PREMERGE FOR VLA: BUILDING A STRONGER MODEL BY MERGING MODELS WITH A SHARED BASE

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models commonly initialize from a single pretrained vision-language model (VLM), although models from different training stages may retain complementary knowledge. We propose PreMerge for VLA to combine VLMs with a shared base before target-task action training. First, we align their vision-language parameters and assign source weights from the magnitude and direction changes of the original weights relative to the base. A sign-consistency mask removes conflicting parameter updates. Second, we use a small set of target images and instructions, without action annotations, to calibrate the merged model through multilayer feature distillation. Source parameters remain frozen, and only tensor-level merging coefficients are optimized. Using the aligned VLM components of Qwen3-VL, Cosmos-Reason2, and GR00T-N1.7, PreMerge achieves average success rates of 98.9% on LIBERO and 92.7% on RoboTwin 2.0, exceeding the best single-source initialization by 1.0 and 3.2 percentage points, respectively. On four real-world manipulation tasks, it achieves 90.0%, compared with 82.5% for the strongest evaluated external baseline. The merged parameters are folded into one VLM, so subsequent action training and inference require no additional source branches. These results demonstrate the value of reusing models from different training stages to construct VLA initializations.

1 INTRODUCTION

Vision-language-action (VLA) models connect visual observations and language instructions to robot actions, commonly starting from one pretrained vision-language model (VLM) (Kim et al., 2025b; Black et al., 2025b;a; NVIDIA et al., 2025). Yet successive training stages can leave different capabilities in models from the same lineage. Qwen3-VL provides a general vision-language base; Cosmos-Reason2 extends it with physical reasoning; GR00T-N1.7 adopts the Cosmos backbone for robot learning (Qwen Team, 2025; NVIDIA, 2026a;b). Selecting only one stage may overlook useful information retained by the others.

Our objective is to reuse these existing models before target-task VLA training while retaining a single VLM during action learning and inference. This differs from retaining multiple experts or verifiers (Guo et al., 2026; Kwok et al., 2025), and from merging task-trained policies to combine their skills (Fu et al., 2026; Yadav et al., 2026). Shared-base model merging provides a parameter-space mechanism for this reuse (Ilharco et al., 2023; Yu et al., 2024). Module-dependent changes and conflicting updates motivate adaptive source weighting and sign filtering. Target images and instructions provide task-specific signals for calibrating source contributions.

We propose **PreMerge**, comprising adaptive weight-level merging and task-level calibration without action annotations (Figure 1). Weight-level merging constructs fixed, conflict-filtered source residuals. Calibration then uses frozen teacher features to optimize tensor-level merging coefficients through joint multilayer distillation. Folding the calibrated residuals into the shared base yields one VLM for supervised action training. "Before training" refers to target-task action training; sources may already have undergone embodied reasoning or robot training.

We evaluate LIBERO, RoboTwin 2.0, and four real-world tasks. PreMerge achieves average success rates of 98.9%, 92.7%, and 90.0%, respectively. Its gains over equal-weight initialization are 1.5, 4.9, and 10.0 percentage points. Our contributions are: (i) parameter and representation analyses identifying heterogeneous updates and input-dependent merging deviations; (ii) an initialization method combining adaptive merging with few-shot, action-free calibration; and (iii) controlled ini-

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.

tialization comparisons, component ablations, action-head comparisons, and real-world evaluations of this model-reuse approach.

2 Related Work

Vision-language-action models. OpenVLA, π_0, $\pi_{0.5}$, and GR00T learn action policies from pretrained vision-language representations (Kim et al., 2025b; Black et al., 2025b;a; NVIDIA et al., 2025). Efficient adaptation, spatial representations, and physical pretraining further explore how to construct useful backbones (Wang et al., 2026; Qu et al., 2025; Feng et al., 2026; Fang et al., 2026). Prior studies examine the relationship between VLM understanding and downstream robot control (Lin et al., 2026a; Zhang et al., 2026). We evaluate VLM initializations through downstream success rates after target-task action learning.

Model merging. Model Soups and Task Arithmetic combine models with compatible parameters (Wortsman et al., 2022; Ilharco et al., 2023); WIDEN scores magnitude and direction changes, while TIES and SCE address conflicting updates (Yu et al., 2024; Yadav et al., 2023; Wan et al., 2025). ProDistill uses unlabeled feature distillation to learn merging coefficients (Xu et al., 2025). MergeVLA and RETAIN focus on integrating or retaining skills in task-trained VLAs, while CapVector uses parameter differences for downstream initialization (Fu et al., 2026; Yadav et al., 2026; Song et al., 2026). PreMerge instead combines aligned VLM components from a shared-base lineage before target-task action training. Its calibration jointly supervises multiple outputs of the complete VLM, which is then connected to a target-task action head.

3 Problem Formulation and Motivation

3.1 Shared Base and Parameter Composition

Let θ_0 be the shared Qwen pretrained base, and let θ_1 and θ_2 denote the aligned Cosmos and GR00T vision-language parameters. For source $i \in \{0, 1, 2\}$, define its cumulative update relative to the base as $\Delta_i^{\text{pre}} = \theta_i - \theta_0$, with $\Delta_0^{\text{pre}} = 0$. A convex parameter combination can then be written as

$$\theta_{\boldsymbol{\alpha}} = \sum_{i=0}^{2} \alpha_i \theta_i = \theta_0 + \sum_{i=1}^{2} \alpha_i \Delta_i^{\text{pre}}, \qquad \alpha_i \geq 0, \quad \sum_{i=0}^{2} \alpha_i = 1. \tag{1}$$

Equal weighting sets all three coefficients to $1/3$; a pairwise midpoint assigns $1/2$ to each selected source. The updates are cumulative relative to Qwen, including any intervening training stages. We use $\mathcal{S} = \{1, 2\}$ for the non-base sources.

Alignment requires matching parameter names, shapes, and semantic positions. We retain the vision encoder, projector, token embeddings, first 16 language blocks, and final normalization, excluding action and language-generation heads. Diagnostics below separately examine Vision, Projector, and the 16 language blocks; their exact parameter scopes and feature boundaries are given in **Appendix Table B.1**. **Appendix B.1** details the shared-base formulation (**B.1.1**), interpolation assumptions (**B.1.2**), and diagnostic sampling (**B.1.3**).

3.2 Parameter Changes and Update Conflicts

For a module with flattened base weights and updates, relative update magnitude is $\|\Delta_i^{\text{pre}}\|_2 / \|\theta_0\|_2$. A coordinate is conflicting when the two non-base updates have opposite signs; the conflict ratio divides their count by all coordinates, including zero-update coordinates. To measure cancellation of the combined update, define

$$R = \frac{\|\sum_i \alpha_i \Delta_i^{\text{pre}}\|_2}{\sum_i \alpha_i \|\Delta_i^{\text{pre}}\|_2}, \qquad C = 1 - R. \tag{2}$$

For a positive denominator, the triangle inequality gives $0 \leq R \leq 1$. Sign conflict measures local disagreement, while C measures aggregate cancellation and depends on both update directions and magnitudes.

Table 1 shows larger GR00T updates in vision and larger Cosmos updates in the projector and language modules, motivating adaptive weighting. Values are percentages; CTL reports mean $\pm$ sample standard deviation. **Appendix Figure B.3(a)** shows persistent sign conflicts even among high-energy updates, motivating sign erasure. Layerwise details are in **Appendix B.2**.

Table 1: Module diagnostics (%).

Module	Cosmos update	GR00T update	Cancellation	LIBERO CTL
Vision	0.08	0.49	12.25	3.09 ± 2.34
Projector	3.28	2.09	27.13	6.02 ± 0.51
Language	2.12	1.36	26.63	1.53 ± 0.13

3.3 Representation Linearity and Similarity

For image–instruction input $x = (I, s)$, let $h_\ell(x;\theta) \in \mathbb{R}^{d_\ell}$ be the pooled representation at output ℓ, with dimension d_ℓ. For distinct sources $A, B \in \{0, \dots, K-1\}$, $K = 3$, cross-task linearity (CTL) (Zhou et al., 2024) relates parameter and representation interpolation:

$$\theta_\alpha = (1-\alpha)\theta_A + \alpha\theta_B, \quad \alpha \in [0,1], \qquad h_\ell(x;\theta_\alpha) \approx (1-\alpha)h_\ell(x;\theta_A) + \alpha h_\ell(x;\theta_B). \tag{3}$$

Under the pairwise CTL and transitivity conditions of the original work, a multisource approximation follows. Write $h_i = h_\ell(x;\theta_i)$, $h_{\boldsymbol{\alpha}} = h_\ell(x;\theta_{\boldsymbol{\alpha}})$, and $\Delta h_i = \Delta h_i^{(\ell)}(x)$, with sums over $i = 0, \dots, K-1$:

$$h_{\boldsymbol{\alpha}} \approx \sum_i \alpha_i h_i, \qquad \Delta h_i = h_i - h_0, \qquad h_{\boldsymbol{\alpha}} - h_0 \approx \sum_i \alpha_i \Delta h_i. \tag{4}$$

Here Δh_i is the source representation change relative to the base; the last relation uses $\sum_i \alpha_i = 1$. We quantify departures from this approximation by

$$\mathrm{Err}_{\mathrm{CTL}} = \frac{\|h_\ell(x;\theta_{\boldsymbol{\alpha}}) - \sum_i \alpha_i h_\ell(x;\theta_i)\|_2}{\sum_i \alpha_i \|h_\ell(x;\theta_i)\|_2}. \tag{5}$$

We measure this approximation through actual merged-model forward passes. LIBERO diagnostics use 64 image–instruction samples from four tasks, with 16 independent demonstrations per task and one frame per demonstration. All models use identical 224×224 preprocessing. Features are pooled at the final vision output, the projector branches, and language block 16. CTL statistics report per-sample means and sample standard deviations.

To compare representation structure across samples, we also use centered linear CKA (Kornblith et al., 2019). Let rows of H_A, H_B contain corresponding sample features, and let $\widetilde{H}_A, \widetilde{H}_B$ denote column-centered matrices. Then

$$\mathrm{CKA}(H_A, H_B) = \frac{\|\widetilde{H}_A^\top \widetilde{H}_B\|_F^2}{\|\widetilde{H}_A^\top \widetilde{H}_A\|_F \, \|\widetilde{H}_B^\top \widetilde{H}_B\|_F}. \tag{6}$$

CTL measures the deviation of a merged representation from the interpolated source features; CKA measures similarity of cross-sample structure. On LIBERO, Qwen–GR00T CKA is 0.51 in Vision, 0.98 in Projector, and 0.99 in Language. Equal-weight CTL errors remain 3.09%, 6.02%, and 1.53%, respectively. High source similarity therefore coexists with measurable merging deviations, particularly at the projector (**Appendix Tables B.3 and B.5**).

Appendix Figure B.6(a) compares all three source pairs on LIBERO. Vision CKA varies from 0.21 to 0.57, whereas projector and language CKA remain between 0.98 and 0.99. These differences motivate target-conditioned calibration. **Appendix B.3** expands CTL (**B.3.1**), its magnitude/direction decomposition (**B.3.2**), source CKA (**B.3.3**), and internal cross-module CKA (**B.3.4**).

3.4 Task Dependence and Design Implications

Appendix Figure B.8(a) shows task-dependent CTL errors on LIBERO, motivating target-conditioned calibration. Task-centered Language CKA also varies (0.65–0.86 versus 0.99 when pooled), revealing differences obscured by between-task variation (**Appendix Table B.8**).

Together, these observations motivate two stages: adaptive weighting and sign erasure account for heterogeneous parameter updates, while target-conditioned feature calibration adjusts the resulting initialization on task inputs. LIBERO provides the main diagnostic case; **Appendix B** retains complete diagnostics; **Appendix B.4** compares task-conditioned CTL (**B.4.1**) and CKA (**B.4.2**) across all three datasets. These diagnostics motivate the design; the component ablations in Section 5 evaluate its downstream utility.

