Domain Arithmetic: One-Shot VLA Adaptation under Environmental Shifts

Taewook Kang*, Taeheon Kim*, Donghyun Shin, and Jonghyun Choi†

Seoul National University
{tw.kang, thkim0305, dawnme, jonghyunchoi}@snu.ac.kr

Abstract. Vision-Language-Action (VLA) models often fail to perform the same learned tasks under environmental shifts, such as changes in camera pose and shifts to a different but similar robot (*e.g.*, from Panda to UR5e). Adapting these models to the shifted environment (*i.e.*, target domain) often requires training on multiple demonstrations for each task, which are costly to collect. To reduce the burden of data curation and training, we propose an analogy-based method that adapts VLA models under environmental shifts through weight vector arithmetic with domain-specific information addition, named **D**omain **ARi**T**h**metic (**DART**). Unlike prior approaches, DART requires collecting only *a single demonstration*, enabling efficient adaptation. To accurately isolate domain-specific information for addition, DART performs subspace alignment between singular components in weight vectors to filter out noisy components. In both simulated and real-world experiments, DART outperforms existing VLA adaptation methods in one-shot scenarios across diverse visual and embodiment shifts. Code is available at https://github.com/snumprlab/dart.

Keywords: Vision-Language-Action models · Environmental shifts · One-shot adaptation · Weight arithmetic

1 Introduction

Vision-Language-Action (VLA) models trained on large-scale corpora show strong multi-task capabilities [3–5, 23, 24, 42, 65]. Despite their success within trained environments, *i.e.*, *source domain*, VLA models face challenges when deployed in new environments to perform learned tasks, a common real-world deployment scenario. These environmental shifts involve altered camera poses, distinct sensor calibrations, or embodiment modifications, leading to substantial performance degradation [12, 14, 27, 51, 60, 63, 64]. Thus, post-hoc adaptation remains essential to guarantee reliable execution in the shifted environment, *i.e.*, *target domain*. However, existing VLA adaptation approaches [1, 11, 12, 26, 50, 53] often require extensive expert demonstrations for *every* policy task in the target domain, resulting in severe deployment bottlenecks [34, 47, 58]. Also, fine-tuning on limited data often fails to generalize to unseen tasks [11, 53].

* These authors contributed equally.
† JC is with ECE, IPAI and ASRI in SNU and a corresponding author.

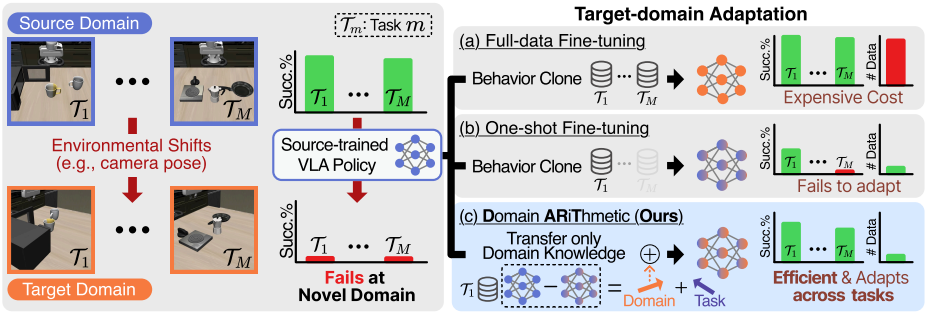


Fig. 1: One-shot VLA adaptation under environmental shifts. Environmental shifts can cause a source-trained VLA policy to fail in the novel target domain. (a) Full-data fine-tuning adapts successfully but requires task-wise demonstrations, incurring expensive data collection costs. (b) One-shot fine-tuning is data-light but often fails to adapt across tasks. (c) Our one-shot adaptation extracts domain-specific directions from fine-tuned weights and adapts the policy via weight arithmetic.

For practical policy deployment, extreme data efficiency for adaptation is essential in settings where collecting task-wise demonstrations at scale is typically infeasible, such as household environments. Thus, we aim for **one-shot VLA adaptation** where a policy adapts under environmental shifts using only a *single* demonstration of a *single* task. To enable this, we leverage the insight that a single demonstration can provide transferable domain knowledge. It allows a source-trained base VLA model to harness its learned task capabilities to solve the same tasks in the target domain, without relearning from scratch (Fig. 1).

To substantiate this idea, we analyze why one-shot fine-tuned models fail to adapt. Through subspace alignment analysis, we find that their parameter changes from a base model, *i.e.*, update-vectors, are predominantly task-specific, with small domain-specific directions present. From this, we further study how these directions coexist. We conjecture that the task and domain directions are additively composable, inspired by the disentangled weights for each task in vision-language models [18, 37, 59], and we find consistent results supporting this structural property in a VLA model.

Building on this observation, we propose **D**omain **AR**i**T**hmetic (**DART**), an analogy-based framework for VLA adaptation inspired by weight arithmetic [18, 46, 62]. Similar to ‘‘`queen = king + woman - man`’’, we add a *domain vector* that captures the environmental shift to the base model, transferring its multi-task capabilities to the target domain without additional data or architectural changes. To isolate the domain vector, we subtract the source-domain update-vector from the target-domain update-vector to cancel out shared task-specific directions, where each update-vector is fine-tuned on one-shot data of the same task.

However, direct subtraction can retain source-domain artifacts and amplify fine-tuning noise [52, 54, 57], corrupting the extracted domain vector. To address this, motivated by the low-rank structure of update-vectors and subspace aligning property for relevant features in model merging [15, 35, 44], we introduce **subspace**

filtering, which filters misaligned subspace basis vectors between source and target updates for subtraction, and **subspace scaling**, which down-weights noisy domain vectors based on source-target subspace alignment.

In simulated and real-world experiments with $\pi_{0.5}$ [4] and π_0-FAST [40], DART outperforms existing VLA adaptation baselines under one-shot scenarios across diverse visual and embodiment shifts. Moreover, our approach enables fast, hyperparameter-robust adaptation and merging across multiple target domains.

Our **contributions** are as follows: (i) Empirical evidence shows that one-shot fine-tuned weights decompose into shareable, additive task- and domain-specific directions. (ii) From this observation, we propose DART that extracts a reusable domain vector from one-shot fine-tuned weights by removing task-specific directions through an analogy operation. (iii) Our approach outperforms prior VLA adaptation methods across diverse simulated and real-world experiments.

2 Related Work

Domain adaptation for Vision-Language-Action models. By integrating pretrained vision-language backbones [2, 32] with large-scale robotic datasets [22, 38, 47], VLA models such as RT-2 [65], OpenVLA [23, 24], and the π series [4, 5, 19, 40] have shown strong performance in a wide range of tasks. However, they often require adaptation in novel environments to avoid performance degradation [12, 27, 50, 51, 63]. A common VLA adaptation approach trains with diverse augmentations to improve generalization [6, 12, 26, 55], but it relies on fine-tuning with additional training data, incurring prohibitive data collection costs. Some approaches reduce this data burden by leveraging semantic-rich visual features [11, 29, 36] or introduce architectural modifications [1, 13, 50]. Yet these choices limit generalizability across backbones and deployment setups. Test-time adaptation methods [9, 33] adapt at inference time without extra VLA training, but are mainly targeting limited shift regimes. To address these limitations, we propose an architecture-agnostic VLA adaptation method with minimal target-domain data under diverse shifts.

