter merging. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=uWJwQ5SZoM.

Shuai Yang, Hao Li, Bin Wang, Yilun Chen, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Vision-language-action instruction tuning: From understanding to manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tsxwloasw5.

# A Analysis of Visual Feature Shift

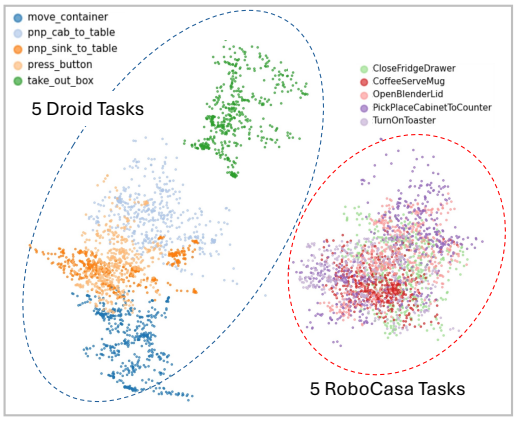



Figure 7: PCA visualization of visual features of ID and OOD tasks.

We discuss visual feature alignment for VLA adaptation. A pre-trained VLA visual encoder can separate diverse source tasks, but downstream image from a new simulator or real setup can be different. Even when the action space, language instruction are similar, such image shifts can leave target features poorly organized for action prediction.

Fig. 7 illustrates this issue. We extract features from a DROID-pretrained $\pi_0$ policy and visualize them with PCA, using five DROID tasks as ID source tasks and five RoboCasa tasks as OOD downstream tasks (Nasiriany et al., 2026); details are in §A.1. DROID features are well separated, while unseen RoboCasa features collapse into a compact region despite representing different manipulation tasks. Thus, the DROID-pretrained visual encoder is not sufficiently task-discriminative in the unseen environment.

Such feature collapse is problematic for policy adaptation. If the visual encoder maps different target states or tasks into a collapsed region, the action model receives weak state information and can easily overfit the few available demonstrations. This motivates visual feature alignment: before learning target actions, we first move target-domain visual features toward a task-relevant region of the pre-trained feature space, making the new observations easier for the pre-trained policy to interpret and providing a better starting point for adaptation.

## A.1 PCA Visualization Details

For the $\pi_0$ feature visualizations, we extract image features from the SigLIP visual encoder before any pooling operation. For each image and each camera view, the encoder outputs 196 patch tokens with feature dimension 1152. We average over the 196 patch tokens to obtain one 1152-dimensional feature vector per camera. The three camera-specific vectors are then concatenated, giving a 3456-dimensional visual feature for each multi-view observation. Stacking features over all selected observations yields a matrix in $\mathbb{R}^{N\times 3456}$. We fit PCA to this feature matrix using SVD and project each observation onto the top two principal components for visualization. When comparing multiple data sources or adaptation checkpoints, we fit PCA once on the combined feature set and use the same projection for all points.

# B Retrieval and Pair Construction Details

This appendix details how RARA retrieves pre-training demonstrations and pairs their frames with target frames (§4.1). The same procedure is used for DROID and for RoboTwin.

**Retrieval by instruction.** Each demonstration comes with a task instruction, and a target task usually has several instructions; on RoboTwin, they are generated from templates. Let $e(g)$ be the

embedding of an instruction $g$ from a frozen CLIP text encoder (Radford et al., 2021). For a target task with instruction set $\mathcal{G}_{\text{tgt}}$, we average the normalized instruction embeddings,

$$\bar{e}_{\text{tgt}} = \frac{1}{|\mathcal{G}_{\text{tgt}}|} \sum_{g \in \mathcal{G}_{\text{tgt}}} \frac{e(g)}{\|e(g)\|_2}. \tag{7}$$

We score each pre-training demonstration by the cosine similarity between the embedding of its instruction and $\bar{e}_{\text{tgt}}$, and keep the top $K = 100$ demonstrations as $D_{\text{ret}}$. Filtering by instruction separates tasks with different intent, such as opening, closing, placing, wiping and insertion, even when their action chunks contain similar end-effector displacements.