4 METHOD

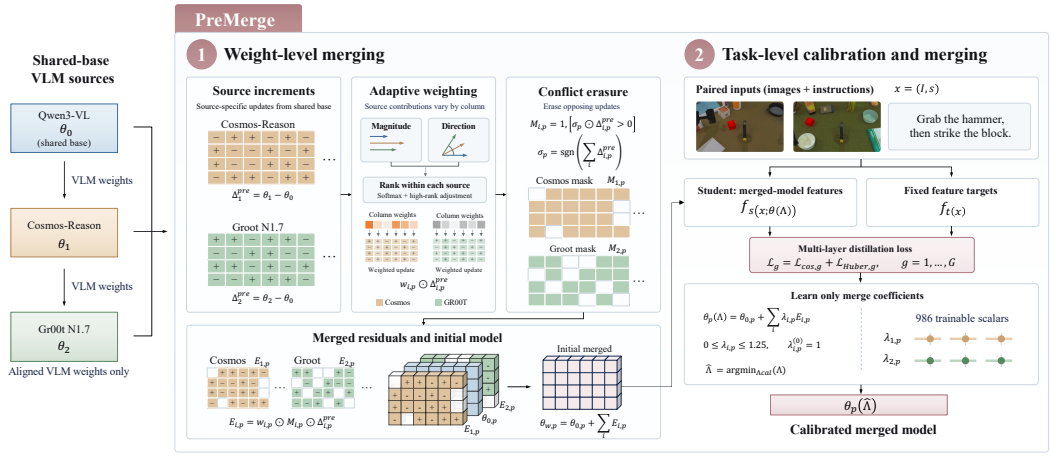


Figure 1: Overview of PreMerge for VLA.

Figure 1 summarizes the two merging stages and subsequent action training. We first construct fixed source residuals, then calibrate their tensor-level scales, and finally fold the result into one VLM. **Appendix A.4 (Figure A.1)** provides the detailed pipeline.

4.1 WEIGHT-LEVEL MERGING

Using the aligned sources from Section 3, let $\mathcal{S} = \{1, 2\}$ and $\Delta^{\text{pre}}_{i,p} = \theta_{i,p} - \theta_{0,p}$ for tensor $p \in \{1, \dots, P\}$. We combine WIDEN source scoring with SCE sign erasure (Yu et al., 2024; Wan et al., 2025); **Appendix A.5 (Figure A.2)** illustrates the procedure.

Source weights. For a matrix view $W_{i,p} \in \mathbb{R}^{o_p \times c_p}$, where o_p and c_p are row and column counts, column $j \in \{1, \dots, c_p\}$ receives magnitude and direction scores relative to the base:

$$v^{\text{mag}}_{i,p,j} = \left| \|W_{i,p,:,j}\|_2 - \|W_{0,p,:,j}\|_2 \right|, \qquad v^{\text{dir}}_{i,p,j} = 1 - \cos_{\varepsilon_c}\left(W_{i,p,:,j}, W_{0,p,:,j}\right). \tag{7}$$

Here $W_{i,p,:,j}$ is column j. The stabilized cosine $\cos_\varepsilon$ divides the inner product by the product of the two norms, each floored at $\varepsilon > 0$; parameter directions use $\varepsilon_c = 10^{-8}$.

For $\nu \in \{\text{mag}, \text{dir}\}$, ascending within-tensor ranks are $\rho^{\nu}_{i,p,j} \in \{0, 1/c_p, \dots, (c_p - 1)/c_p\}$, with ties resolved by fixed column order. Let $\bar{\rho}^{\nu}_{i,p} = c_p^{-1} \sum_j \rho^{\nu}_{i,p,j}$ be their mean. Intermediate scores $q^{\nu}_{i,p,j}$ and normalized column weights $w_{i,p,j}$ are

$$q^{\nu}_{i,p,j} = \begin{cases} \kappa, & \rho^{\nu}_{i,p,j} > \tau \bar{\rho}^{\nu}_{i,p}, \\ \dfrac{\exp(\rho^{\nu}_{i,p,j})}{\sum_{i' \in \mathcal{S}} \exp(\rho^{\nu}_{i',p,j})}, & \text{otherwise}, \end{cases} \qquad w_{i,p,j} = \frac{q^{\text{mag}}_{i,p,j} + q^{\text{dir}}_{i,p,j}}{\sum_{i' \in \mathcal{S}} \left(q^{\text{mag}}_{i',p,j} + q^{\text{dir}}_{i',p,j}\right)}. \tag{8}$$

The threshold multiplier τ and replacement score κ both equal 1; i' indexes non-base sources. Thus $w_{i,p,j} \geq 0$ and $\sum_i w_{i,p,j} = 1$. Linear weights retain their matrix layout, embeddings are transposed, and higher-dimensional tensors flatten all but the first axis. For one-dimensional biases and normalization parameters, absolute updates replace ranks in the Softmax and above-mean replacement, without separate magnitude and direction terms.

Conflict erasure. For row $r \in \{1, \dots, o_p\}$, the unweighted update sum determines a reference sign $\sigma_{p,r,j}$ and a binary retention mask $M_{i,p,r,j}$:

$$\sigma_{p,r,j} = \operatorname{sgn}\left(\sum_{i \in \mathcal{S}} \Delta^{\text{pre}}_{i,p,r,j}\right), \qquad M_{i,p,r,j} = \mathbf{1}\left[\sigma_{p,r,j} \Delta^{\text{pre}}_{i,p,r,j} > 0\right]. \tag{9}$$

Here sgn is the sign function and $\mathbf{1}[\cdot]$ the indicator; a zero reference sign retains no update. Broadcast column weights across rows and restore the original tensor shapes to obtain $w_{i,p}$ and $M_{i,p}$. With $\odot$ denoting elementwise multiplication, define the fixed residual and initial merge:

$$E_{i,p} = w_{i,p} \odot M_{i,p} \odot \Delta^{\text{pre}}_{i,p}, \qquad \theta_{\text{w},p} = \theta_{0,p} + \sum_{i \in \mathcal{S}} E_{i,p}. \tag{10}$$

Weights are not renormalized after erasure. The effective source share is $\gamma_{i,p,r,j} = w_{i,p,j} M_{i,p,r,j}$, and the base share is $\gamma_{0,p,r,j} = 1 - \sum_{i \in \mathcal{S}} \gamma_{i,p,r,j} \geq 0$. One-dimensional parameters use the same elementwise mask. All $E_{i,p}$ remain fixed during calibration.

4.2 Task-Level Calibration and Merging

Calibration uses task-balanced image–instruction samples $x = (I, s)$ from $\mathcal{D}_{\text{cal}}$, with identical preprocessing across models and no action labels. Following feature-distillation-based merging (Xu et al., 2025), we optimize one scale per source and tensor, collected in $\Lambda \in \mathbb{R}^{|\mathcal{S}| \times P}$:

$$\theta_p(\Lambda) = \theta_{0,p} + \sum_{i \in \mathcal{S}} \lambda_{i,p} E_{i,p}, \qquad 0 \leq \lambda_{i,p} \leq \lambda_{\max}. \tag{11}$$

We use $P = 493$, $|\mathcal{S}| = 2$, and $\lambda_{\max} = 1.25$. The 986 scales start at 1, so $\theta(\mathbf{1}) = \theta_{\text{w}}$; base parameters and residuals remain frozen.

Teacher targets. Let $F_g(x; \theta) \in \mathbb{R}^{T_g(x) \times d_g^{\text{feat}}}$ be output g, with $T_g(x)$ positions and d_g^{feat} channels. The $G = 44$ outputs comprise the visual input projection, 24 vision blocks, projector, token embeddings, 16 language blocks, and final normalization. Positions use the same flattening order across models.

The normalization $\mathcal{N}_\varepsilon$ divides a vector by $\max(\| \cdot \|_2, \varepsilon)$. At position t, define source features $z_{i,g,t} = \mathcal{N}_{\varepsilon_f}(F_{g,t}(x; \theta_i))$, initial merged features $z_{\text{w},g,t} = \mathcal{N}_{\varepsilon_f}(F_{g,t}(x; \theta_{\text{w}}))$, and changes $\Delta z_{i,g,t} = z_{i,g,t} - z_{0,g,t}$, using $\varepsilon_f = 10^{-6}$. The fixed target is

$$Y_{g,t} = \begin{cases} \mathcal{N}_{\varepsilon_f}\left(\dfrac{z_{1,g,t} + z_{2,g,t}}{2} \right), & \cos_{\varepsilon_f}(\Delta z_{1,g,t}, \Delta z_{2,g,t}) \geq \xi, \\ z_{\text{w},g,t}, & \text{otherwise}, \end{cases} \qquad \xi = 0. \tag{12}$$

The threshold ξ tests directional consistency: consistent changes use the normalized source mean; otherwise the target is the initial merged feature. Targets are precomputed and frozen (**Appendix A.6 (Figure A.3)**).

Distillation loss. Normalize student and target features as $Z_{g,t} = \mathcal{N}_{\varepsilon_n}(F_{g,t}(x; \theta(\Lambda)))$ and $\widehat{Y}_{g,t} = \mathcal{N}_{\varepsilon_n}(Y_{g,t})$, with $\varepsilon_n = 10^{-12}$. Their per-output loss combines cosine distance and Huber error:

$$\mathcal{L}_g(x; \Lambda) = \frac{1}{T_g(x)} \sum_{t=1}^{T_g(x)} \left[1 - \cos_{\varepsilon_c}(Z_{g,t}, \widehat{Y}_{g,t}) \right] + \frac{1}{T_g(x) d_g^{\text{feat}}} \sum_{t=1}^{T_g(x)} \sum_{k=1}^{d_g^{\text{feat}}} \text{Huber}_\delta \left(Z_{g,t,k} - \widehat{Y}_{g,t,k} \right). \tag{13}$$

Here k indexes channels and cosine uses $\varepsilon_c = 10^{-8}$. For scalar error e, $\text{Huber}_\delta(e)$ is $e^2/2$ when $|e| \leq \delta$ and $\delta(|e| - \delta/2)$ otherwise, with $\delta = 0.1$. Both loss terms have unit weight.

On a fixed reference set $\mathcal{D}_{\text{ref}}$, compute $b_g = |\mathcal{D}_{\text{ref}}|^{-1} \sum_{x \in \mathcal{D}_{\text{ref}}} \mathcal{L}_g(x; \mathbf{1})$ before optimization; $\mathbf{1}$ has the shape of Λ. With task-balanced sampling distribution π_{cal} and denominator floor $\varepsilon_b = 10^{-3}$, the objective is

$$\mathcal{L}_{\text{cal}}(\Lambda) = \mathbb{E}_{x \sim \pi_{\text{cal}}} \left[\frac{1}{G} \sum_{g=1}^{G} \frac{\mathcal{L}_g(x; \Lambda)}{\max(b_g, \varepsilon_b)} \right] + \mu_{\text{anc}} \mathcal{R}_{\text{anc}}(\Lambda) + \mu_{\text{grp}} \mathcal{R}_{\text{grp}}(\Lambda) + \mu_{\text{adj}} \mathcal{R}_{\text{adj}}(\Lambda). \tag{14}$$

All regularization weights $\mu_{\text{anc}}, \mu_{\text{grp}}, \mu_{\text{adj}}$ equal 0.01.

Regularization and optimization. The anchor, within-group, and adjacent-group penalties constrain deviation from initial scales, variation within structural groups, and differences between neighboring group means, respectively. Their definitions are given in **Appendix A.9**. AdamW jointly updates all scales for 20,000 steps at learning rate 10^{-3}, with 1000 warmup steps and cosine scheduling. Each update projects scales onto $[0, 1.25]$; the final scales are $\widehat{\Lambda}$. Calibration, reference, and audit sets have disjoint sample identities; sampling details are in **Appendix A.8**.

4.3 Target-Task VLA Training

Fold the calibrated weights $\theta_{\text{cal}} = \theta(\widehat{\Lambda})$ into one VLM; equivalently, $\theta_p(\Lambda) = \theta_{\text{w},p} + \sum_{i \in \mathcal{S}} (\lambda_{i,p} - 1) E_{i,p}$. Let ϕ collect action-head and adapter parameters initialized at ϕ_0, and let $a \in \mathbb{R}^{T_a \times d_a}$ be an action sequence of length T_a and step dimension d_a. On action data $\mathcal{D}_{\text{act}}$, jointly optimize

$$\min_{\theta, \phi} \ \mathbb{E}_{(x,a) \sim \mathcal{D}_{\text{act}}} \left[\mathcal{L}_{\text{act}}\big(f_\theta(x), a; \phi\big) \right], \qquad (\theta, \phi)_{\text{init}} = (\theta_{\text{cal}}, \phi_0). \tag{15}$$

Here $f_\theta(x)$ denotes VLM features before action-side adapters and $\mathcal{L}_{\text{act}}$ is the selected head's loss. Training and inference use one VLM and one action head (**Appendix A.7 (Figure A.4)**).