Analogy operation using weight arithmetic. Task Arithmetic (TA) [18] manipulates models using weight-space arithmetic operations, primarily through *merging* (*i.e.*, addition) to compose capabilities, and *analogy* (*i.e.*, ‘‘`queen - king = woman - man`’’) to estimate parameter changes that transfer target properties. While merging has advanced through interference mitigation [8, 45, 52, 54] and subspace alignment [15, 35, 39, 44, 49], analogy remains limited to direct subtraction, applied in language models for cross-lingual adaptation [10, 62] and human-alignment transfer [17, 46]. In VLA models, existing work focuses exclusively on merging to improve generalization [11, 53, 56], or compose skills [13, 25, 48]. However, merging cannot selectively transfer specific capabilities such as domain knowledge while preserving others. This limitation motivates revisiting analogy for efficient VLA adaptation. Our empirical analysis reveals disentangled, additive task- and domain-specific directions in one-shot fine-tuned VLA models, making analogy a natural fit for isolating domain vector.

3 Preliminaries

VLA fine-tuning. Let $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t, \mathcal{T})$ denote a Vision-Language-Action (VLA) policy parameterized by θ, which maps an observation $\mathbf{o}_t$ (*e.g.*, third-person and wrist camera images) and a task instruction $\mathcal{T}$ (*e.g.*, language prompts) to a distribution over actions $\mathbf{a}_t$ at time step t. Let $\mathcal{E}_{\text{src}}$ and $\mathcal{E}_{\text{tgt}}$ represent the source and target domains, respectively, where $\mathcal{E}_{\text{tgt}}$ introduces environmental shifts (*e.g.*, camera viewpoint or embodiment changes) in a single (or small number of) environment of $\mathcal{E}_{\text{src}}$. We assume access to base policy θ_0 that has been trained to solve a suite of policy tasks $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_1, \ldots, \mathcal{T}_M\}$ within $\mathcal{E}_{\text{src}}$. For the target domain $\mathcal{E}_{\text{tgt}}$, we are given a dataset $\mathcal{D}_{m,\text{tgt}} = \{(\mathbf{o}_t^{\text{tgt}}, \mathbf{a}_t, \mathcal{T}_m)\}$ comprising a *single* demonstration for one *adaptation* task $\mathcal{T}_m \in \boldsymbol{\mathcal{T}}$ collected per environment in $\mathcal{E}_{\text{tgt}}$.

Our objective is to produce adapted parameters θ^* such that π_{θ^*} performs well in $\mathcal{E}_{\text{tgt}}$ across *all tasks* in $\boldsymbol{\mathcal{T}}$, despite observing only a one-shot, single-task supervision $\mathcal{D}_{m,\text{tgt}}$ per environment. For adapting VLA models, we consider the adaptation through behavior cloning (BC) fine-tuning [5, 24, 65]. Initializing with θ_0, we obtain the target-domain fine-tuned parameters $\theta_{m,\text{tgt}}$ by minimizing a BC objective over the actions in $\mathcal{D}_{m,\text{tgt}}$. We then evaluate $\theta_{m,\text{tgt}}$ within $\mathcal{E}_{\text{tgt}}$ across all tasks in $\boldsymbol{\mathcal{T}}$, including $\mathcal{T}_m$ and remaining *held-out* tasks $\mathcal{T}_{k\neq m} \in \boldsymbol{\mathcal{T}}$.

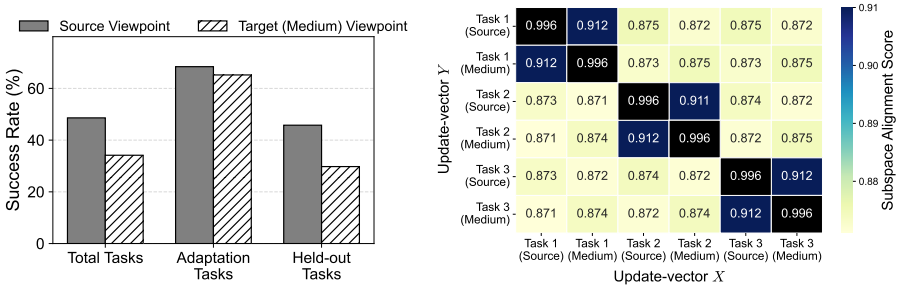
Update-vector. To understand the property of fine-tuned weights, we analyze how the model changes through optimization. Building upon Task Arithmetic [18], we represent an adaptation as a parameter change. Let $\theta_0^{(l)}$ and $\theta_{m,\text{tgt}}^{(l)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denote the weights of layer l in the base and fine-tuned models, respectively. We define the target-domain *update-vector* in layer l, $\Delta_{m,\text{tgt}}^{(l)}$, as:

$$\Delta_{m,\text{tgt}}^{(l)} = \theta_{m,\text{tgt}}^{(l)} - \theta_0^{(l)}, \tag{1}$$

and denote the full update-vector by $\Delta_{m,\text{tgt}} = \{\Delta_{m,\text{tgt}}^{(l)}\}_{l=1}^{L}$.

4 Analysis of One-shot Fine-tuning Failures

Given the substantial cost of collecting data across tasks in each new environment, we consider **one-shot adaptation** using a single demonstration $\mathcal{D}_{m,\text{tgt}}$ for one adaptation task $\mathcal{T}_m$ in the target domain $\mathcal{E}_{\text{tgt}}$. We conjecture in § 1 that $\mathcal{D}_{m,\text{tgt}}$ can provide transferable domain knowledge, enabling the base model θ_0 to harness its multi-task capabilities in $\mathcal{E}_{\text{tgt}}$. A natural approach is to fine-tune θ_0 on $\mathcal{D}_{m,\text{tgt}}$, yielding $\theta_{m,\text{tgt}}$. However, on the LIBERO [31] benchmark with $\pi_{0.5}$ [4], Fig. 2a shows that one-shot fine-tuning generally fails on *held-out* tasks $\mathcal{T}_{k\neq m}$ when evaluated in $\mathcal{E}_{\text{tgt}}$, where tgt = `Medium` (viewpoint shift relative to the source domain, see § 6 for details). At the same time, performance of $\theta_{m,\text{tgt}}$ on $\mathcal{T}_m$ remains higher than that on $\mathcal{T}_{k\neq m}$ when evaluated in the source domain (*i.e.*, `Source` viewpoint). This indicates that one-shot model updates primarily capture task-specific behavior rather than adapting to the target domain across tasks.



(a) One-shot fine-tuning performance. **(b) Alignment $\gamma(X, Y)$ between update-vectors.**

Fig. 2: Properties of one-shot fine-tuning. (a) The model is fine-tuned on adaptation tasks in target (`Medium`) camera viewpoint. Performance remains high in adaptation tasks but generalizes poorly to other held-out tasks. (b) Subspace alignment $\gamma(\cdot, \cdot)$ among update-vectors $\Delta_{m,\text{tgt}} = \theta_{m,\text{tgt}} - \theta_0$ on $m \in \{1, 2, 3\}$ and tgt $\in \{\texttt{Source}, \texttt{Medium}\}$. Vectors align for the same task and domain, showing task- and domain-shared directions.