**Action chunks.** We represent each demonstration as $\tau = \big(g, \{o_t, q_t, a_t\}_{t=1}^{T}\big)$, where $g$ is the instruction, $o_t$ the image observation, $q_t$ the proprioceptive state, and $a_t$ the action in the action space of the policy. Each demonstration is split into overlapping action chunks

$$A_t = [a_t, a_{t+1}, \ldots, a_{t+H-1}], \qquad t \in \{1, 1+s, 1+2s, \ldots\}, \tag{8}$$

with horizon $H = 16$ and stride $s$, and each chunk is paired with the observation at its first step. Continuous action dimensions are normalized with statistics of the pre-training data, and the gripper dimension is scaled separately so that binary open/close changes do not dominate the translational and rotational dimensions. Target chunks are normalized in the same way.

**DTW distance.** For two action chunks $A$ and $B$, we use the dynamic-time-warping distance

$$\delta(A, B) = \min_{\omega \in \Omega(A,B)} \frac{1}{|\omega|} \sum_{(u,v) \in \omega} \|A_u - B_v\|_2 , \tag{9}$$

where $\Omega(A, B)$ is the set of monotone temporal alignments between the two chunks. DTW matches similar actions that are executed at different speeds.

**Positive and negative pairs.** For each target chunk $A_i^{\text{tgt}}$, we compute $\delta$ to every chunk of the demonstrations in $D_{\text{ret}}$. The $m_+$ closest chunks are the positives $\mathcal{P}(i)$; they can come from any retrieved demonstration. The negatives $\mathcal{N}(i)$ are sampled from the retrieved chunks whose distance to $A_i^{\text{tgt}}$ is larger than the median of these distances. Each chunk is represented by the observation at its first step. Because pairs are drawn only from $D_{\text{ret}}$, a target frame is not paired with a frame of an unrelated task that happens to have similar actions, such as a pushing or placing segment for a lid-opening target task.

Figure 8 shows additional examples of retrieval with and without the instruction filter.

## C Implementation Details

**Anchored representation.** Our target tasks differ from the pre-training data mainly in visual appearance, so we anchor the representation computed from the visual input. For the Diffusion Policy, this is the pooled output of the ResNet-18 encoders of all cameras, concatenated into a 1024-dimensional feature. For GR00T, it is the output tokens of the vision tower. Stage 1 trains the layers that produce this representation, together with the projector (Appx. E), for 3 000 steps, and keeps the rest of the policy fixed.

**Anchor distance.** The distance $d$ in the anchor loss (Eq. 6) is the mean squared error for the Diffusion Policy. For GR00T, we divide the squared error by the batch variance of the reference representation $\bar{h}^{\ell}$, so that the same weight can be used for representations of different scales.

**Retrieval and training budget.** In all settings, we retrieve $K = 100$ demonstrations and use action chunks of $H = 16$ steps (Appx. B). All methods in a setting use the same optimizer settings and training budget. Appx. D.2 lists the full settings of the Diffusion Policy on RoboTwin.

## D Evaluation Environments

This appendix describes the RoboCasa and real-robot settings (§D.1), the RoboTwin setting (§D.2), and the representation metrics used in RQ1 (§D.3).