Action interfaces. For complete token features $f_\theta^{(\ell)}(x)$ at language layer $\ell \in \{1, \dots, L\}$, $L = 16$, the GR00T, PI, and OFT conditioning features are (NVIDIA et al., 2025; Black et al., 2025b; Kim et al., 2025a)

$$\mathcal{C}_{\text{GR00T}}(x) = f_\theta^{(L)}(x), \qquad \mathcal{C}_{\text{PI}}(x) = \Big(\Pi_\ell\Big(f_\theta^{(\ell)}(x)\Big)\Big)_{\ell=1}^{L}, \qquad \mathcal{C}_{\text{OFT}}(x) = \mathcal{Q}\Big(f_\theta^{(L)}(x^{\text{qry}})\Big). \tag{16}$$

GR00T uses final-layer features through cross-attention. PI applies layerwise normalization and projection Π_ℓ, whose parameters belong to ϕ. For OFT, x^{qry} appends T_a action queries and $\mathcal{Q}$ extracts their final features in order.

GR00T and PI use flow matching. Sample standard Gaussian noise n shaped like a and time $\zeta \in [0, 1]$ from the selected head's schedule, and set $a_\zeta = (1 - \zeta)n + \zeta a$. For head $\omega \in \{\text{GR00T}, \text{PI}\}$ with velocity predictor $\mathcal{V}_\phi^\omega$, the loss is

$$\mathcal{L}_{\text{act}}^\omega = \mathbb{E}_{\zeta,n}\left[\frac{\left\|\mathcal{V}_\phi^\omega(a_\zeta, \zeta; \mathcal{C}_\omega(x)) - (a - n)\right\|_F^2}{T_a d_a}\right]. \tag{17}$$

OFT predicts $\widehat{a} = \mathcal{A}_\phi(\mathcal{C}_{\text{OFT}}(x))$ using an MLP $\mathcal{A}_\phi$ and minimizes

$$\mathcal{L}_{\text{act}}^{\text{OFT}} = \frac{\|\widehat{a} - a\|_1}{T_a d_a}. \tag{18}$$

Norms are taken over all action elements, and both losses are normalized by $T_a d_a$.

5 Experiments

5.1 Experimental Setup

We evaluate all 40 LIBERO tasks (Liu et al., 2023), all 50 RoboTwin 2.0 tasks in Clean and Randomized domains (Chen et al., 2025), and four real-world tasks. Initialization comparisons use the same GR00T action head, training data, and training budget, varying the aligned VLM initialization. Calibration uses training images and instructions without action labels. **Appendix A.1** specifies data, action training, evaluation, and reporting; **Appendix A.8** details calibration implementation. ***Code is provided in the supplementary materials.***

5.2 Simulation Results

In Tables 2 and 3, PreMerge reaches 98.9% on LIBERO and 92.7% on RoboTwin, improving on the best single source by 1.0 and 3.2 percentage points, and on Equal Weight by 1.5 and 4.9 points. Published-result protocols are detailed in **Appendix A.2**.

On LIBERO, PreMerge matches GR00T on Spatial and improves Object, Goal, and Long by 0.8, 0.4, and 2.8 points. The largest gain is on Long (94.8% to 97.6%), highlighting improved extended action sequences. Equal Weight underperforms GR00T on both benchmarks; PreMerge surpasses both. On RoboTwin, gains over Equal Weight reach 4.8 points in Clean and 5.1 in Randomized, with a smaller domain gap of 0.4 points.

5.3 Real-World Experiments

We evaluate placement, replacement, sorting, and stacking on a tabletop platform (Figure 2), using 100 demonstrations and 20 evaluation trials per task. Calibration uses eight demonstrations. **Appendix A.3** details task definitions, preprocessing, and sampling.

5.4 Component Ablations

In Table 5, conflict erasure raises success over Equal Weight from 97.4% to 98.0% on LIBERO and from 87.8% to 89.7% on RoboTwin. Adding adaptive weighting brings a further 0.3 and 1.2 points, respectively, supporting adaptive allocation. Calibrating this initialization yields another 0.6 and 1.8 points, reaching 98.9% and 92.7%.

Table 2: LIBERO success rates (%).

Method	Venue/year	Spatial	Object	Goal	Long	Avg. ↑
OpenVLA (Kim et al., 2025b)	*CoRL* '24	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT (Kim et al., 2025a)	*RSS* '25	97.6	98.4	97.9	94.5	97.1
$\pi_{0.5}$ (Black et al., 2025a)	*CoRL* '25	98.8	98.2	98.0	92.4	96.9
π_0 (Black et al., 2025b)	*RSS* '25	96.8	98.8	95.8	85.2	94.2
ACoT-VLA (Frozen) (Zhong et al., 2026)	*CVPR* '26	99.4	99.6	98.8	96.0	98.5
LingBot-VA (Li et al., 2026)	*RSS* '26	98.5	99.6	97.2	**98.5**	98.5
VLA-Adapter-Pro (Wang et al., 2026)	*AAAI* '26	**99.6**	99.6	98.2	96.4	98.5
X-VLA (Zheng et al., 2026)	*ICLR* '26	98.2	98.6	97.8	97.6	98.1
MemoryVLA (Shi et al., 2026)	*ICLR* '26	98.4	98.4	96.4	93.4	96.7
FastWAM-Joint (Yuan et al., 2026)	*arXiv* '26	**99.6**	99.4	98.2	96.8	98.5
ABot-M0 (Yang et al., 2026b)	*arXiv* '26	98.8	**99.8**	99.0	96.6	98.6
MolmoAct2-Think (Fang et al., 2026)	*arXiv* '26	98.8	**99.8**	98.5	95.4	98.1
Qwen3-VL	—	97.5	96.7	96.7	92.9	96.0
Cosmos-Reason2	—	98.5	98.3	98.1	94.1	97.3
GR00T-N1.7	—	99.0	98.6	99.0	94.8	97.9
Equal-weight initialization	—	98.4	99.0	98.2	94.0	97.4
PreMerge (Ours)	—	99.0	99.4	**99.4**	97.6	**98.9**+1.5

Table 3: RoboTwin 2.0 success rates (%).

Method	Venue/year	Clean	Randomized	Avg. ↑
π_0 (Black et al., 2025b)	*RSS* '25	65.9	58.4	62.2
$\pi_{0.5}$ (Black et al., 2025a)	*CoRL* '25	82.7	76.8	79.8
LingBot-VA (Li et al., 2026)	*RSS* '26	**92.9**	91.6	92.2
Motus (Bi et al., 2026)	*CVPR* '26	88.7	87.0	87.8
HoloBrain-0-QW (Lin et al., 2026b)	*arXiv* '26	91.9	92.3	92.1
FastWAM (Yuan et al., 2026)	*arXiv* '26	91.9	91.8	91.8
ABot-M0 (Yang et al., 2026b)	*arXiv* '26	86.1	85.1	85.6
LingBot-VLA (Wu et al., 2026)	*arXiv* '26	88.6	86.7	87.6
FlowWAM (Chen et al., 2026)	*arXiv* '26	**92.9**	92.1	92.5
InternVLA-A1 (Cai et al., 2026)	*arXiv* '26	89.4	89.6	89.5
VLAct (Yang et al., 2026a)	*arXiv* '26	92.5	90.8	91.7
Qwen3-VL	—	84.4	82.9	83.7
Cosmos-Reason2	—	86.5	86.0	86.3
GR00T-N1.7	—	89.7	89.3	89.5
Equal-weight initialization	—	88.1	87.4	87.8
PreMerge (Ours)	—	**92.9**+4.8	**92.5**+5.1	**92.7**+4.9

Table 4: Real-world success rates (%).

Method / VLM initialization	Placement	Replacement	Sorting	Stacking	Avg. ↑
RDT-1B (Liu et al., 2025)	75.0	55.0	70.0	50.0	62.5
X-VLA (Zheng et al., 2026)	85.0	65.0	80.0	60.0	72.5
VLA-Adapter (Wang et al., 2026)	70.0	80.0	85.0	60.0	73.8
OpenVLA-OFT (Kim et al., 2025a)	75.0	85.0	90.0	65.0	78.8
$\boldsymbol{\pi_0}$ (Black et al., 2025b)	90.0	70.0	85.0	75.0	80.0
$\boldsymbol{\pi_{0.5}}$ (Black et al., 2025a)	90.0	75.0	80.0	85.0	82.5
Qwen3-VL	80.0	65.0	85.0	55.0	71.3
Cosmos-Reason2	85.0	80.0	70.0	75.0	77.5
GR00T-N1.7	85.0	90.0	80.0	70.0	81.3
Equal-weight initialization	80.0	75.0	90.0	75.0	80.0
PreMerge (Ours)	**90.0**	**90.0**	**95.0**	**85.0**	**90.0**+10.0

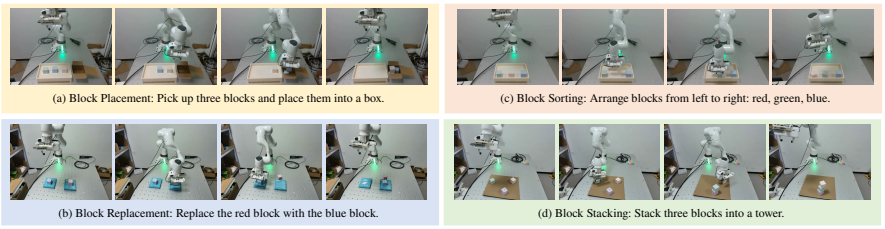


Figure 2: Real-world manipulation tasks.

PreMerge achieves 90.0% average success, compared with 80.0% for Equal Weight, 81.3% for GR00T, and 82.5% for $\pi_{0.5}$ (Table 4). Gains over Equal Weight are 10, 15, 5, and 10 points on placement, replacement, sorting, and stacking, respectively. Among single sources, GR00T leads on replacement, Qwen on sorting, and Cosmos on stacking. PreMerge matches the strongest single source on replacement and exceeds the single-source maximum on the other three tasks. Against $\pi_{0.5}$, it improves replacement and sorting by 15 points each and matches placement and stacking, achieving the highest or tied-highest success on every task.

Table 5: Component ablations (%).

Method / VLM initialization	LIBERO					RoboTwin 2.0		
	Spatial	Object	Goal	Long	Avg. ↑	Clean	Randomized	Avg. ↑
Base model (Qwen3-VL)	97.5	96.7	96.7	92.9	96.0	84.4	82.9	83.7
Best single source (GR00T-N1.7)	**99.0**	98.6	99.0	94.8	97.9	89.7	89.3	89.5
Equal-weight initialization	98.4	99.0	98.2	94.0	97.4	88.1	87.4	87.8
Conflict erasure only	98.8	99.2	98.8	95.0	98.0 −0.9	90.0 −2.9	89.4 −3.1	89.7 −3.0
Conflict erasure + adaptive weighting	98.8	99.3	99.0	96.1	98.3 −0.6	91.2 −1.7	90.6 −1.9	90.9 −1.8
Task calibration only	98.7	99.2	98.8	95.3	98.0 −0.9	90.3 −2.6	89.7 −2.8	90.0 −2.7
PreMerge (Ours)	**99.0**	**99.4**	**99.4**	**97.6**	**98.9**	**92.9**	**92.5**	**92.7**

Task calibration alone reaches 98.0% and 90.0%, trailing full PreMerge by 0.9 and 2.7 points. Thus, target-conditioned calibration benefits from the conflict-filtered, adaptively weighted initialization. On LIBERO Long, adding calibration to weight-level merging raises success from 96.1% to 97.6%; on RoboTwin, it adds 1.7 points in Clean and 1.9 in Randomized.

5.5 Action-Head Comparison

Table 6: Action-head comparison (%).

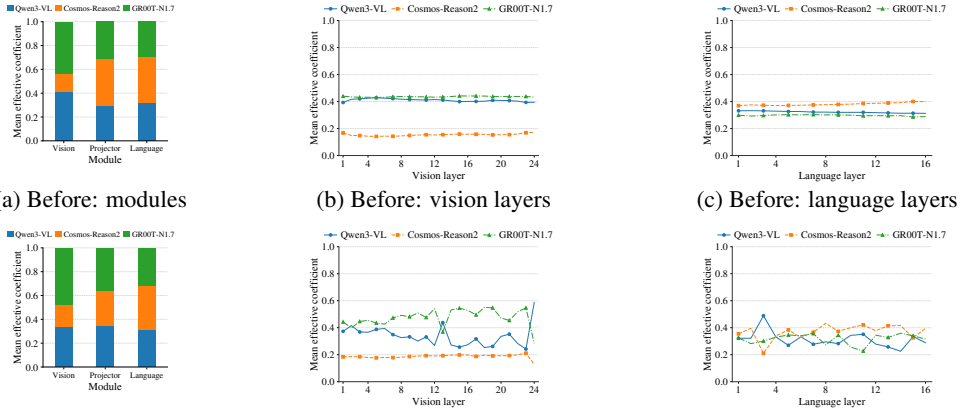
Action head	LIBERO					RoboTwin 2.0		
	Spatial	Object	Goal	Long	Avg. ↑	Clean	Randomized	Avg. ↑
GR00T	**99.0**	99.4	**99.4**	**97.6**	**98.9**	**92.9**	**92.5**	**92.7**
PI	**99.0**	99.6	98.6	97.0	98.6	92.6	91.8	92.2
OFT	97.6	**99.8**	99.0	96.6	98.3	92.2	90.8	91.5

Table 6 compares GR00T, PI, and OFT on both simulation benchmarks. GR00T has the highest mean on both and is used in the main initialization comparisons. PI and OFT provide additional interfaces for the merged VLM; some individual LIBERO suites favor another head.