4.1 Subspace Alignment Between Update-Vectors

To understand why one-shot fine-tuned weights fail in multi-task settings, we inspect the components of parameter updates. In particular, we analyze the similarity between layer-wise update-vectors $\Delta^{(l)}_{m,\text{tgt}}$ across tasks and domains. Since models encoding the same knowledge or capability exhibit high similarity [20, 52], we expect to see the same pattern for update-vectors trained on the same task or domain. As subspace similarity is a strong predictor of transferability and downstream performance [7, 15, 35, 44], we quantify the similarity between two update-vectors $\Delta^{(l)}_i$ and $\Delta^{(l)}_j$ at layer l using the subspace alignment score [35]:

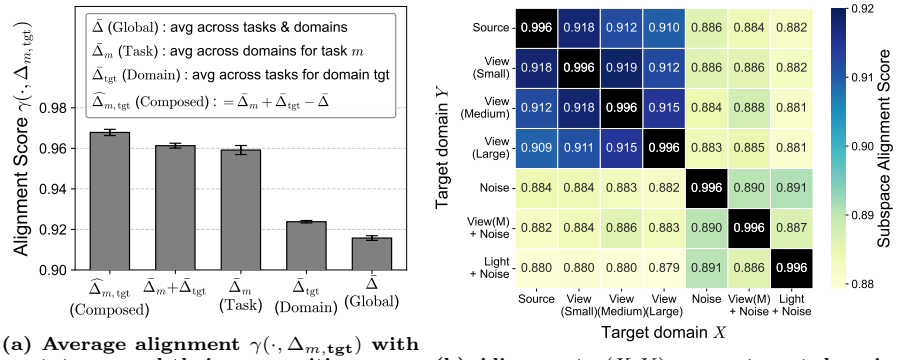
$$\gamma^{(l)}(\Delta_i, \Delta_j) = \frac{\left\| U^{(l)}_j U^{(l)\top}_j \Delta^{(l)}_i \right\|_F}{\left\| \Delta^{(l)}_i \right\|_F}, \tag{2}$$

where $U^{(l)}_j$ are left singular vectors from the Singular Value Decomposition (SVD): $\Delta^{(l)}_j = U^{(l)}_j \Sigma^{(l)}_j {V^{(l)}_j}^{\top}$.[1] Conceptually, $\gamma^{(l)}(\Delta_i, \Delta_j)$ measures the fraction of $\Delta^{(l)}_i$ that can be represented by the subspace of $\Delta^{(l)}_j$. In practice, we aggregate $\gamma^{(l)}(\cdot, \cdot)$ across layers into $\gamma(\cdot, \cdot) = \frac{1}{L} \sum_{l=1}^{L} \gamma^{(l)}(\cdot, \cdot)$ to obtain a single alignment score.[2]

As shown in Fig. 2b, update-vectors from the same adaptation task exhibit strong alignment across domains, indicating that one-shot updates are dominated by task-specific directions. At the same time, we observe slightly higher overlap among vectors targeting the same domain across different tasks than those targeting different domains, suggesting a consistent domain-shared component in

[1] We use top-r vectors of $U^{(l)}_j$ with $r = \min \left\{ r' : \frac{\sum_{i=r'+1}^{R} \sigma_i^2}{\sum_{i=1}^{R} \sigma_i^2} \leq 0.05^2 \right\}$, following [35].

[2] We exclude layers with one-dimensional (*e.g.*, biases and normalization) weights.



(a) Average alignment $\gamma(\cdot, \Delta_{m,\text{tgt}})$ with prototypes and their composition.

(b) Alignment $\gamma(X, Y)$ across target domains.

Fig. 3: Additive task-domain directions in update-vectors. (a) Prototypes are computed by 16 update-vectors from 4 tasks and 4 domains. Strong alignment by additive composition suggests orthogonal, linearly combinable task and domain components. (b) `View` is viewpoint shift, `Noise` is camera noise, `Light` is light change. Alignment among similar domain shifts shows that domain components are structured and correlated with semantic of domain shifts.

the updates. This indicates the presence of domain knowledge within the weight space that can be learned through one-shot fine-tuning.

4.2 Additive Composition of One-shot Update-Vector

Building on observation of task and domain-shared directions in update-vectors in § 4.1, we investigate how these directions coexist. Prior work shows that fine-tuning vision and language models for different tasks often updates orthogonal [59], disentangled weight subspaces [18, 21, 37], enabling interference-free additive composition of task capabilities. Because VLA models are built upon these pretrained vision-language backbones [4, 24], we expect them to inherit these structural properties. We further conjecture that this disentanglement extends beyond task capabilities to domain knowledge. Specifically, we hypothesize that within a one-shot target-domain update-vector $\Delta_{m,\text{tgt}}$, the directions capturing task-specific behavior and those capturing domain-specific adaptation can be independently identified and additively composed without mutual interference.

Validation of additive composition via prototypes. We empirically validate the additive composition hypothesis. Suppose we obtain a set of one-shot update-vectors $\Delta_{m,\text{tgt}} = \theta_{m,\text{tgt}} - \theta_0$ from a set of tasks $\boldsymbol{\mathcal{T}}$ and a set of domains $\boldsymbol{\mathcal{E}}$. From this set of update-vectors, we define three prototypes: (i) task prototype $\bar{\Delta}_m := \frac{1}{|\boldsymbol{\mathcal{E}}|}\sum_{\text{tgt}} \Delta_{m,\text{tgt}}$ averaged across domains, (ii) domain prototype $\bar{\Delta}_{\text{tgt}} := \frac{1}{|\boldsymbol{\mathcal{T}}|}\sum_{m} \Delta_{m,\text{tgt}}$ averaged across tasks, and (iii) global prototype $\bar{\Delta} := \frac{1}{|\boldsymbol{\mathcal{T}}||\boldsymbol{\mathcal{E}}|}\sum_{m,\text{tgt}} \Delta_{m,\text{tgt}}$. We expect that an update-vector for any specific task-domain pair (m, tgt) should be correctly estimated via the additive composition as $\widehat{\Delta}_{m,\text{tgt}} := \bar{\Delta}_m + \bar{\Delta}_{\text{tgt}} - \bar{\Delta}$. Intuitively, $\bar{\Delta}_m$ and $\bar{\Delta}_{\text{tgt}}$ capture domain-invariant

task directions and task-agnostic domain directions, respectively, and subtracting $\bar{\Delta}$ removes common shifts inherent in fine-tuning.

We evaluate how well each prototype and composition explains $\Delta_{m,\text{tgt}}$ using the subspace alignment metric $\gamma(\cdot, \Delta_{m,\text{tgt}})$ in Eq. (2). In Fig. 3a, the alignment scores of each prototype across 16 update-vectors exhibit low standard deviation, implying consistent task and domain directions in prototypes and thus in each update-vector. Moreover, the composed estimate $\widehat{\Delta}_{m,\text{tgt}}$ yields the highest alignment with $\Delta_{m,\text{tgt}}$, suggesting that domain and task adaptations map to linearly combinable directions in weight space, validating our hypothesis. We further discuss why the task and domain components are linearly decomposable in the supplementary material.

Alignment between update-vectors on different domains. To further understand the properties of the domain directions in update-vectors, we keep the adaptation task fixed and evaluate the alignment of update-vectors across diverse target domains (*e.g.*, viewpoint shifts of varying magnitude, camera noise, and their compositions) from LIBERO-Plus [12] benchmark. The subspace alignment results in Fig. 3b reveal that the domain-specific updates learned from fine-tuning are not arbitrary, but structured. Similar environmental changes (*e.g.*, viewpoint shifts) produce similar update directions, while combining two shifts (*e.g.*, viewpoint + noise) produces an update that partially reuses the directions learned for each shift alone. This suggests the model organizes domain knowledge in a compositional way, where each type of environment change corresponds to a distinct, reusable direction in weight space.