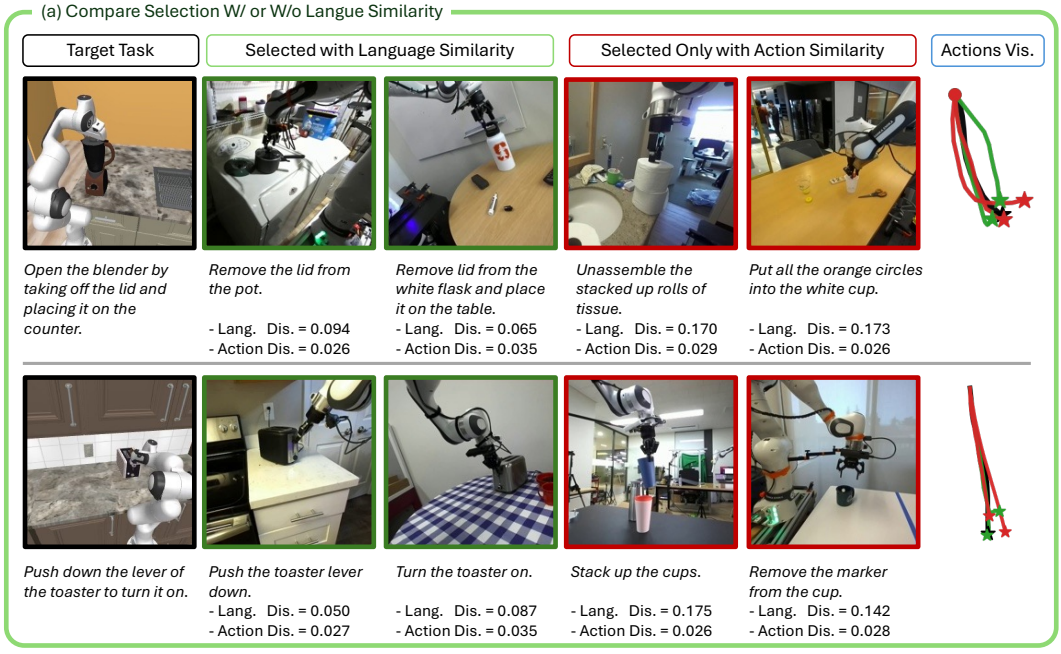



Figure 8: Additional examples of retrieval for RARA. Without the instruction filter, selection by action similarity can return pre-training demonstrations with similar action chunks but a different task. Filtering by the task instruction returns demonstrations that better match the target task.

### D.1 RoboCasa and Real Robot

Figures 9 and 10 show the training and OOD evaluation environments of all RoboCasa and real-robot tasks.

**Evaluation protocol.** In RoboCasa, each policy is evaluated for 50 episodes per task, and each experiment is repeated with three training seeds. For GR00T-based policies, the model predicts an action chunk of 16 steps and executes all 16 steps before replanning. In the real-world Franka experiments, each trained policy is evaluated for 10 trials per task, and this evaluation is repeated three times. To improve fairness across real-world comparisons, we record the randomization setting for each trial, including scene layout, object placement, distractors, and other visual variations. We then evaluate the next model under the same recorded setting whenever possible before switching to a new randomized setting. For real-world GR00T-based policies, the model predicts a 16-step action chunk but executes only the first 8 steps before replanning.

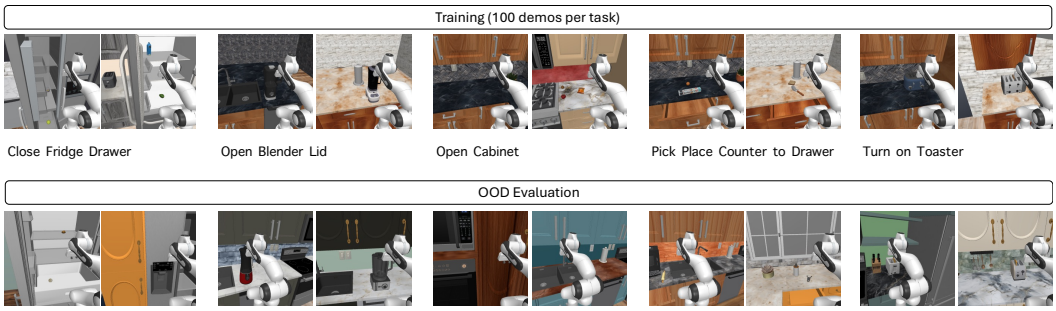



Figure 9: RoboCasa training and OOD evaluation environments for all simulated robot tasks. OOD settings include variations in camera view, background, distractors, objects, clutter, and layouts.