5.6 Merging and Calibration Analysis

We use LIBERO as the representative calibration case in the main text. **Appendix C** retains complete LIBERO, RoboTwin, and real-world results; **Appendix D** compares visual (**D.1**) and final-layer VLM features (**D.2**) before and after action training on LIBERO and RoboTwin.

Shared initialization. Figure 3 compares module, vision-layer, and language-layer coefficients before and after LIBERO calibration. Table 7 weights module statistics by parameter coordinate counts. The datasets share one weight-only initialization (**Appendix C.1**): GR00T has the largest Vision coefficient (0.43), while Cosmos has the largest Projector and Language coefficients (0.40 and 0.38). These coefficients give algebraic parameter proportions.



(a) Before: modules (b) Before: vision layers (c) Before: language layers

(d) After: modules (e) After: vision layers (f) After: language layers

Figure 3: Source coefficients before (top) and after LIBERO calibration (bottom).

Module	Stage	Qwen	Cosmos	GR00T
Vision	Before	0.41	0.15	0.43
	After	0.34(−0.07)	0.19(+0.03)	0.47(+0.04)
Projector	Before	0.29	0.40	0.31
	After	0.34(+0.05)	0.30(−0.10)	0.36(+0.05)
Language	Before	0.32	0.38	0.30
	After	0.31(−0.01)	0.37(−0.01)	0.32(+0.02)

Table 7: LIBERO module coefficients.

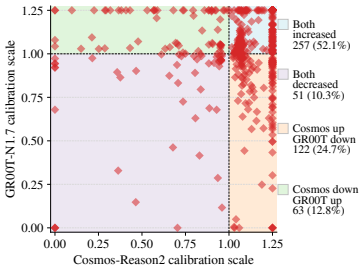


Figure 4: Paired source scales.

Target-conditioned reallocation. After calibration, LIBERO Vision coefficients shift toward Cosmos and GR00T, Projector shifts away from Cosmos, and Language changes remain smaller (Table 7). Among tensor pairs, 52.13% amplify both sources and 37.53% adjust them in opposite directions (Figure 4), showing joint amplification and source reallocation (**Appendix Table C.4**). Across all three datasets, **Appendix C.2** reports effective coefficients, **Appendix C.3** scale distributions, **Appendix C.4** paired adjustments, and **Appendix C.5** tensor-level surfaces.

6 Limitations and Future Work

Limitations. PreMerge requires a shared base and aligned parameters; heterogeneous architectures need additional correspondence. Calibration scales fixed source residuals using a small set of target inputs. Evaluation covers one model lineage, two simulation benchmarks, and four tabletop tasks. Larger-scale validation is needed.

Future work. Future work could explore additional model lineages, source selection based on parameter and representation differences, and representative sampling with incremental calibration. Broader robot platforms, long-horizon tasks, and observation conditions would help assess generalization and initialization cost.

7 Conclusion

We introduced PreMerge for VLA, which constructs merged initialization before target-task action training by reusing VLMs with a shared base and alignable parameters. Parameter and representation analyses identified module-specific source updates, local sign conflicts, and task-dependent merging deviations. Motivated by these findings, the method combines adaptive weight merging and few-shot calibration without action annotations to integrate models from multiple stages into one VLM. PreMerge achieves average success rates of 98.9% on LIBERO and 92.7% on RoboTwin 2.0, exceeding the best single source by 1.0 and 3.2 percentage points. It averages 90.0% on four real-world tasks, 7.5 percentage points above the highest mean among the evaluated existing methods. Ablations further support the effectiveness of the evaluated component combination. After parameter folding, action training and inference use a single merged VLM, providing a way to improve VLA initialization by reusing existing models.

AI USE STATEMENT

Generative AI tools assisted with English translation, language editing, figure preparation and terminology consistency, LaTeX formatting, and checks of numerical and cross-reference consistency. The authors are responsible for the final manuscript and its claims.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A Unified Latent Action World Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bi_Motus_A_Unified_Latent_Action_World_Model_CVPR_2026_paper.html.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, Yanan Lu, Qi Lv, Haoxiang Ma, Jiangmiao Pang, Yu Qiao, Zherui Qiu, Yanqing Shen, Xu Shi, Yang Tian, Bolun Wang, Hanqing Wang, Jiaheng Wang, Tai Wang, Xueyuan Wei, Chao Wu, Yiman Xie, Boyang Xing, Yuqiang Yang, Yuyin Yang, Qiaojun Yu, Feng Yuan, Jia Zeng, Jingjing Zhang, Shenghan Zhang, Shi Zhang, Zhuoma Zhaxi, Bowen Zhou, Yuanzhen Zhou, Yunsong Zhou, Hongrui Zhu, Yangkun Zhu, and Yuchen Zhu. InternVLA-A1: Unifying Understanding, Generation and Action for Robotic Manipulation. *arXiv preprint arXiv:2601.02456*, 2026. URL https://arxiv.org/abs/2601.02456.

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

Yixiang Chen, Peiyan Li, Yuan Xu, Qisen Ma, Jiabing Yang, Kai Wang, Jianhua Yang, Dong An, He Guan, Gaoteng Liu, Jianlou Si, Jun Huang, Jing Liu, Nianfeng Liu, Yan Huang, and Liang Wang. FlowWAM: Optical Flow as a Unified Action Representation for World Action Models. *arXiv preprint arXiv:2607.13017*, 2026. URL https://arxiv.org/abs/2607.13017.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026. URL https://arxiv.org/abs/2605.02881.

Yicheng Feng, Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Sipeng Zheng, and Zongqing Lu. Spatial-aware VLA pretraining through visual-physical alignment from human videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 712–723, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Feng_Spatial-Aware_VLA_Pretraining_through_Visual-Physical_Alignment_from_Human_Videos_CVPR_2026_paper.html.

Yuxia Fu, Zhizhen Zhang, Yuqi Zhang, Zijian Wang, Zi Huang, and Yadan Luo. MergeVLA: Cross-skill model merging toward a generalist vision-language-action agent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22335–22347, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fu_MergeVLA_Cross-Skill_Model_Merging_Toward_a_Generalist_Vision-Language-Action_Agent_CVPR_2026_paper.html.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, 2025a. doi: 10.15607/RSS.2025.XXI.017. URL https://www.roboticsproceedings.org/rss21/p017.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. RoboMonkey: Scaling test-time sampling and verification for vision-language-action models. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3200–3217. PMLR, 2025. URL https://proceedings.mlr.press/v305/kwok25a.html.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. In *Robotics: Science and Systems XXII*, 2026.

doi: 10.15607/RSS.2026.XXII.016. URL https://www.roboticsproceedings.org/rss22/p016.html.

Weifeng Lin, Siyuan Huang, Hao Li, Tingwei Chen, Ruichuan An, Xinyu Wei, Jianbo Liu, and Hongsheng Li. Rethinking VLM representation for VLA initialization. *arXiv preprint arXiv:2605.25802*, 2026a. URL https://arxiv.org/abs/2605.25802.

Xuewu Lin, Tianwei Lin, Yun Du, Hongyu Xie, Yiwei Jin, Jiawei Li, Shijie Wu, Qingze Wang, Mengdi Li, Mengao Zhao, Ziang Li, Chaodong Huang, Hongzhe Bi, Lichao Huang, and Zhizhong Su. HoloBrain-0 Technical Report. *arXiv preprint arXiv:2602.12062*, 2026b. URL https://arxiv.org/abs/2602.12062.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yAzN4tz7oI.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

NVIDIA. Cosmos-Reason2-2B. Hugging Face model card, 2026a. URL https://huggingface.co/nvidia/Cosmos-Reason2-2B. Accessed 2026-09-23.

NVIDIA. GR00T-N1.7-3B. Hugging Face model card, 2026b. URL https://huggingface.co/nvidia/GR00T-N1.7-3B. Accessed 2026-09-23.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.011. URL https://www.roboticsproceedings.org/rss21/p011.html.

Qwen Team. Qwen3-VL-2B-Instruct. Hugging Face model card, 2025. URL https://huggingface.co/Qwen/Qwen3-VL-2B-Instruct. Accessed 2026-09-23.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-Cognitive Memory in Vision-Language-Action Models for Robotic Manipulation. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10011504.

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

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive Post-Training for Vision-Language-Action Models. *arXiv preprint arXiv:2505.17016*, 2025. URL https://arxiv.org/abs/2505.17016.

Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. FuseChat: Knowledge fusion of chat models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21618–21642. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1096. URL https://aclanthology.org/2025.emnlp-main.1096/.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, Siteng Huang, Yifan Tang, Wenhui Wang, Ru Zhang, Jianyi Liu, and Donglin Wang. VLA-Adapter: An effective paradigm for tiny-scale vision-language-action model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18638–18646, 2026. doi: 10.1609/aaai.v40i22.38931. URL https://ojs.aaai.org/index.php/AAAI/article/view/38931.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A Pragmatic VLA Foundation Model. *arXiv preprint arXiv:2601.18692*, 2026. URL https://arxiv.org/abs/2601.18692.

Jing Xu, Jiazheng Li, and Jingzhao Zhang. Scalable model merging with progressive layer-wise distillation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69451–69477. PMLR, 2025. URL https://proceedings.mlr.press/v267/xu25r.html.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

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

Senqiao Yang, Chengyao Wang, Yuxin Chen, Zixuan Wang, Longxiang Tang, Haokun Gui, Jinhui Ye, Changsheng Lu, Xiaoyang Wu, Mingkang Zhu, Pengguang Chen, Shu Liu, Zhuotao Tian, Hengshuang Zhao, Bei Yu, and Jiaya Jia. Beyond Data Scaling: Representation-Centric Continued Pre-training for Vision-Language-Action Models. *arXiv preprint arXiv:2608.27550*, 2026a. URL https://arxiv.org/abs/2608.27550.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-M0: VLA Foundation Model for Robotic Manipulation with Action Manifold Learning. *arXiv preprint arXiv:2602.11236*, 2026b. URL https://arxiv.org/abs/2602.11236.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Extend model merging from fine-tuned to pre-trained large language models via weight disentanglement. *arXiv preprint arXiv:2408.03092*, 2024. URL https://arxiv.org/abs/2408.03092.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Jianke Zhang, Xiaoyu Chen, Yanjiang Guo, Yucheng Hu, and Jianyu Chen. VLM4VLA: Revisiting vision-language-models in vision-language-action models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tc2UsBeODW.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007740.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Si Liu, and Guanghui Ren. ACoT-VLA: Action Chain-of-Thought for Vision-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8152–8162, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhong_ACoT-VLA_Action_Chain-of-Thought_for_Vision-Language-Action_Models_CVPR_2026_paper.html.

Zhanpeng Zhou, Zijun Chen, Yilan Chen, Bo Zhang, and Junchi Yan. On the emergence of cross-task linearity in pretraining-finetuning paradigm. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61854–61884. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhou24e.html.

A Experimental Protocols and Implementation

A.1 Experimental Setup

Datasets and tasks We conducted comparisons on two simulation benchmarks, LIBERO (Liu et al., 2023) and RoboTwin 2.0 (Chen et al., 2025), and evaluated real-world success rates on four manipulation tasks, alongside independent representation diagnostics and calibration-coefficient analyses. LIBERO includes the selected Spatial, Object, Goal, and Long suites, totaling 40 tasks with 50 demonstrations per task. RoboTwin 2.0 covers 50 bimanual manipulation tasks; the common mixed-training protocol uses 50 Clean and 500 Randomized demonstrations per task. Independent multitask policies were trained for the two benchmarks. Published-result protocols and sources are distinguished in **Appendix A.2**.

Models and inputs The sources were Qwen3-VL-2B-Instruct, Cosmos-Reason2-2B, and GR00T-N1.7-3B with its original action head removed (Bai et al., 2025; NVIDIA, 2026a;b). All retained the complete vision encoder, projector, and first 16 language blocks, and were connected to a GR00T action head (NVIDIA et al., 2025) in the main experiments. Inputs comprised task descriptions and images from two cameras for LIBERO or three for RoboTwin, all at 224×224 resolution. Initialization comparisons held training data, action-side initialization, and training budgets fixed, varying only VLM initialization.