5 DART: Domain ARiThmetic

In § 4, we find that a one-shot fine-tuned model fails to adapt since its update-vector is dominated by task-relevant directions, but we also find that the update-vector can be linearly decomposable into common task and domain components. Motivated by this, we aim to decompose and utilize the domain component for model adaptation. Therefore, we propose **D**omain **AR**i**T**hmetic (**DART**), an *analogy*-based method inspired by weight arithmetic [18,62]. Instead of computing the domain prototype as in § 4.2, which requires multiple target-domain tasks, we extract the domain direction from a single target update-vector by removing task directions using a source-domain demonstration. To further enhance this extraction, we introduce subspace filtering and scaling that remove domain-irrelevant noise from update-vectors. Fig. 4 provides an overview of our approach.

5.1 Domain Vector Extraction

Building on the additive properties in § 4.2, we extract the domain-specific directions by subtracting the task-specific directions from the target update-vector $\Delta_{m,\text{tgt}}$. To estimate these task directions without additional target-domain data, we leverage a *source-domain* demonstration $\mathcal{D}_{m,\text{src}}$ of the *same* task $\mathcal{T}_m$, which is typically available from the data used to train the base model θ_0. In practice,

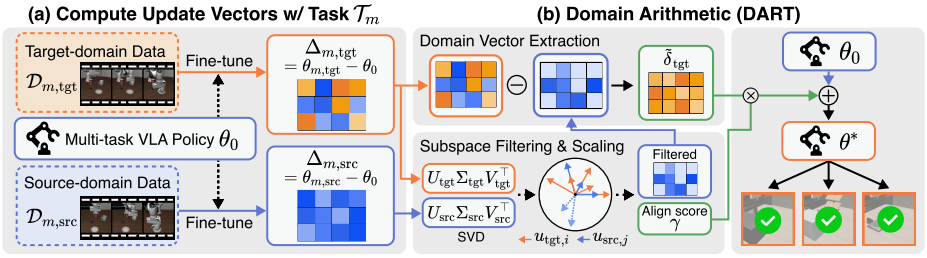


Fig. 4: Overview of the proposed VLA adaptation approach. (a) We compute update-vectors $\Delta_{m,src}$ and $\Delta_{m,tgt}$ by fine-tuning a base policy θ_0 on a single task $\mathcal{T}_m$ using source and target data. (b) A domain vector $\tilde{\delta}_{tgt}$ is extracted by subtracting task directions, with subspace filtering to suppress misaligned components. Adding $\tilde{\delta}_{tgt}$ back to θ_0 yields a multi-task policy θ^* adapted to the target domain.

one can select $\mathcal{T}_m$ from the source dataset and then collect the corresponding target-domain expert demonstration to guarantee the same task. Please refers to the supplementary material for detailed discussion of source-domain data.

Let $\theta^{(l)}_{m,\text{src}}$ denote the parameters for layer l fine-tuned on $\mathcal{D}_{m,\text{src}}$. Since both source update-vector $\Delta^{(l)}_{m,\text{src}} = \theta^{(l)}_{m,\text{src}} - \theta^{(l)}_0$ and target update-vector $\Delta^{(l)}_{m,\text{tgt}}$ in Eq. (1) are learned from the same adaptation task, they primarily share common task components. Thus, we define the **domain vector** $\delta^{(l)}_{\text{tgt}}$ for layer l as:

$$\delta^{(l)}_{\text{tgt}} = \Delta^{(l)}_{m,\text{tgt}} - \Delta^{(l)}_{m,\text{src}}. \tag{3}$$

This subtraction neutralizes the task-specific directions, leaving only domain-specific directions that encode the environmental shift to the target domain.

5.2 Subspace Alignment for Enhanced Domain Vector

The domain vector extraction through direct subtraction between target and source update-vectors may successfully isolate target-domain knowledge. However, fine-tuning often includes task-irrelevant noise [52, 54, 57], and even minor source-domain artifacts can be encoded in $\Delta_{m,\text{src}}$. Because weight updates reside in low-rank subspaces that can be rotated or misaligned by fine-tuning artifacts [35,43,61], naive subtraction may inadvertently inject source-domain noise into the target domain vector or fail to remove task semantics. To remove irrelevant noise in domain vectors, we leverage the spectral properties in one-shot update-vectors.

Subspace filtering. Our intuition is that the shared task semantics lie in the mutually shared subspace between $\Delta_{m,\text{src}}$ and $\Delta_{m,\text{tgt}}$. By filtering the basis vectors of $\Delta_{m,\text{src}}$ that weakly align with $\Delta_{m,\text{tgt}}$, we can prevent source-specific noise from corrupting the domain vector. We only filter the source update-vector as unique bases of target update-vector likely encode the domain directions we seek to isolate. Specifically, we decompose each update-vector for layer l via SVD: let $\Delta^{(l)}_{m,\text{tgt}} = U^{(l)}_{\text{tgt}}\Sigma^{(l)}_{\text{tgt}}V^{(l)\top}_{\text{tgt}}$ and $\Delta^{(l)}_{m,\text{src}} = U^{(l)}_{\text{src}}\Sigma^{(l)}_{\text{src}}V^{(l)\top}_{\text{src}}$, omitting m for brevity.

We first identify which source basis vectors are geometrically aligned with the target subspace. Following subspace alignment in model merging [28, 35, 41], we focus on aligning the column space of the left singular vectors U, as it captures how the update perturbs a layer's *output-feature directions* across all input directions, highly related to output changes. We form the interaction matrix $C^{(l)} := U_{\text{tgt}}^{(l)\top} U_{\text{src}}^{(l)}$ and define the overlap energy $e_j^{(l)}$ of each source basis vector $\mathbf{u}_{\text{src},j}$ as

$$e_j^{(l)} := \left\| C_{:,j}^{(l)} \right\|_2^2 = \left\| U_{\text{tgt}}^{(l)\top} \mathbf{u}_{\text{src},j}^{(l)} \right\|_2^2, \tag{4}$$

where $j \in \{1, \ldots, R\}$ indexes the column vectors of U_{src}. A high energy $e_j^{(l)}$ indicates that the j-th source basis vector lies largely within the target subspace, signifying a shared task feature. To retain only these shared features in $\Delta_{m,\text{src}}^{(l)}$, we determine a dynamic cutoff based on the subspace alignment score $\gamma^{(l)}(\Delta_{m,\text{src}}, \Delta_{m,\text{tgt}})$ in Eq. (2). Since this score quantifies the fractional overlap between the two subspaces, it serves as a natural criterion for determining how many basis vectors to retain. Let $e_{(1)}^{(l)} \geq \cdots \geq e_{(R)}^{(l)}$ denote energies sorted in descending order. We select basis vectors that are over the energy threshold by

$$r_l := \min \left\{ r : \sum_{i=1}^{r} e_{(i)}^{(l)} \geq \gamma^{(l)} \sum_{j=1}^{R} e_j^{(l)} \right\}, \qquad \mathcal{J}_l := \{\, j : \; e_j^{(l)} \geq e_{(r_l)}^{(l)} \,\}, \tag{5}$$

leading to the *aligned* source basis matrix $\tilde{U}_{\text{src}}^{(l)} := U_{\text{src}}^{(l)}[:, \mathcal{J}_l]$. We then obtain the filtered source update-vector $\tilde{\Delta}_{m,\text{src}}^{(l)} = \tilde{U}_{\text{src}}^{(l)} \tilde{U}_{\text{src}}^{(l)\top} \Delta_{m,\text{src}}^{(l)}$.

This subspace filtering ensures that we only subtract components that are aligned with each other. Unlike subspace aligning methods in model merging [15, 28,35,54], which remove insignificant subspaces in all update-vectors and maximize each vector's unique components to maintain capabilities from each update vector, we find the common singular components between update-vectors to correctly remove common knowledge.