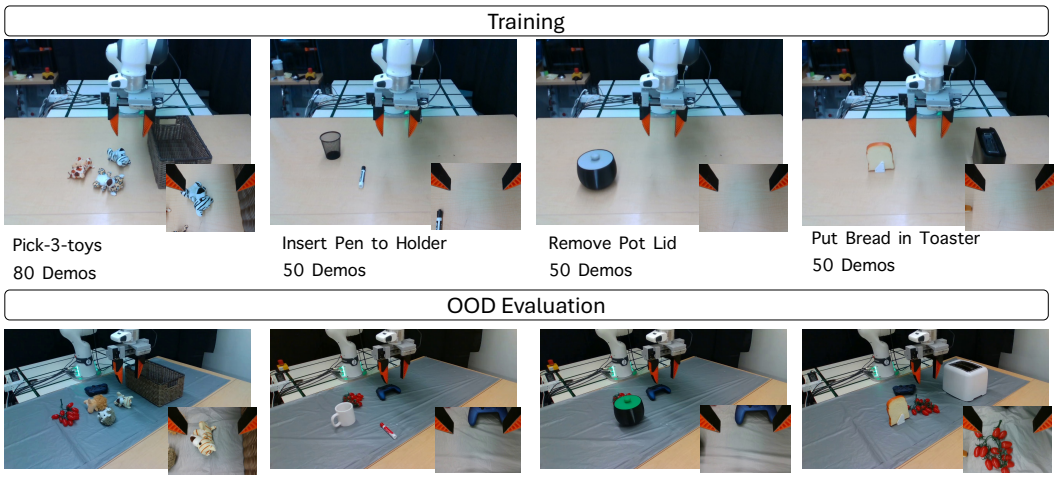



Figure 10: Real-world Franka training and OOD evaluation environments for all robot tasks. OOD settings include variations in camera view, background, distractors, and objects.

### D.2 RoboTwin

RoboTwin 2.0 (Chen et al., 2026) makes it easy to collect demonstrations, and its randomization of object instances, backgrounds, clutter and object positions gives enough diversity to build both a broad pre-training set and narrower target tasks in the same environment. We use the right arm of the Aloha-AgileX embodiment with the head and wrist RGB cameras.

**Pre-training and target data.** We collect about 2,000 pre-training demonstrations in total, in two groups, each paired with two target tasks.

The first group contains pick-and-place demonstrations (`place_a2b_right`), in which the right arm picks up an object and places it next to a second object. The first object is drawn from 11 categories (playing cards, mouse, bell, stapler, soap, toy car, coffee box, phone, Rubik's cube, bread and tea box), and the second from six (Rubik's cube, wooden block, phone, bread, bell and soap). This group serves Place Container and Move Pill Bottle. These targets use the same skill, but their objects do not appear in the pre-training data: Move Pill Bottle uses a pill bottle and a blue pad, and Place Container uses a plate together with a bowl or a cup. The novelty of these two targets is therefore at the object level.

The second group contains four skills: pressing a bell (`click_bell`), shaking a bottle vertically (`shake_bottle`), opening a laptop (`open_laptop`) and rotating a QR code (`rotate_qrcode`). This group serves Click Alarm Clock and Shake Bottle. Click Alarm Clock uses the same pressing skill as `click_bell` on a new object, and Shake Bottle shakes the same kind of object horizontally instead of vertically; neither target task appears in the pre-training data. Opening a laptop and rotating a QR code are unrelated to both targets, and retrieval is meant to exclude them.