Calibration Calibration samples were drawn from target-task training sets with balanced task sampling and contained only images and text, without action annotations. Source weights remained frozen. AdamW (Loshchilov & Hutter, 2019) optimized only the 986 merging coefficients at learning rate 10^{-3} for 20,000 steps, after which the result was folded into a single VLM.

Compute resources and action training All experiments ran on a server with 8 NVIDIA GeForce RTX 5090 GPUs. Action learning jointly updated the VLM and action-side parameters at learning rates of 10^{-5} and 10^{-4}, respectively, using AdamW with cosine scheduling. LIBERO training ran for 50,000 steps with an effective batch size of 64, and RoboTwin training ran for 500,000 steps with an effective batch size of 64.

Evaluation protocol LIBERO evaluation used 50 episodes per task, reporting success rates for the four suites and their mean. RoboTwin evaluation used 100 episodes per task in each of the Clean and Randomized domains, reporting the domain means over all 50 tasks and their average. Gains over the best single source and gains over Equal Weight use distinct reference initializations, both measured in percentage points. Sections 4.1–4.3 describe alignment, calibration, and action interfaces; Sections 5.2 and 5.3 report downstream success rates.

Reporting conventions All result tables report success percentages; Avg. is the unweighted mean of the listed suites, domains, or tasks. Bold scores mark column maxima in the simulation tables, including ties. Main-result increments compare with Equal Weight; ablation increments compare with full PreMerge. Red denotes increases and green decreases, in percentage points for success rates and coefficient units for source allocations. **Appendix A.2** records published-result sources and versions. Module coefficients are weighted by parameter coordinate counts; values and changes from initialization are rounded independently. Calibration plots use coefficient units.

A.2 Published-Result Sources

LIBERO (Table 2). Published results use their respective pretraining, input, and evaluation protocols. The π_0 results are from Table 1 of RIPT-VLA (Tan et al., 2025). Conference entries use the conference year, and preprints use the release year of the cited version. LingBot-VA uses its v2 preprint results and was formally published as CauVA. Bold scores denote the best result in each column, including all ties; published methods retain their originally reported scores.

RoboTwin (Table 3). Published results use Data Scaling Settings: 50 Clean and 500 Randomized demonstrations per task across all 50 tasks. Pretraining, inputs, and optimization settings still differ. Results for π_0, $\pi_{0.5}$, InternVLA-A1, and VLAct are from the Data Scaling columns of Table 2

of VLAct (Yang et al., 2026a); the OFT-head result is retained for VLAct. Conference entries use the conference year, and preprints use the release year of the cited version. LingBot-VA uses its v2 preprint results and was formally published as CauVA. Motus uses v2 results. When the original source does not report a two-domain mean, we compute the arithmetic mean of Clean and Randomized. Bold scores denote the best result in each column, including all ties; published methods retain their originally reported scores.

A.3 Real-World Experiments

Tasks and observations Real-world experiments used a tabletop robot-arm platform with four tasks: block placement, replacement, sorting, and stacking (Figure 2). The tasks require picking up three blocks and placing them in a box, replacing a red block with a blue block, arranging blocks from left to right in red–green–blue order, and stacking three blocks into a tower. Together they cover grasping and placement, color discrimination, spatial ordering, and stacking. Figure 2 presents these tasks in panels (a)–(d), with manipulation sequences read from left to right.

Calibration and diagnostic sampling Calibration used seed 42 and two demonstrations per task, totaling eight demonstrations, with 256 calibration frames and 80 nonoverlapping validation frames. The 80 held-out frames are split into 40 reference frames and 40 audit frames (10 per task in each split). Validation uses held-out frames from the same demonstrations as calibration. The 64 diagnostic demonstrations were sampled separately, excluding these eight.

Evaluation and baselines Each method was evaluated in 20 real-world trials per task, totaling 80 trials, with success rates based on completion of the entire task. With 20 trials per task, one trial changes the task success rate by five percentage points.

Real-world comparisons include RDT-1B, X-VLA, VLA-Adapter, π_0, $\pi_{0.5}$, and OpenVLA-OFT. Each method was evaluated on the same platform and four tasks using 100 demonstrations per task. Table 4 summarizes existing methods and different VLM initializations with the GR00T action head.

Task-level results PreMerge averages 90.0% success across the four real-world tasks, compared with 81.3% for the best single-source initialization, GR00T-N1.7, and 80.0% for equal-weight initialization. Relative to $\pi_{0.5}$, the existing method with the highest mean, PreMerge improves by 7.5 percentage points: gains are 15.0 percentage points each on replacement and sorting, while placement and stacking are tied. PreMerge achieves the highest or tied-highest success rate on every task, showing that merged initialization also benefits the evaluated tabletop real-world tasks.

A.4 Detailed Method Diagrams

The core method equations are provided in Sections 4.1–4.3; implementation settings and regularizer definitions are given in **Appendices A.8 and A.9**. The diagrams below illustrate the full pipeline and its individual stages.

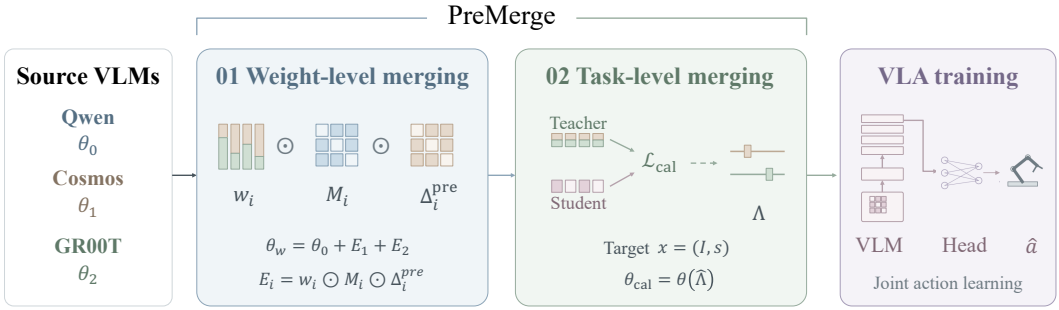


Figure A.1: Stages of PreMerge. Weight-level merging constructs fixed residuals from column weights, sign masks, and source updates to obtain the initial merged model. Task-level calibration and merging uses teacher and student features on target images and instructions to optimize only the merging coefficients Λ. The calibrated VLM is then used for target-task VLA training. Light blue, green, and purple denote these three components.

A.5 Weight-Level Merging Diagram

Section 4.1 defines source scoring, coordinatewise sign erasure, and the initial merged parameters in Equations (7)–(10).

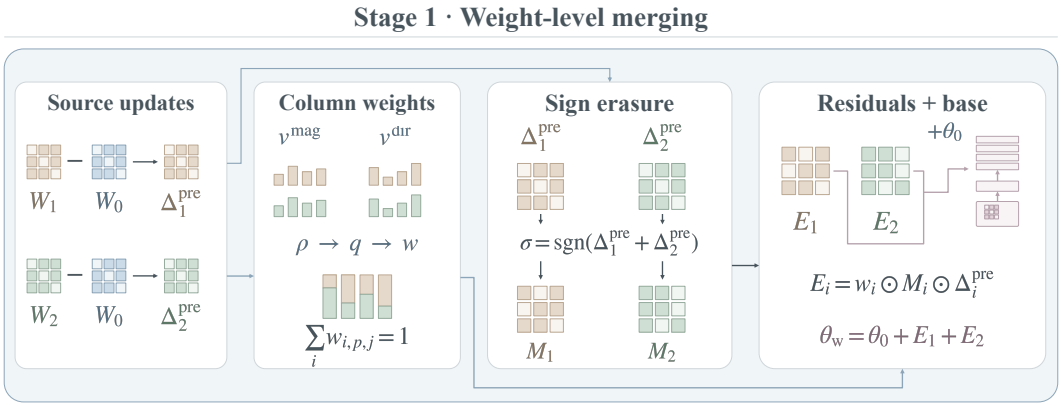


Figure A.2: Weight-level merging. Changes in original weight norms and directions relative to the base determine column contribution scores; the sum of unweighted source updates provides the reference signs. Column weights, sign masks, and updates form fixed residuals $E_{i,p}$, which are added to the base to obtain $\theta_{\text{w},p}$. Weights are not renormalized after erasure. Shared tensor or coordinate indices are omitted.

A.6 Task-Level Calibration Diagram

Section 4.2 provides the calibrated parameterization, fixed teacher targets, complete distillation loss, and joint objective in Equations (11)–(14). The three regularizers are defined in **Appendix A.9**.

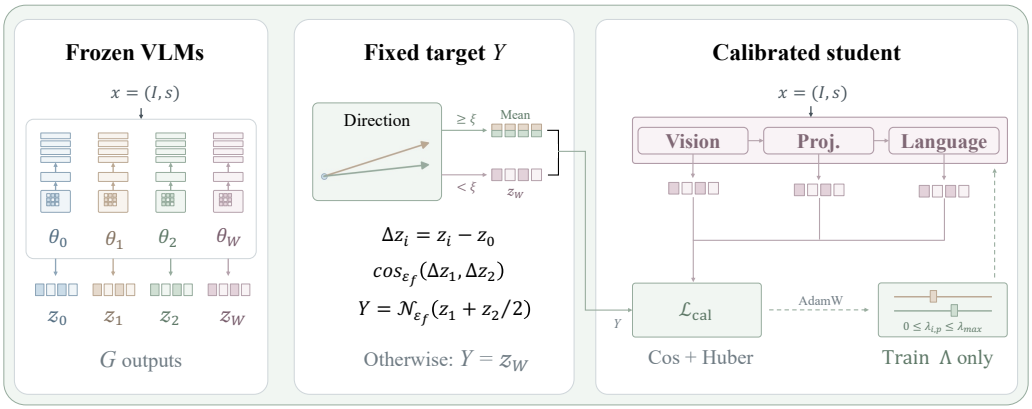


Figure A.3: Task-level calibration and merging. Feature changes of frozen sources are measured relative to the base. Directionally consistent changes yield a normalized mean target; otherwise, the initial merged feature z_{w} is used. The student computes distillation losses at G complete outputs and updates only the merging coefficients Λ, while teacher targets, base parameters, and source residuals remain fixed. The joint objective is given in Equation (14); common output and position indices g, t are omitted from features.

A.7 Target-Task VLA Training Diagram

Section 4.3 defines joint action training, the three action interfaces, flow matching, and OFT regression in Equations (15)–(18).

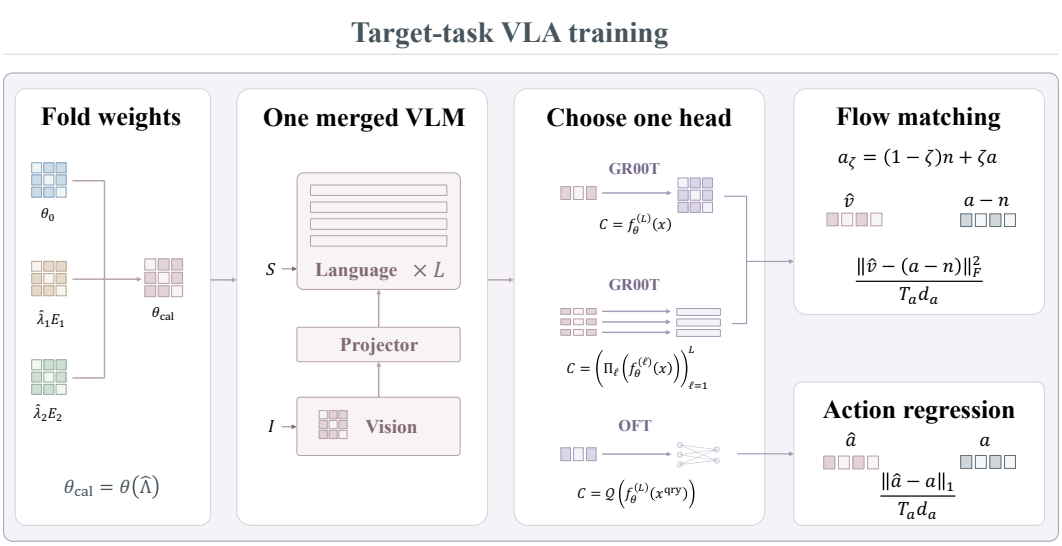


Figure A.4: Target-task VLA training. Calibrated residuals and the base are folded into one VLM, which is connected to one action interface. GR00T uses final-layer features, PI uses layerwise features, and OFT uses features at action-query positions. GR00T and PI use flow matching, while OFT uses L1 action regression. The corresponding losses jointly update VLM parameters θ and action-side parameters ϕ. The figure shows per-sample losses; sampling expectations and interfaces are defined in the text.