Subspace scaling. Although subspace filtering can remove some misaligned singular vectors, if the two update-vectors are fundamentally misaligned, *i.e.*, $\gamma^{(l)} \to 0$, filtering alone cannot fully ensure the correct domain vector. Thus, we scale the domain vector by the alignment score $\gamma^{(l)}$ to down-weight if the domain vector is noise-dominant or irrelevant due to misaligned updates. Specifically, we obtain refined domain vector $\tilde{\delta}_{\text{tgt}}^{(l)}$ by scaling the domain vector using $\gamma^{(l)}$ as

$$\tilde{\delta}_{\text{tgt}}^{(l)} = \gamma^{(l)} \cdot \left(\Delta_{m,\text{tgt}}^{(l)} - \tilde{\Delta}_{m,\text{src}}^{(l)} \right). \tag{6}$$

Finally, we adapt the base policy θ_0 to the target domain by injecting the domain vector into θ_0:

$$\theta^* = \theta_0 + \alpha \cdot \tilde{\delta}_{\text{tgt}}, \tag{7}$$

where $\tilde{\delta}_{\text{tgt}} = \{\tilde{\delta}_{\text{tgt}}^{(l)}\}_{l=1}^{L}$ and α is a scalar coefficient controlling the adaptation strength. This approach efficiently transfers the base policy to the target domain $\mathcal{E}_{\text{tgt}}$ while preserving the multi-task capabilities inherent in θ_0.

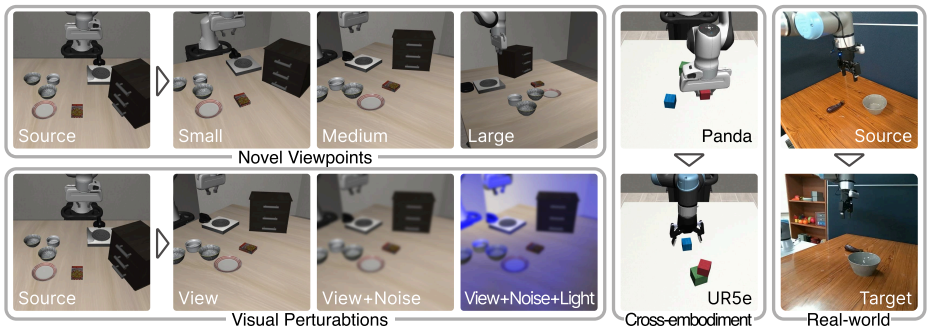


Fig. 5: Overview of experimental setups. We experiment on four setups: simulation setups with novel viewpoints (top-left) and combined visual perturbations (bottom-left) on LIBERO [31], a cross-embodiment transfer setup on MimicGen [34] (middle), and a real-world setup on two third-person camera viewpoints (right).

6 Experiments

To evaluate our method, we conduct (i) simulation experiments under diverse visual shifts and cross-embodiment transfer (§ 6.2), (ii) real-world experiments under viewpoint shifts (§ 6.3), and (iii) additional analyses (§ 6.4).

6.1 Setups

Models. We primarily evaluate our approach on $\pi_{0.5}$ [4], a flow-matching-based VLA model [30]. To assess architectural generality, we additionally evaluate on π_0-FAST [40], which uses autoregressive action-token generation.

Baselines. In both simulation and real-world experiments, we compare DART with architecture-agnostic VLA adaptation baselines: (i) **Zero-shot** (no adaptation), (ii) **One-shot FT** (full fine-tuning on the one-shot dataset), (iii) **FLA** [26] (vision encoder adaptation using LoRA [16]), and (iv) **RETAIN** [53] (model merging between source model and One-shot FT with module-wise scaling).

Simulation setup. For visual shifts, we evaluate on LIBERO [31], a robot manipulation benchmark with four task suites of total 40 tasks. Following prior work [12, 26, 50], we apply third-person viewpoint shifts and visual perturbations to LIBERO to mimic real-world environmental shifts (Fig. 5 left). We consider three levels of viewpoint shift relative to the source camera pose: `Small`, `Medium`, and `Large`, and two visual perturbations applied on top of the viewpoint shifts: `Noise` (noise injection) and `Light` (illumination change). We adapt the base model trained on the original LIBERO training dataset [4, 40] to each target domain using a scene-wise one-shot dataset collected in the target domain, containing a single demonstration from one task in each of the five LIBERO scenes. We repeat one-shot adaptation three times with different randomly selected adaptation tasks and report the average *Success Rate* (%), with 50 rollouts per task.

Table 1: Performance on LIBERO across novel viewpoints using $\pi_{0.5}$. We report average success rates of total 40 tasks for three trials of one-shot adaptation with different adaptation tasks, with the best in **bold**.

	Novel Viewpoints (Success Rate, %)			
Methods ($\pi_{0.5}$)	`Small`	`Medium`	`Large`	Average
Zero-shot	88.3	63.9	11.3	54.5
One-shot FT	43.4	33.3	17.8	31.5
RETAIN (ICLR 2026)	87.4	72.4	48.9	69.6
FLA (CVPR 2026)	**92.2**	76.4	54.3	74.3
DART (**Ours**)	92.0	**80.8**	**64.4**	**79.1**

Table 2: Performance on LIBERO under combined visual shifts using $\pi_{0.5}$. We evaluate under the `Medium` viewpoint shift (`View`) and two combined settings: `View+Noise` (camera noise) and `View+Noise+Light` (camera noise with illumination).

	Visual Perturbations (Success Rate, %)			
Methods ($\pi_{0.5}$)	`View`	`View+Noise`	`View+Noise+Light`	Average
Zero-shot	63.9	60.3	57.2	60.5
One-shot FT	33.3	27.7	28.5	29.8
RETAIN (ICLR 2026)	72.4	65.2	68.5	68.7
FLA (CVPR 2026)	76.4	67.8	70.2	71.5
DART (**Ours**)	**80.8**	**69.2**	**75.0**	**75.0**

For cross-embodiment transfer, we evaluate on MimicGen [34] with `Stack` and `Stack Three` tasks, transferring from Panda to UR5e (Fig. 5 middle). Based on MimicGen-pretrained $\pi_{0.5}$, the base policy θ_0 is trained on two tasks with Panda. We then derive the source- and target-domain vector using one `Stack` demonstration from each robot. We report *Progress Rate* (Prog., %) and *Success Rate* (Succ., %) as metrics, averaged over 5 seeds with 50 rollouts per seed.

Real-world setup. We evaluate on five real-world tasks using a 6-DoF UR10e robot arm with a Robotiq 2F-85 gripper (Fig. 5 right): three pick-and-place tasks (`Eggplant`, `Lemon`, `Carrot`) and two fine-grained manipulation tasks (`Stack Cube`, `Press Stapler`). We use 120 demonstrations (24 per task) collected under the `Source` viewpoint to train $\pi_{0.5}$, and collect one additional `Stack Cube` demonstration under the `Target` viewpoint for adaptation. We report *Success Rate* (%), averaged over 12 rollouts per task with distinct object placements.

Implementation details. For DART, we fine-tune source and target one-shot models ($\theta_{m,\text{src}}, \theta_{m,\text{tgt}}$) for 1,000 steps from the base model θ_0. We set the scaling coefficient α to 0.8 for DART via a small search (10 rollouts per task) on the `Medium` viewpoint in a single task suite of LIBERO, and use the same value for all other task suites, viewpoints, architectures, and in the real-world experiments. We use the same procedure to find hyperparameters and evaluate each baseline

Table 3: Performance on LIBERO across novel viewpoints using π_0-FAST. We report average success rates of total 40 tasks for three trials of one-shot adaptation with different adaptation tasks, with the best in **bold**.