The pre-training demonstrations are collected with wide randomization: 100 table textures and backgrounds, cluttered distractors, and object positions over the whole reachable table. Each target task has 50 demonstrations, collected with a narrow layout, 10 backgrounds and light clutter. ID evaluation uses the narrow layout of the demonstrations; OOD evaluation widens the object layout, the backgrounds and the distractors. Figure 11 shows the training and OOD evaluation environments of the four target tasks.

**Policy and training.** We use the Diffusion Policy (Chi et al., 2023) implementation in XPolicyLab (XPolicyLab Community et al., 2026): a ResNet-18 encoder per camera with group normalization and spatial soft-argmax pooling (1024-dimensional visual feature over both cameras), a 1-D UNet denoiser, and 16-step action chunks. It is pre-trained for 120 epochs on the pre-training set. Every adaptation method then trains for 300 epochs on the 50 target demonstrations with a batch of 128. Co-FT fills each batch with 64 target chunks and 64 chunks of the retrieved pre-training demonstrations $D_{\text{ret}}$, and applies the behavior-cloning loss to both. In Stage 2 of RARA, each batch

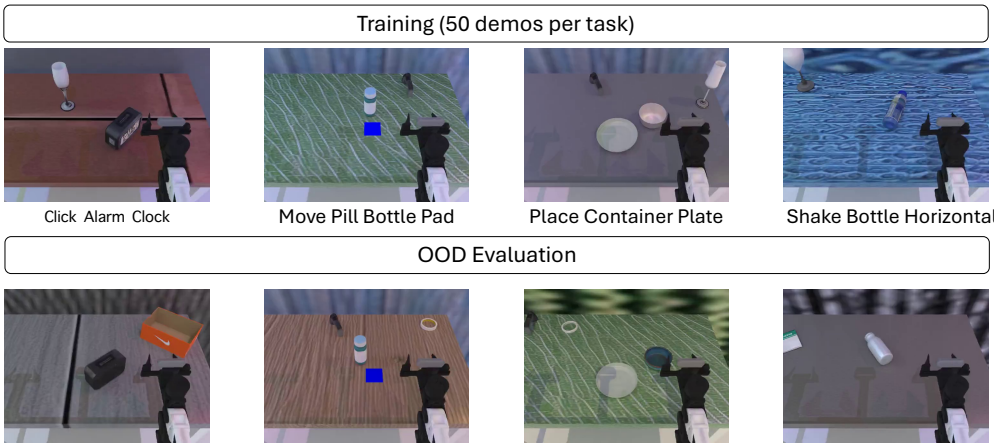



Figure 11: RoboTwin training and OOD evaluation environments for the four target tasks. OOD settings widen the object layout and vary the background and distractors.

holds 64 target chunks and 64 pre-training frames; the behavior-cloning loss is applied to the target chunks and $\mathcal{L}_{\text{anchor}}$ (mean squared error, $\lambda_2 = 1$) to the pre-training frames, without their actions. The *frozen* reference in RQ1 is FT with the layers that produce the anchored representation fixed. Table 4 lists the remaining settings.

**Anchored layer.** By default, the anchored representation $h^\ell$ is the pooled output of the two ResNet-18 encoders, concatenated into a 1024-dimensional feature. Stage 1 updates these encoders and the projector. For the layer study in RQ3, we move the anchored layer to seven other locations while keeping everything else fixed: the outputs of the four ResNet stages (spatially pooled), the FiLM conditioning vector of the UNet, the output of the UNet middle block, and the predicted action. When the anchored layer is inside the action network, the diffusion timestep and the noisy action input are shared between the trainable and the reference policy, so the two differ only in their weights.

**Stage 1.** Retrieval and pair construction follow §4.1 and Appx. B: the top $K = 100$ pre-training demonstrations are retrieved by instruction similarity, and the positives and negatives are defined by the DTW distance between action chunks, with $m_+ = 4$. Stage 1 converges on both targets: the contrastive loss falls from $2.73$ to $0.22$ on Place Container ($92\%$) and from $2.62$ to $0.87$ on Move Pill Bottle ($67\%$).