A.8 Implementation Settings

Initial merging used FP32; calibration used BF16 effective weights and FP32 coefficients. At export, BF16 parameter values composed using the final-step coefficients were stored in FP32 tensors, preserving the numerical weights used in calibration forward passes.

Scoring and tensor-shape rules are given in Section 4.1; numerical floors, feature normalization, and calibration optimization settings are in Section 4.2; parameter folding and action interfaces are in Section 4.3.

Alignment and calibration sampling Sources are aligned by parameter name, shape, and semantic position. The aligned VLM retains 24 vision blocks, the projector, token embeddings, the first 16 language blocks, and final normalization, with language width 2048; generation and action heads are excluded. The processor and tokenizer follow the shared base. WIDEN ranks here begin at zero, rather than at $1/c_p$ as in its original formulation. Teacher features can be extracted sequentially and cached.

Gradient updates, reference baselines, and diagnostics use $\mathcal{D}_{\text{cal}}$, $\mathcal{D}_{\text{ref}}$, and $\mathcal{D}_{\text{audit}}$, respectively, with disjoint sample identities. For LIBERO, one demonstration per task supplies calibration frames and another supplies reference and audit frames. For RoboTwin, the three frame sets come from one Randomized-domain demonstration per task. Real-world sampling is specified in **Appendix A.3**; its calibration audit frames are distinct from the separately sampled representation-diagnostic demonstrations. All frames use common camera order, preprocessing, and prompt templates.

A.9 Coefficient Regularization

Partition tensors into G nonempty structural groups $\mathcal{P}_g$ and define $\bar{\lambda}_{i,g} = |\mathcal{P}_g|^{-1} \sum_{p \in \mathcal{P}_g} \lambda_{i,p}$. For module $\eta \in \{\mathrm{V}, \mathrm{L}\}$ (vision or language), $\mathcal{E}_\eta$ contains adjacent group pairs (g, g'), each counted

once. The anchor, within-group, and adjacent-group penalties are

$$\mathcal{R}_{\text{anc}} = \frac{1}{|\mathcal{S}|P} \sum_{i \in \mathcal{S}} \sum_{p=1}^{P} (\lambda_{i,p} - 1)^2, \tag{19}$$

$$\mathcal{R}_{\text{grp}} = \frac{1}{|\mathcal{S}|G} \sum_{i \in \mathcal{S}} \sum_{g=1}^{G} \frac{1}{|\mathcal{P}_g|} \sum_{p \in \mathcal{P}_g} (\lambda_{i,p} - \bar{\lambda}_{i,g})^2, \tag{20}$$

$$\mathcal{R}_{\text{adj}} = \frac{1}{2|\mathcal{S}|} \sum_{\eta \in \{\text{V,L}\}} \frac{1}{|\mathcal{E}_\eta|} \sum_{i \in \mathcal{S}} \sum_{(g,g') \in \mathcal{E}_\eta} (\bar{\lambda}_{i,g} - \bar{\lambda}_{i,g'})^2. \tag{21}$$

The vision sequence includes the input projection and projector; the language sequence includes embeddings and final normalization. All three regularization weights equal 0.01 in Equation (14).

B Complete Parameter and Representation Analyses

This appendix expands Section 3: parameter changes and conflicts, representation linearity and similarity, and task-conditioned diagnostics are reported under common definitions. Downstream evaluation covers all 40 LIBERO and 50 RoboTwin tasks; representation diagnostics use four-task subsets.

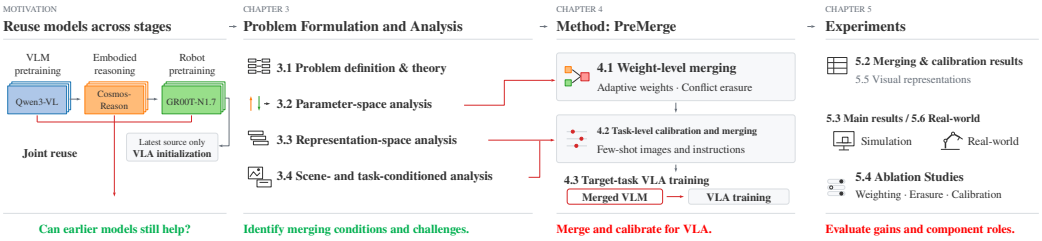


Figure B.1: Research overview of PreMerge. Starting from the motivation to reuse models across training stages, we analyze parameters, representations, and scene- and task-dependent behavior to identify merging conditions and challenges. This analysis motivates weight-level merging and task-level calibration, whose benefits and component contributions are evaluated experimentally.

Figure B.1 outlines the progression from model reuse to analysis, method design, and experimental validation. This section analyzes parameter composition in VLMs with a shared base, focusing on differences in source updates, representation deviations, and their task dependence to inform adaptive merging and target-task calibration.

B.1 Problem Definition and Theoretical Background

B.1.1 Shared Base and Parameter Composition

A shared base provides a common reference for comparing training-induced changes across sources. Let θ denote the aligned VLM parameters, θ_0 the shared base parameters, and θ_i the parameters of source i, where $i \in \{0, \ldots, K-1\}$ and K is the number of sources including the base itself. Following parameter arithmetic (Ilharco et al., 2023), let $\mathcal{T}_i$ denote the cumulative subsequent training of source i and Δ_i^{pre} its corresponding parameter update:

$$\theta_i = \mathcal{T}_i(\theta_0), \qquad \Delta_i^{\text{pre}} = \theta_i - \theta_0, \qquad \theta_i = \theta_0 + \Delta_i^{\text{pre}}. \tag{22}$$

We use $K = 3$, with sources 0, 1, and 2 corresponding to the vision-language components of Qwen3-VL-2B-Instruct, Cosmos-Reason2-2B, and GR00T-N1.7-3B, respectively. Cosmos-Reason2 is post-trained from a Qwen model, and the GR00T VLM backbone is inherited from Cosmos-Reason2. We therefore use Qwen as the shared base, indexed by 0, with zero update $\Delta_0^{\text{pre}} = \mathbf{0}$. Architecture details are available on the model release pages (NVIDIA, 2026a;b).

Simple parameter merging constructs a model by weighting source updates. Let $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_{K-1}) \in \mathbb{R}^K$ be a vector of global coefficients, where α_i is the weight of source i. The

merged parameters $\theta_{\boldsymbol{\alpha}}$ are

$$\theta_{\boldsymbol{\alpha}} = \theta_0 + \sum_{i=0}^{K-1} \alpha_i \Delta_i^{\mathrm{pre}} = \sum_{i=0}^{K-1} \alpha_i \theta_i, \qquad \alpha_i \geq 0, \quad \sum_{i=0}^{K-1} \alpha_i = 1. \tag{23}$$

This section uses the equal-weight combination of three sources, $\alpha_i = 1/3$ (*Equal mean*), and the three pairwise midpoints as comparisons. Each pairwise midpoint assigns weight $1/2$ to the two selected sources and zero to the remaining source.

B.1.2 Parameter and Representation Interpolation

The pairwise CTL relation and its multisource and base-relative forms are defined in Equations (3) and (4). We evaluate deviations using the pooled sample representations and CTL error in Section 3.3.

B.1.3 Unified Analysis Setup

The analysis covers the vision, projector, and language modules. The three sources were aligned by parameter keys and shapes, retaining the complete vision encoder and the first 16 language blocks. **Table B.1** specifies the parameter scopes and representation extraction points.

For each of LIBERO, RoboTwin, and the real-world dataset, the representation analysis used 64 independent demonstrations from four tasks, with 16 demonstrations per task and one image–original-instruction pair per demonstration. Statistics were computed separately for each dataset. The sampling scopes are detailed in the dataset-specific subsections.

The three source models and four combined models used identical preprocessing and FP32 forward passes. The three panels in each parameter analysis figure show module-level, vision-layer, and language-layer results, respectively. Nine-panel figures for the three datasets arrange LIBERO, RoboTwin, and real-world data by row, and module-level, vision-layer, and language-layer results by column. The captions specify the representation extraction points.

Table B.1: Parameter analysis scopes and sample-level representation boundaries. Module-level metrics are computed directly on the corresponding parameters or outputs; layerwise metrics use the pooled output of each block.

Module	**Parameter scope and coordinate count**	**Sample-level representation boundary**
Vision	Image patch projection, visual positional embeddings, and all vision blocks; 306,242,560	Mean of 196 image tokens after the final vision block and before the main projector; $d_\ell = 1024$
Projector	Main merger and 3 DeepStack mergers; 100,714,496	Mean of 49 tokens in each of four branches, concatenated in the order of the main branch and DeepStack 1–3; $d_\ell = 8192$
Language	First 16 language blocks; 805,376,000	After block 16, before the final RMSNorm, averaged over all valid image and text tokens; $d_\ell = 2048$

B.2 Parameter-Space Analysis

Figures B.2 and B.3 and **Table B.2** depend only on the same source weights. These results are shared across the three datasets and were computed once.

We first compare the magnitudes and directions of source updates and their cancellation under composition. Relative update magnitude is measured by $\|\Delta_i^{\mathrm{pre}}\|_2 / \|\theta_0\|_2$, where $\|\cdot\|_2$ is the Euclidean norm and parameters are flattened within the corresponding scope. Directional differences are characterized by the cosine similarity between nonzero updates and coordinatewise sign conflicts. A coordinate is conflicting when its two non-base updates have opposite signs. The conflict ratio uses all parameter coordinates, including those with zero updates, as its denominator. Coordinate energy is the equally weighted mean of squared updates from the three sources, and the conflict energy

share is the energy summed over conflicting coordinates divided by the energy over all coordinates. For the equal-weight combination of three sources, we further define the update retention ratio as

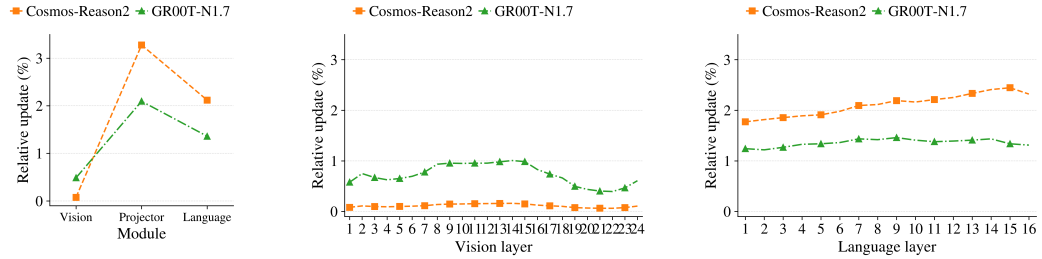
$$R = \frac{\left\| \sum_{i=0}^{K-1} \alpha_i \Delta_i^{\mathrm{pre}} \right\|_2}{\sum_{i=0}^{K-1} \alpha_i \left\| \Delta_i^{\mathrm{pre}} \right\|_2}, \qquad \alpha_i = 1/3.$$

When the denominator is positive, $R \in [0, 1]$. Smaller R indicates stronger cancellation, and the cancellation ratio is $1 - R$. **Table B.2** summarizes the module-level results.

Table B.2: Source parameter differences relative to the shared base and cancellation under equal-weight merging. Conflict energy is the share of update energy carried by conflicting coordinates.

Module	Cosmos relative update (%)	GR00T relative update (%)	Nonzero-update cosine	Conflicting coords. (%)	Conflict energy (%)	$1 - R$ (%)
Vision	0.08	0.49	8.13×10^{-3}	25.89	27.54	12.25
Projector	3.28	2.09	0.01	43.70	45.35	27.13
Language	2.12	1.36	0.03	39.30	41.76	26.63

The relative magnitudes of source updates vary across modules: GR00T has larger vision updates, whereas Cosmos has larger projector and language updates. The cosine similarities between the two nonzero updates are close to zero, indicating approximately orthogonal overall directions (**Table B.2**). Update magnitudes also vary with layer position. For example, in vision layer 14, the relative updates of Cosmos and GR00T are 0.16% and 1.01%, respectively (**Figure B.2(b)**). **Panels (a), (b), and (c)** show module-level, vision-layer, and language-layer results, respectively.



(a) Module-level relative updates (b) Vision-layer relative updates (c) Language-layer relative updates

Figure B.2: Source update magnitudes relative to the shared base at the module, vision-layer, and language-layer levels.