	Novel Viewpoints (Success Rate, %)			
Methods (π_0-FAST)	`Small`	`Medium`	`Large`	Average
Zero-shot	84.6	73.6	62.0	73.4
One-shot FT	71.1	63.0	52.2	62.1
RETAIN (ICLR 2026)	88.3	78.4	62.7	76.5
FLA (CVPR 2026)	86.5	78.4	64.9	76.6
DART (**Ours**)	**91.2**	**80.8**	**66.2**	**79.4**

Table 4: Performance on MimicGen under cross-embodiment transfer using $\pi_{0.5}$. We adapt a source policy trained on the Panda robot to the UR5e robot, and report progress rate and success rate.

	`Stack`		`Stack Three`		Average	
Methods ($\pi_{0.5}$)	Prog. (%)	Succ. (%)	Prog. (%)	Succ. (%)	Prog. (%)	Succ. (%)
Zero-shot	89.4	86.8	70.1	37.2	79.8	62.0
One-shot FT	87.8	84.8	60.9	28.0	74.4	56.4
DART (**Ours**)	**94.8**	**93.4**	**73.8**	**45.4**	**84.3**	**69.4**

on the same one-shot dataset as our method. Additional details of experimental setup are provided in the supplementary material.

6.2 Simulation Results

Novel visual domains. As shown in Tab. 1, DART outperforms all baselines under diverse novel viewpoints. Notably, applying the domain vector $\tilde{\delta}_{\text{tgt}}$ to the base policy (θ_0) yields a substantial gain of 24.6 percentage points (pp). These results support our hypothesis that domain vectors provide domain-specific knowledge while preserving multi-task capabilities. Our method outperforms FLA [26], which fine-tunes only the vision encoder, highlighting the importance of adapting the entire model in a data-limited setting. Compared with RETAIN [53], our analogy-based approach performs better, suggesting that explicitly isolating domain-shift directions helps under data scarcity.

Tab. 2 summarizes performance under combined visual perturbations. DART maintains a clear advantage over baselines across all settings. This trend indicates that the advantage of our method persists even under combined environmental shifts, rather than being limited to a single perturbation type.

Applicability to an alternative VLA architecture. To test generalizability beyond $\pi_{0.5}$'s flow-matching formulation, we apply our approach to π_0-FAST, an autoregressive VLA model. As shown in Tab. 3, our method also consistently outperforms all baselines on this architecture. This result supports that the additive structure of update-vectors is applicable to diverse VLA architectures.

Table 5: Performance on real-world UR10e robot using $\pi_{0.5}$. We use a single `Stack Cube` demonstration to adapt models to the target domain (novel viewpoint).

	Novel Viewpoint (Success Rate, %)					
	Pick-and-Place			Fine-grained Manipulation		
Methods ($\pi_{0.5}$)	`Eggplant`	`Lemon`	`Carrot`	`Stack Cube`	`Press Stapler`	Average
Zero-shot	50.0	33.3	41.7	16.7	75.0	43.3
One-shot FT	58.3	58.3	41.7	33.3	66.7	51.7
RETAIN (ICLR 2026)	58.3	41.7	41.7	16.7	83.3	48.3
FLA (CVPR 2026)	58.3	50.0	50.0	16.7	**100.0**	55.0
DART (**Ours**)	**91.7**	**91.7**	**83.3**	**41.7**	**100.0**	**81.7**

Cross-embodiment transfer. Beyond visual domain shifts, upgrading hardware or transferring policies across different robotic platforms remains a major bottleneck in real-world deployment. To test if our approach can bridge this physical domain gap, we examine whether the analogy principle in Eq. (3) extends to cross-embodiment transfer. Tab. 4 shows that DART is also applicable to the cross-embodiment setting, which differs substantially from visual domain adaptation. This highlights that our approach can be applied across diverse visual and physical environmental shifts without any algorithmic modification.

6.3 Real-world Results

We evaluate DART in a real-world setup to assess whether it remains effective under real-world variability. Table 5 summarizes results under third-person camera viewpoint shifts on a UR10e robot. Despite adapting from only a single `Stack Cube` demonstration, our method achieves high success rates across all five tasks, indicating that their domain-level transfer is effective even in diverse real-world conditions. In contrast, baselines perform substantially worse under the same one-shot budget, indicating limited transfer beyond the demonstration. We provide experiment videos in the supplementary material.

6.4 Detailed Analysis

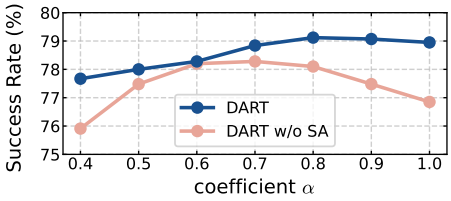
Ablation study. Tab. 6 isolates the effect of each component in DART. DART without any subspace alignment component yields substantial improvement over One-shot FT (in Tab. 1), suggesting that the analogy-based weight arithmetic between source- and target-domain update vectors effectively isolates domain-specific knowledge. This validates the predominant, domain-agnostic task-specific directions among update vectors evidenced in §4.1, and confirms their linearly decomposable property consistent with §4.2. Building on this, subspace filtering leads to marked improvement, highlighting the importance of suppressing noisy source-domain artifacts in the source-domain update vector for more precise domain vector extraction. Furthermore, subspace scaling yields additional gains, suggesting the presence of highly misaligned, low-quality domain vectors whose contribution is better modulated through alignment-aware reweighting.

Table 6: Ablation study of each component in DART. We report average success rates (%) across `Small`, `Medium`, and `Large` viewpoints in LIBERO. `Sub. Filter.` is subspace filtering, and `Sub. Scale.` is subspace scaling in § 5.2.

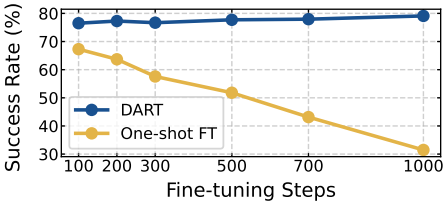
Components		
`Sub. Filter.`	`Sub. Scale.`	Average
✗	✗	78.1
✓	✗	78.8
✗	✓	78.5
✓	✓	**79.1**

Table 7: Merging domain vectors $\tilde{\delta}_{\mathbf{tgt}}$ across novel viewpoints. DART reports average success rates (%) of a separately adapted model in three novel viewpoints in LIBERO. Each single merged model is evaluated across the three domains.

Methods ($\pi_{0.5}$)		Average
DART		**79.1**
DART + Merging	TA [18]	74.5
	TIES [52]	70.9
	TSV [15]	75.7
	Iso-C [35]	74.8



(a) Impact of scaling coefficient α.



(b) Impact of fine-tuning steps.

Fig. 6: Performance under hyperparameter choices on LIBERO across novel viewpoints. We average success rates (%) across `Small`, `Medium`, `Large` camera views.

Merging multiple domain vectors. We study whether domain vectors for different target domains from DART can be consolidated into a single transferable vector, motivated by the additivity of domain directions in weight space (§ 4.2). We merge three novel viewpoint-shift domain vectors $\tilde{\delta}_{\text{tgt}}$ from LIBERO into a combined vector δ^* using model-merging methods [15, 18, 35, 52], and adapt the base model as $\theta^* = \theta_0 + \alpha \cdot \delta^*$. As shown in Tab. 7, the merged vector successfully adapts the base model to all three domains, emphasizing the composable nature of domain vectors. This indicates the practicality of DART that only a single consolidated vector can be maintained across multiple target domains, reducing the memory overhead of storing each domain vector.