**Baselines.** All baselines start from the same pre-trained policy and use the same 50 target demonstrations and 300-epoch budget as FT. For weight-space preservation (RETAIN; Yadav et al., 2026), we interpolate the pre-trained and fine-tuned weights as $\theta(\alpha) = (1-\alpha)\theta_{\text{pre}} + \alpha\theta_{\text{ft}}$ with $\alpha \in \{0.25, 0.5, 0.75\}$. For same-image teacher alignment (BlindVLA; Kachaev et al., 2026), we add a feature-distillation term toward frozen DINOv2 features of each target image, with weight $\lambda \in \{1, 0.1, 0.01\}$. For retrieval and co-training (STRAP; Memmel et al., 2025), we follow the original method: the image observations are encoded with DINOv2, target sub-trajectories are matched to pre-training sub-trajectories by subsequence DTW over these visual features, and we keep the top 50 matches and train with the Co-FT loss and the same 1:1 batch mixture.

**Evaluation.** Rollouts use the benchmark's success check with a 400-step horizon. Each result uses the last three checkpoints (epochs 250, 275 and 300), each evaluated on 50 rollouts, so that checkpoint-to-checkpoint noise is averaged rather than selected. Each experiment is repeated with three training seeds, and we report the mean and standard deviation over the seeds.

### D.3 RQ1 Metrics

All metrics use the pooled 512-dimensional output of the head-camera ResNet-18, on frames the policy was not trained on: 100 held-out target episodes in the ID region, 100 in the OOD region,

Table 4: Diffusion Policy settings on RoboTwin.

| | |
|---|---|
| Anchor representation $h^\ell$ (default) | pooled encoder output of both cameras, concatenated (1024-d) |
| Other locations (RQ3) | ResNet stages 1–4, FiLM conditioning, UNet middle block, action output |
| *Stage 1 (alignment)* | |
| Trainable | encoder (both cameras) and projector |
| Projector $g_\psi$ | MLP $1024 \to 512 \to 256$, GELU, $\ell_2$-normalized output |
| Positives / negatives | $m_+ = 4$ nearest by action-chunk DTW; 64 negatives per step, sampled beyond the median distance |
| Temperature, $\lambda_1$ | $\tau = 0.07$, $\lambda_1 = 1$ |
| Optimizer | AdamW, lr $10^{-5}$ (encoder) / $10^{-4}$ (projector), weight decay 0.01, 200 warm-up steps then cosine, gradient clip 1.0 |
| Steps, batch | 3 000 steps; 32 target frames and 32 pre-training frames for $\mathcal{L}_{\text{anchor}}$ per step |
| *Stage 2 (task training with the anchor)* | |
| Trainable | whole policy (encoder and UNet) |
| Batch | 128; target chunks and pre-training frames 1:1 |
| Distance $d$, $\lambda_2$ | mean squared error, $\lambda_2 = 1$ |
| Optimizer | AdamW, lr $10^{-4}$, cosine schedule, 300 epochs |
| Evaluation | epochs 250, 275, 300; 50 rollouts each per split |

and pre-grasp frames of the 250 pre-training episodes. $f_\theta$ denotes the adapted encoder and $f_P$ the pre-trained one.

*Feature distortion* (Kumar et al., 2022) is $\mathbb{E}_o \, \|f_\theta(o) - f_P(o)\|_2 / \|f_P(o)\|_2$ over the frames of a set; 0 means the features did not move. We divide by the norm of the pre-trained feature so that a change of scale does not dominate.

*Effective rank* (Garrido et al., 2023) is $\exp(-\sum_i p_i \log p_i)$ with $p_i = \sigma_i / \sum_j \sigma_j$, where $\sigma_i$ are the singular values of the centered feature matrix of a set.