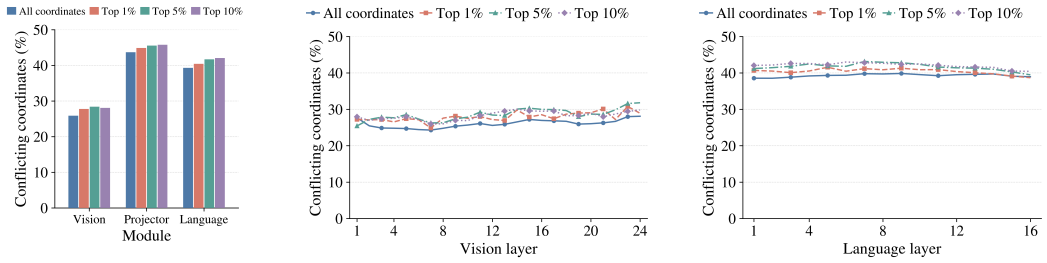
Sign conflicts are more common in the projector and language modules. To examine conflicts among larger updates, we selected the top 1%, 5%, and 10% of coordinates within each tensor by the equally weighted sum of squared updates. In the top 1% subset, the conflict ratios for Vision, Projector, and Language are 27.77%, 44.88%, and 40.44%, respectively. These are close to the full-coordinate statistics, showing that sign disagreements also occur among larger updates (**Figure B.3(a)**). **Panels (a), (b), and (c)** show module-level, vision-layer, and language-layer results, respectively.

Under equal-weight merging, the cancellation ratios in Projector and Language are 27.13% and 26.63%, exceeding the 12.25% in Vision. Cancellation depends on both direction and relative magnitude: approximately orthogonal updates can still have a low cancellation ratio when one source dominates in magnitude. These differences across modules and layers motivate adapting source weights by position and handling local conflicts.

Layerwise cancellation ratios range from 11.10% to 13.31% in vision and from 25.14% to 27.39% in language, consistent with the module-level differences.

B.3 Representation-Space Analysis

All diagnostics below used 224×224 inputs, FP32 forward passes, and the pooling boundaries in **Table B.1**. The coefficients were $1/3$ for equal-weight three-source merging and $1/2$ for pairwise



(a) Module-level sign conflicts (b) Vision-layer sign conflicts (c) Language-layer sign conflicts

Figure B.3: Coordinatewise sign conflicts between source parameter updates at the module, vision-layer, and language-layer levels.

midpoints. For each sample, let the actual merged representation be $u = h_\ell(x; \theta_{\boldsymbol{\alpha}})$, the interpolation of source representations be $v = \sum_{i=0}^{K-1} \alpha_i h_\ell(x; \theta_i)$, and the weighted sum of source norms be $D = \sum_{i=0}^{K-1} \alpha_i \|h_\ell(x; \theta_i)\|_2$. The normalized CTL error (Zhou et al., 2024), identical to Equation (5), is

$$\mathrm{Err}_{\mathrm{CTL}}(x; \boldsymbol{\alpha}, \ell) = \frac{\|u - v\|_2}{D}. \tag{24}$$

The tables report the mean and sample standard deviation of the per-sample errors multiplied by 100; the standard deviation describes variation across input samples. CKA uses the same centered linear-kernel similarity (Kornblith et al., 2019) as Equation (6). Let each row of H_A, H_B represent the same sample in the two feature spaces, and let $\widetilde{H}_A, \widetilde{H}_B$ be the matrices centered along the sample dimension. Then

$$\mathrm{CKA}(H_A, H_B) = \frac{\|\widetilde{H}_A^\top \widetilde{H}_B\|_F^2}{\|\widetilde{H}_A^\top \widetilde{H}_A\|_F \|\widetilde{H}_B^\top \widetilde{H}_B\|_F}. \tag{25}$$

Here, $\|\cdot\|_F$ is the Frobenius norm. Overall and task-group statistics use independent centering within their respective sample sets. Each dataset was analyzed separately, with the following sampling scopes.

LIBERO sampling and measurement protocol

The main LIBERO (Liu et al., 2023) evaluation covers 40 tasks across Spatial, Object, Goal, and Long. Representation diagnostics used four tasks, with 16 independent demonstrations per task and one image–instruction sample per demonstration, yielding 64 frames.

RoboTwin sampling and measurement protocol

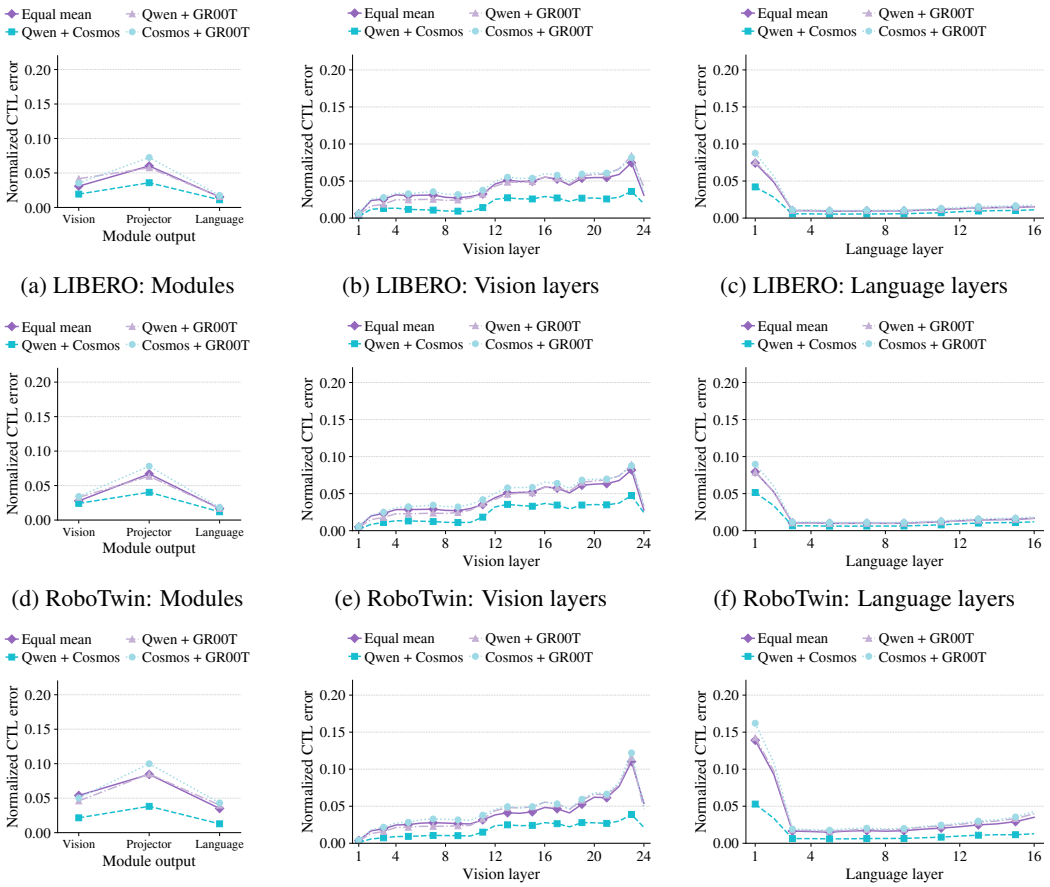
The main RoboTwin (Chen et al., 2025) evaluation covers all 50 tasks in both Clean and Randomized domains. For representation diagnostics, one middle frame was sampled from each of 16 independent training demonstrations for each of four tasks in the Randomized domain, yielding 64 frames.

Real-world sampling and measurement protocol

The four tasks each contain 100 demonstrations, totaling 400. For each task, we excluded the two demonstrations used for calibration, randomly selected 16 demonstrations from the remainder, and took the middle frame of each, yielding 64 independent demonstrations. The inputs, FP32 forward passes, pooling, and CTL/CKA definitions follow the diagnostic protocol above. Model inputs were preprocessed to 224×224 under the same protocol.

B.3.1 CTL Error

Figure B.4 shows module-level and layerwise CTL errors on the three datasets, and **Table B.3** summarizes the module-level statistics. Projector consistently has the largest CTL error under equal-weight three-source merging: 6.02%, 6.70%, and 8.45% on LIBERO, RoboTwin, and real-world data, respectively. The corresponding Vision errors are 3.09%, 2.78%, and 5.39%, and Language errors are 1.53%, 1.67%, and 3.50%. Real-world errors are higher than those on both simulation



(a) LIBERO: Modules (b) LIBERO: Vision layers (c) LIBERO: Language layers

(d) RoboTwin: Modules (e) RoboTwin: Vision layers (f) RoboTwin: Language layers

(g) Real-world: Modules (h) Real-world: Vision layers (i) Real-world: Language layers

Figure B.4: Module-level and layerwise CTL errors on three datasets. Rows correspond to datasets; columns show module outputs, vision layers, and language layers. Curves show sample-mean errors for each merging combination. Module-level means and standard deviations are reported in **Table B.3**.

Table B.3: Module-level CTL errors on three datasets (%, mean ± sample standard deviation).

Dataset	Output	Three-source mean	Qwen–Cosmos	Qwen–GR00T	Cosmos–GR00T
LIBERO	Vision	3.09 ± 2.34	1.94 ± 1.53	4.17 ± 2.75	3.52 ± 3.00
	Projector	6.02 ± 0.51	3.58 ± 0.30	5.75 ± 0.82	7.24 ± 0.56
	Language	1.53 ± 0.13	1.11 ± 0.09	1.57 ± 0.20	1.75 ± 0.13
RoboTwin	Vision	2.78 ± 2.30	2.42 ± 1.75	3.29 ± 2.79	3.43 ± 2.69
	Projector	6.70 ± 0.73	4.03 ± 0.55	6.35 ± 0.95	7.81 ± 0.83
	Language	1.67 ± 0.24	1.20 ± 0.12	1.67 ± 0.30	1.84 ± 0.34
Real-world	Vision	5.39 ± 3.27	2.15 ± 1.57	4.60 ± 3.41	5.04 ± 3.32
	Projector	8.45 ± 1.52	3.82 ± 0.32	8.51 ± 2.16	9.99 ± 1.90
	Language	3.50 ± 0.60	1.28 ± 0.10	3.99 ± 1.16	4.31 ± 0.82

datasets in all three modules, indicating larger representation deviations caused by parameter merging in real scenes. Among pairwise combinations, Qwen–Cosmos has the lowest error in every module on all three datasets.

B.3.2 Squared CTL Error Decomposition

Using u, v, D from **Equation (24)**, squared CTL error decomposes into magnitude and direction terms:

$$\mathrm{Err}^2_{\mathrm{CTL}} = \underbrace{\frac{(\|u\|_2 - \|v\|_2)^2}{D^2}}_{E_{\mathrm{mag}}} + \underbrace{\frac{2\|u\|_2\|v\|_2(1-\cos(u,v))}{D^2}}_{E_{\mathrm{dir}}}. \tag{26}$$

For each output point, the two components are summed separately over 64 samples and divided by the total squared error. The magnitude share is $\sum_x E_{\mathrm{mag}}(x) / \sum_x \mathrm{Err}_{\mathrm{CTL}}(x)^2$, and the direction share is computed analogously. The decomposition requires positive D and feature norms; the shares require nonzero total squared error. All analyzed samples satisfy these conditions. The shares describe error composition, while the CTL value measures error magnitude.

Figure B.5 shows the squared CTL error decomposition on the three datasets, and **Table B.4** summarizes the magnitude and direction shares for each module. In **Table B.4**, error at the final Vision output is dominated by the magnitude term, which accounts for 98.86%, 98.59%, and 99.08% on LIBERO, RoboTwin, and real-world data, respectively. Projector and Language errors are dominated by the direction term. The Projector direction share exceeds 95% on all three datasets, while the Language share decreases from 95.17% on LIBERO to 88.32% on RoboTwin and 80.90% on real-world data. Thus, error composition at final module outputs varies by module.

B.3.3 CKA Between Source Models

Figure B.6 shows module-level and layerwise CKA between source models on the three datasets; **Table B.5** summarizes module-level source-pair similarities. All three datasets exhibit clear differences across modules: source CKA is generally high in Projector and Language, whereas Vision similarity is lower and varies by dataset. For Qwen–GR00T, Vision CKA is 0.51, 0.08, and 0.39 on LIBERO, RoboTwin, and real-world data, respectively. The corresponding Projector values are 0.98, 0.93, and 0.97, and Language values are 0.99, 0.99, and 1.00. This indicates that the source models develop more similar cross-sample representation structures after the projector and language modules.