Scaling coefficient α. Figure 6a shows how DART and its simplified version, DART without Subspace Alignment (DART w/o SA) that does not apply subspace filtering or scaling, vary in performance with the scaling coefficient α in Eq. (7). Small α is insufficient to address the environmental shift, whereas large α can interfere with the base policy's multi-task capabilities. Nevertheless, DART maintains strong performance across a wide range of α. In particular, DART is more stable than DART w/o SA, suggesting that subspace filtering and scaling suppress noisy components that would otherwise be amplified by α.

Fine-tuning steps. Fig. 6b shows the effect of the number of fine-tuning steps for the update-vectors in Eq. (3). While One-shot FT degrades over time due to catastrophic forgetting [53], DART, which extracts the domain vector from One-shot FT weights, shows small but consistent performance gains, likely as task-specific directions become more pronounced with training, leading to better domain vector extraction. Notably, DART maintains strong performance even at small fine-tuning steps, enabling time-efficient adaptation.

We further show that DART avoids source-domain forgetting, compare DART with existing model-merging and test-time-adaptation methods, and analyze the choice of layers to which we add the domain vector and the choice of tasks used to extract it. Please see the supplementary material for details.

7 Limitation

While DART consistently outperforms baselines, performance degrades under severe shifts (e.g., `Large` viewpoint), a challenge shared by all one-shot methods. We leave this to future work, *e.g.*, via more reliable domain vector extraction or stronger base-model training/fine-tuning schemes. Additionally, the scalar coefficient α requires a small hyperparameter search, though our analysis shows DART remains stable across a wide range of values and generalizes well across environments. Hyperparameter-free per-layer adaptive scaling for practical real-world application is left for future work.

8 Conclusion

We propose a method to adapt VLA models for environmental shifts with only a single demonstration collection. Motivated by our observation that one-shot fine-tuned parameters admit an approximately additive decomposition into task- and domain-specific directions, we introduce DART, an analogy-based approach that adds filtered domain-specific directions isolated by weight arithmetic. Extensive evaluation in simulation and on real-world setups shows consistent improvement and applicability of DART across diverse visual and embodiment shifts.

Acknowledgements

We thank Seongwon Cho, Youhan Lee, and Jimin Nam for their helpful comments. This work was partly supported by the InnoCORE program (26-InnoCORE-01), the IITP grants (RS-2022-II220077, RS-2022-II220113, RS-2022-II220959, RS-2022-II220871, RS-2026-25507282, RS-2026-25518317, RS-2021-II211343 (SNU AI), RS-2025-25442338 (AI Star Fellowship-SNU)), 02-26-01-0285 (Advanced GPU Utilization Support Program by NIPA) funded by the Korea government (MSIT), grants (RS-2025-25462891 (US-KOR BARI), RS-2025-25453780) funded by MOTIR, a grant (RS-2025-25460896) funded by MOTIR and KIAT, a grant of Korean ARPA-H Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (RS-2025-25424639), and the BK21 FOUR program, SNU in 2025.

References

1. Abouzeid, A., Mansour, M., Sun, Z., Song, D.: Geoaware-vla: Implicit geometry aware vision-language-action model. arXiv preprint arXiv:2509.14117 (2025)
2. Beyer, L., Steiner, A., Pinto, A.S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., et al.: Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726 (2024)
3. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)
4. Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M.R., Finn, C., Fusai, N., Galliker, M.Y., et al.: $\pi_{0.5}$: A vision-language-action model with open-world generalization. In: CoRL (2025)
5. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0: A vision-language-action flow model for general robot control. RSS (2025)
6. Chen, L.Y., Xu, C., Dharmarajan, K., Irshad, M.Z., Cheng, R., Keutzer, K., Tomizuka, M., Vuong, Q., Goldberg, K.: Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. In: CoRL (2024)
7. Chen, X., Wang, S., Fu, B., Long, M., Wang, J.: Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. NeurIPS (2019)
8. Cheng, R., Xiong, F., Wei, Y., Zhu, W., Yuan, C.: Whoever started the interference should end it: Guiding data-free model merging via task vectors. In: ICML (2025)
9. Choi, H., Ahn, D., Lee, Y., Kang, T., Cho, S., Choi, J.: Scale: Self-uncertainty conditioned adaptive looking and execution for vision-language-action models. In: ICML (2026)
10. Chronopoulou, A., Pfeiffer, J., Maynez, J., Wang, X., Ruder, S., Agrawal, P.: Language and task arithmetic with parameter-efficient layers for zero-shot summarization. In: EMNLP Workshop (2024)
11. Dey, S., Zaech, J.N., Nikolov, N., Van Gool, L., Paudel, D.P.: Revla: Reverting visual domain limitation of robotic foundation models. In: ICRA (2025)
12. Fei, S., Wang, S., Shi, J., Dai, Z., Cai, J., Qian, P., Ji, L., He, X., Zhang, S., Fei, Z., et al.: Libero-plus: In-depth robustness analysis of vision-language-action models. In: CVPR (2026)
13. Fu, Y., Zhang, Z., Zhang, Y., Wang, Z., Huang, Z., Luo, Y.: Mergevla: Cross-skill model merging toward a generalist vision-language-action agent. In: CVPR (2026)
14. Gao, J., Belkhale, S., Dasari, S., Balakrishna, A., Shah, D., Sadigh, D.: A taxonomy for evaluating generalist robot manipulation policies. RA-L (2026)
15. Gargiulo, A.A., Crisostomi, D., Bucarelli, M.S., Scardapane, S., Silvestri, F., Rodola, E.: Task singular vectors: Reducing task interference in model merging. In: CVPR (2025)
16. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: ICLR (2022)
17. Huang, S.C., Li, P.Z., Hsu, Y.C., Chen, K.M., Lin, Y.T., Hsiao, S.K., Tsai, R., Lee, H.Y.: Chat vector: A simple approach to equip llms with instruction following and model alignment in new languages. In: ACL (2024)
18. Ilharco, G., Ribeiro, M.T., Wortsman, M., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. In: ICLR (2023)