*Linear CKA* (Kornblith et al., 2019) is $\|X^\top Y\|_F^2 / (\|X^\top X\|_F \, \|Y^\top Y\|_F)$ for the centered feature matrices $X$ of the adapted and $Y$ of the pre-trained encoder on the same frames; 1 means the two representations agree up to rotation and scale. Because CKA can change without a change in function (Davari et al., 2023), we do not rely on it alone.

*Position-probe* $R^2$ (Davari et al., 2022) is the score of a ridge regression from the features to the object position at the grasp of the same episode (32 PCA components, $\alpha = 10$, five-fold GroupKFold by episode). Frames after the grasp are excluded, because the object then moves with the hand. A drop relative to $P$ indicates forgetting.

## E Projector Analysis

Stage 1 computes the contrastive loss in a projected space. A shared two-layer MLP projector ($1024 \to 512 \to 256$, GELU, $\ell_2$-normalized output) maps the anchored representation of both the target frames and the retrieved frames to the contrastive embedding, following common practice in contrastive learning (Chen et al., 2020; Bardes et al., 2022). The projector is trained together with the policy (learning rate $10^{-4}$, compared with $10^{-5}$ for the policy layers) and is discarded after Stage 1, so Stage 2 starts from the updated policy without it. We compare this default with two alternatives on RoboTwin, using the policy pre-trained with one seed: no projector, where the loss is applied directly to the normalized representation, and a randomly initialized frozen projector, which Kachaev et al. (2026) use when aligning with a frozen teacher. Each success rate is computed over 150 rollouts.

Removing the projector does not change the result of Stage 1 followed by fine-tuning without the anchor: $86.7\%$/$60.7\%$ vs. $87.3\%$/$60.0\%$ (ID/OOD) on Place Container, and $30.7\%$ vs. $32.0\%$ (ID) on Move Pill Bottle. A frozen projector changes Stage 1 itself. The representation aligns more strongly: on held-out Move Pill Bottle frames, the AUC of ranking a positive frame above the negatives improves by $8.1$ points over the pre-trained policy, compared with $5.1$ points with the trainable projector. The representation also moves further from that of the pre-trained policy (CKA on pre-training frames $0.86$ vs. $0.95$), while the contrastive loss stays two to three times higher. A trainable projector therefore absorbs part of the alignment that would otherwise change the representation. These differences do not carry over to success. With either projector, Stage 1 followed by fine-tuning reaches $32.0\%$ ID on Move Pill Bottle and $85.3\%$ vs. $87.3\%$ on Place Container, and full RARA reaches $49.3\%$ vs. $53.3\%$ and $82.0\%$ vs. $84.7\%$ (frozen vs. trainable projector, ID). We therefore use the trainable projector by default.

# F CHOICE OF PRE-TRAINING DATA

This section studies which pre-training data RARA should align to. To test whether the skill shown in the data matters, we swap the pre-training data while keeping everything else fixed: episode count, randomization, cameras and acting arm (Table 5). With pick-and-place data, RARA gains $18.8$ and $4.9$ points on the two pick-and-place targets, but only $0.2$ points on Click Alarm Clock. With Click Bell data, the pattern reverses: RARA gains $10.3$ points on Click Alarm Clock, a pressing task like Click Bell, and loses $6.7$ and $20.0$ points on the pick-and-place targets. The anchor therefore tends to help when the pre-training data shows the same skill as the target, and it can hurt when the skill differs. This is why RARA retrieves the pre-training data instead of using all of it.

Table 5: Gain of RARA over FT (points, RoboTwin) when only the pre-training data is changed. Bold: larger gain for each target.

| | Target task | | |
|---|---|---|---|
| Pre-training data | Move Pill Bottle | Place Container | Click Alarm Clock |
| Pick-and-place | $\mathbf{+18.8}$ | $\mathbf{+4.9}$ | $+0.2$ |
| Click Bell | $-6.7$ | $-20.0$ | $\mathbf{+10.3}$ |