19. Intelligence, P., Amin, A., Aniceto, R., Balakrishna, A., Black, K., Conley, K., Connors, G., Darpinian, J., Dhabalia, K., DiCarlo, J., et al.: $\pi^*_{0.6}$: a vla that learns from experience. arXiv preprint arXiv:2511.14759 (2025)
20. Jang, D.H., Yun, S., Han, D.: Model stock: All we need is just a few fine-tuned models. In: ECCV (2024)
21. Jin, R., Hou, B., Xiao, J., Su, W.J., Shen, L.: Fine-tuning attention modules only: Enhancing weight disentanglement in task arithmetic. In: ICLR (2025)
22. Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M.K., Chen, L.Y., Ellis, K., et al.: Droid: A large-scale in-the-wild robot manipulation dataset. In: RSS (2024)
23. Kim, M.J., Finn, C., Liang, P.: Fine-tuning vision-language-action models: Optimizing speed and success. In: RSS (2025)
24. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., et al.: Openvla: An open-source vision-language-action model. In: CoRL (2024)
25. Lawson, D., Qureshi, A.H.: Merging decision transformers: Weight averaging for forming multi-task policies. In: ICRA (2024)
26. Li, W., Zhang, Q., Zhai, R., Lin, L., Wang, G.: Vla models are more generalizable than you think: Revisiting physical and spatial modeling. In: CVPR (2026)
27. Li, X., Hsu, K., Gu, J., Mees, O., Pertsch, K., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., et al.: Evaluating real-world robot manipulation policies in simulation. In: CoRL (2024)
28. Li, Y., Peng, Z., Zhang, J., Guo, J., Duan, Y., Shi, Y.: When shared knowledge hurts: Spectral over-accumulation in model merging. In: ICML (2026)
29. Lin, T., Li, G., Zhong, Y., Zou, Y., Du, Y., Liu, J., Gu, E., Zhao, B.: Evo-0: Vision-language-action model with implicit spatial understanding. arXiv preprint arXiv:2507.00416 (2025)
30. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023)
31. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. In: NeurIPS (2023)
32. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023)
33. Liu, S., Singh, I.S., Xu, Y., Duan, J., Krishna, R.: Vls: Steering pretrained robot policies via vision-language models. In: CVPRW (2026)
34. Mandlekar, A., Nasiriany, S., Wen, B., Akinola, I., Narang, Y., Fan, L., Zhu, Y., Fox, D.: Mimicgen: A data generation system for scalable robot learning using human demonstrations. In: CoRL (2023)
35. Marczak, D., Magistri, S., Cygert, S., Twardowski, B., Bagdanov, A.D., van de Weijer, J.: No task left behind: Isotropic model merging with common and task-specific subspaces. In: ICML (2025)
36. Nair, S., Rajeswaran, A., Kumar, V., Finn, C., Gupta, A.: R3m: A universal visual representation for robot manipulation. In: CoRL (2022)
37. Ortiz-Jimenez, G., Favero, A., Frossard, P.: Task arithmetic in the tangent space: Improved editing of pre-trained models. In: NeurIPS (2023)
38. O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In: ICRA (2024)
39. Panariello, A., Marczak, D., Magistri, S., Porrello, A., Twardowski, B., Bagdanov, A.D., Calderara, S., van de Weijer, J.: Accurate and efficient low-rank model merging in core space. In: NeurIPS (2025)

40. Pertsch, K., Stachowicz, K., Ichter, B., Driess, D., Nair, S., Vuong, Q., Mees, O., Finn, C., Levine, S.: Fast: Efficient action tokenization for vision-language-action models. In: RSS (2025)
41. Qiu, H., Wu, Y., Li, D., Guo, J., Yao, Q.: Superpose task-specific features for model merging. In: EMNLP (2025)
42. Qu, D., Song, H., Chen, Q., Chen, Z., Gao, X., Ye, X., Lv, Q., Shi, M., Ren, G., Ruan, C., et al.: Eo-1: Interleaved vision-text-action pretraining for general robot control. arXiv preprint arXiv:2508.21112 (2025)
43. Seo, M., Kim, T., Lee, H., Choi, J., Tuytelaars, T.: Not all clients are equal: Collaborative model personalization on heterogeneous multi-modal clients. In: ICLR (2026)
44. Stoica, G., Ramesh, P., Ecsedi, B., Choshen, L., Hoffman, J.: Model merging with svd to tie the knots. In: ICLR (2025)
45. Sun, W., Li, Q., Geng, Y., Li, B.: Cat merging: A training-free approach for resolving conflicts in model merging. In: ICML (2025)
46. Thakkar, M., Fournier, Q., Riemer, M., Chen, P.Y., Zouaq, A., Das, P., Chandar, S.: Combining domain and alignment vectors to achieve better knowledge-safety trade-offs in llms. In: ACL (2025)
47. Walke, H.R., Black, K., Zhao, T.Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A.W., Myers, V., Kim, M.J., Du, M., et al.: Bridgedata v2: A dataset for robot learning at scale. In: CoRL (2023)
48. Wang, L., Zhang, K., Zhou, A., Simchowitz, M., Tedrake, R.: Robot fleet learning via policy merging. In: ICLR (2024)
49. Wei, Y., Cheng, R., Jin, W., Yang, E., Shen, L., Hou, L., Du, S., Yuan, C., Cao, X., Tao, D.: Optmerge: Unifying multimodal llm capabilities and modalities via model merging. In: ICLR (2026)
50. Wilcox, A., Ghanem, M., Moghani, M., Barroso, P., Joffe, B., Garg, A.: Adapt3r: Adaptive 3d scene representation for domain transfer in imitation learning. In: CoRL (2025)
51. Xie, A., Lee, L., Xiao, T., Finn, C.: Decomposing the generalization gap in imitation learning for visual robotic manipulation. In: ICRA (2024)
52. Yadav, P., Tam, D., Choshen, L., Raffel, C.A., Bansal, M.: Ties-merging: Resolving interference when merging models. In: NeurIPS (2023)
53. Yadav, Y., Zhou, Z., Wagenmaker, A., Pertsch, K., Levine, S.: Robust finetuning of vision-language-action robot policies via parameter merging. In: ICLR (2026)
54. Yang, J., Jin, D., Tang, A., Shen, L., Zhu, D., Chen, Z., Zhao, Z., Wang, D., Cui, Q., Zhang, Z., et al.: Mix data or merge models? balancing the helpfulness, honesty, and harmlessness of large language model via model merging. In: NeurIPS (2025)
55. Yang, S., Yu, W., Zeng, J., Lv, J., Ren, K., Lu, C., Lin, D., Pang, J.: Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. In: RSS (2025)
56. Yu, C., Sima, C., Jiang, G., Zhang, H., Mai, H., Li, H., Wang, H., Chen, J., Wu, K., Chen, L., Zhao, L., Shi, M., Luo, P., Bu, Q., Peng, S., Li, T., Yuan, Y.: χ_0: Resource-aware robust manipulation via taming distributional inconsistencies. arXiv preprint arXiv:2602.09021 (2026)
57. Yu, L., Yu, B., Yu, H., Huang, F., Li, Y.: Language models are super mario: Absorbing abilities from homologous models as a free lunch. In: ICML (2024)
58. Yu, T., Xiao, T., Stone, A., Tompson, J., Brohan, A., Wang, S., Singh, J., Tan, C., Peralta, J., Ichter, B., et al.: Scaling robot learning with semantically imagined experience. In: arXiv preprint arXiv:2302.11550 (2023)

59. Yun, S., Chae, S., Lee, D., Ro, Y.: Soma: Singular value decomposed minor components adaptation for domain generalizable representation learning. In: CVPR (2025)
60. Zhang, W., Li, Y., Qiao, Y., Huang, S., Liu, J., Dayoub, F., Ma, X., Liu, L.: Effective tuning strategies for generalist robot manipulation policies. In: ICRA (2025)
61. Zhao, J., Zhang, Z., Chen, B., Wang, Z., Anandkumar, A., Tian, Y.: Galore: Memory-efficient llm training by gradient low-rank projection. In: ICML (2024)
62. Zhao, Y., Zhang, W., Wang, H., Kawaguchi, K., Bing, L.: Adamergex: Cross-lingual transfer with large language models via adaptive adapter merging. In: NAACL (2025)
63. Zhou, X., Xu, Y., Tie, G., Chen, Y., Zhang, G., Chu, D., Zhou, P., Sun, L.: Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. arXiv preprint arXiv:2510.03827 (2025)
64. Zhu, R., Sun, E., Huang, G., Celiktutan, O.: Efficient continual imitation learning with online meta-adapters. RA-L (2026)
65. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: CoRL (2023)