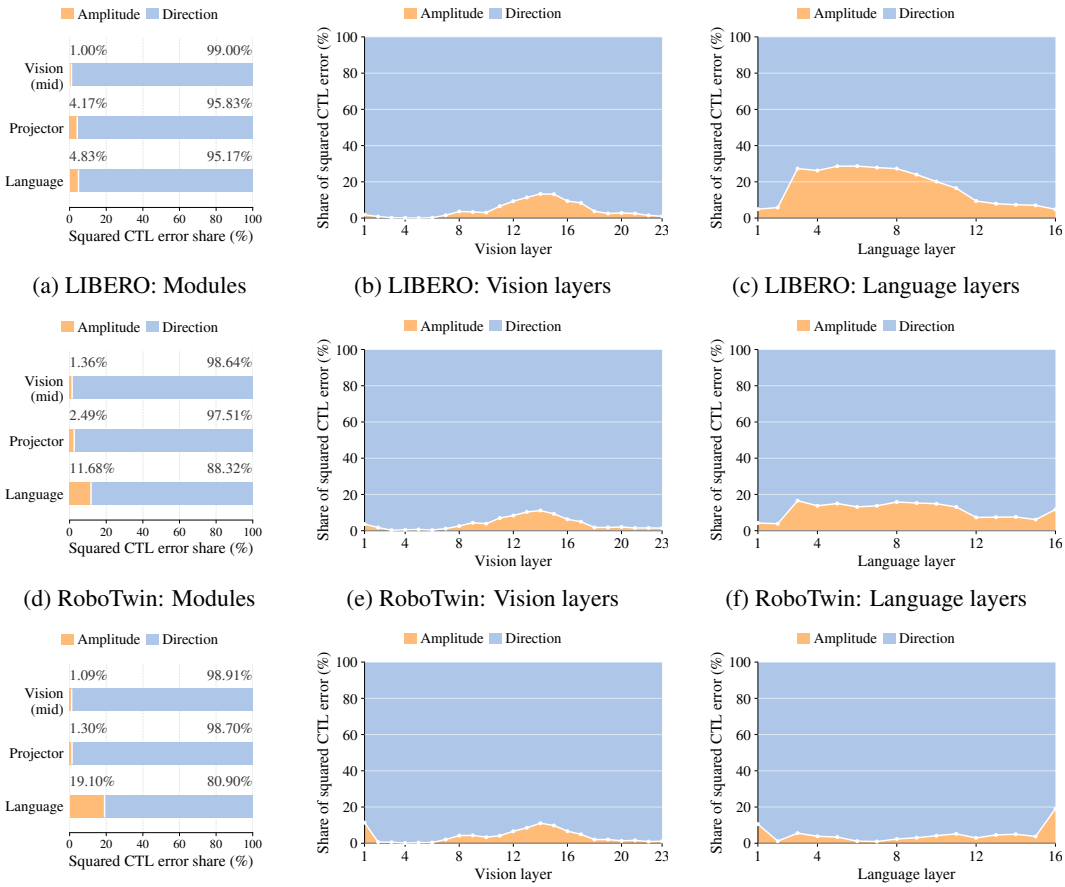
B.3.4 Internal CKA of the Equal-Weight Model

Figure B.7 shows cross-module and cross-layer CKA within the equal-weight model on the three datasets, and **Table B.6** summarizes the three output-pair similarities. Projector–Language CKA is the highest among the three module pairs on every dataset, reaching 0.99, 0.53, and 0.85 on LIBERO, RoboTwin, and real-world data, respectively. Pairs involving Vision have lower similarities. RoboTwin has the lowest CKA for all three module pairs, while LIBERO Projector and Language have nearly identical cross-sample structures. Real-world Projector–Language CKA is 0.85, between the two simulation datasets; its Vision–Projector and Vision–Language values are 0.65 and 0.71, both higher than the corresponding simulation values.

B.4 Scene- and Task-Conditioned Analysis

B.4.1 Task-Conditioned CTL

Figure B.8 shows CTL under different task conditions on the three datasets, and **Table B.7** summarizes module-level results by task group. CTL varies substantially with task conditions. Black bowl has the highest Projector CTL on LIBERO at 6.46%, while Hammering a block has the highest on RoboTwin at 7.02%. Among the real-world tasks, Block sorting has the highest CTL in Vision, Projector, and Language at 8.00%, 9.86%, and 4.01%, respectively; Block stacking has the lowest in all three modules at 3.70%, 7.17%, and 2.97%. These results show that scene and task combinations affect the extent of representation deviation caused by parameter merging.



(a) LIBERO: Modules (b) LIBERO: Vision layers (c) LIBERO: Language layers

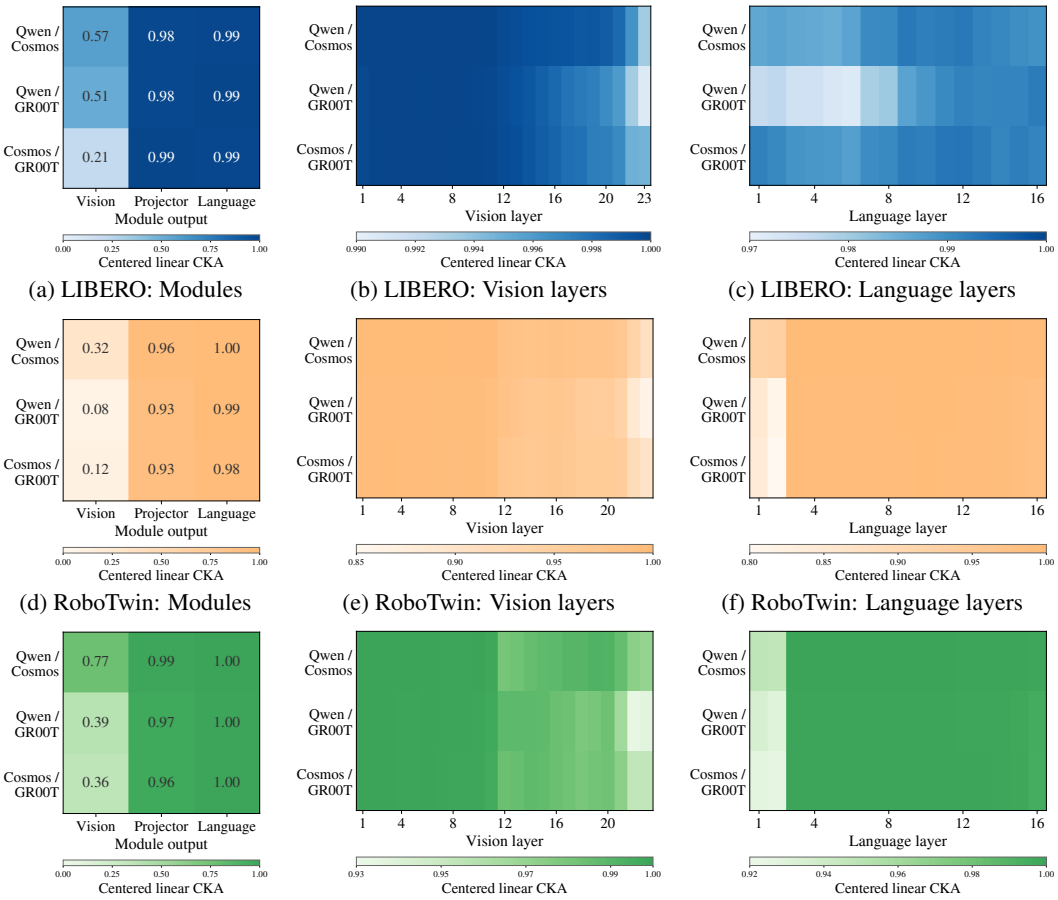
(d) RoboTwin: Modules (e) RoboTwin: Vision layers (f) RoboTwin: Language layers

(g) Real-world: Modules (h) Real-world: Vision layers (i) Real-world: Language layers

Figure B.5: Squared CTL error decomposition for equal-weight merging on three datasets. Each row corresponds to one dataset. The left column compares Vision, Projector, and Language; the middle and right columns show vision-layer and language-layer results, respectively.

Table B.4: Squared CTL error decomposition for equal-weight merging on three datasets.

Dataset	Output	Mean squared CTL	Magnitude share (%)	Direction share (%)
LIBERO	Vision	0.001495	98.86	1.14
	Projector	0.003650	4.17	95.83
	Language	0.000236	4.83	95.17
RoboTwin	Vision	0.001292	98.59	1.41
	Projector	0.004537	2.49	97.51
	Language	0.000285	11.68	88.32
Real-world	Vision	0.003957	99.08	0.92
	Projector	0.007366	1.30	98.70
	Language	0.001263	19.10	80.90



(a) LIBERO: Modules (b) LIBERO: Vision layers (c) LIBERO: Language layers

(d) RoboTwin: Modules (e) RoboTwin: Vision layers (f) RoboTwin: Language layers

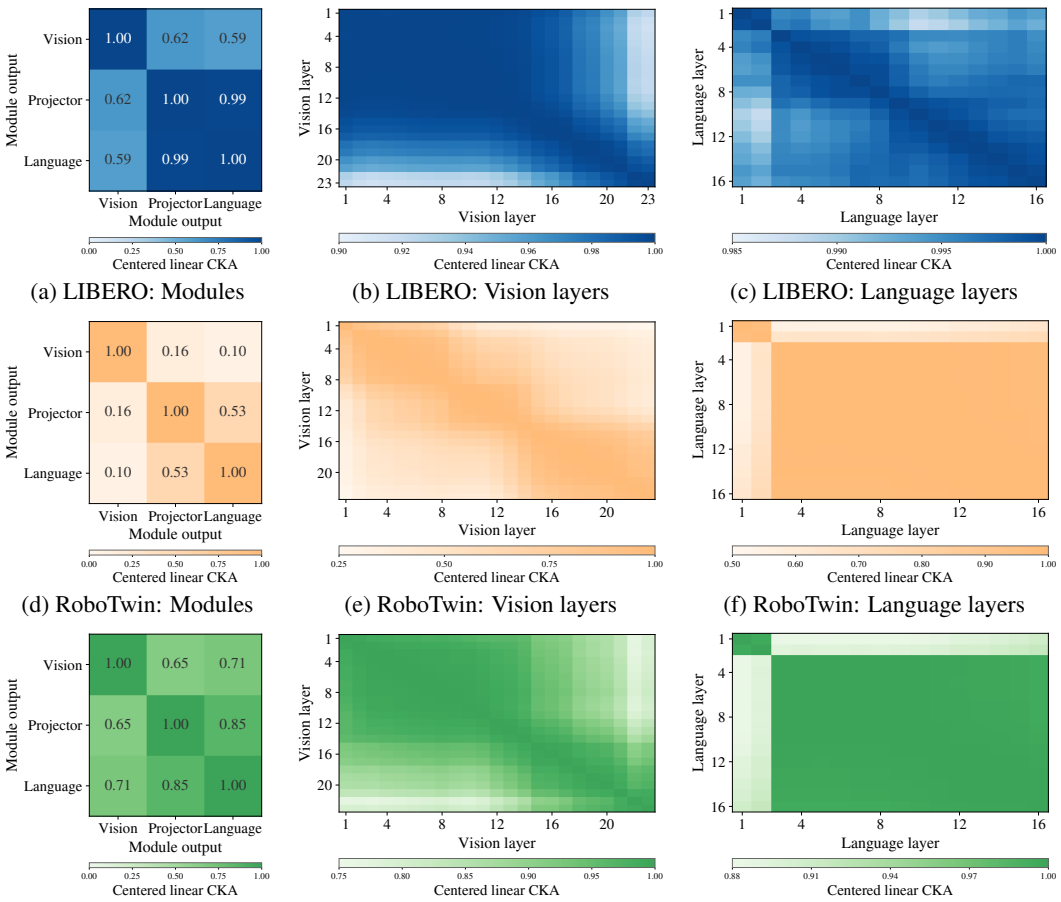
(g) Real-world: Modules (h) Real-world: Vision layers (i) Real-world: Language layers

Figure B.6: CKA between source models on three datasets, comparing module outputs and layerwise representations. Rows correspond to datasets; columns show modules, vision layers, and language layers. Rows within each panel correspond to source-model pairs. Blue, orange, and green denote LIBERO, RoboTwin, and real-world data, respectively. Each panel has a separately labeled color scale.

Table B.5: Module-level CKA between source models on three datasets.

Dataset	Vision			Projector			Language		
	Q–C	Q–G	C–G	Q–C	Q–G	C–G	Q–C	Q–G	C–G
LIBERO	0.57	0.51	0.21	0.98	0.98	0.99	0.99	0.99	0.99
RoboTwin	0.32	0.08	0.12	0.96	0.93	0.93	1.00	0.99	0.98
Real-world	0.77	0.39	0.36	0.99	0.97	0.96	1.00	1.00	1.00

Q: Qwen; C: Cosmos; G: GR00T.



(a) LIBERO: Modules (b) LIBERO: Vision layers (c) LIBERO: Language layers

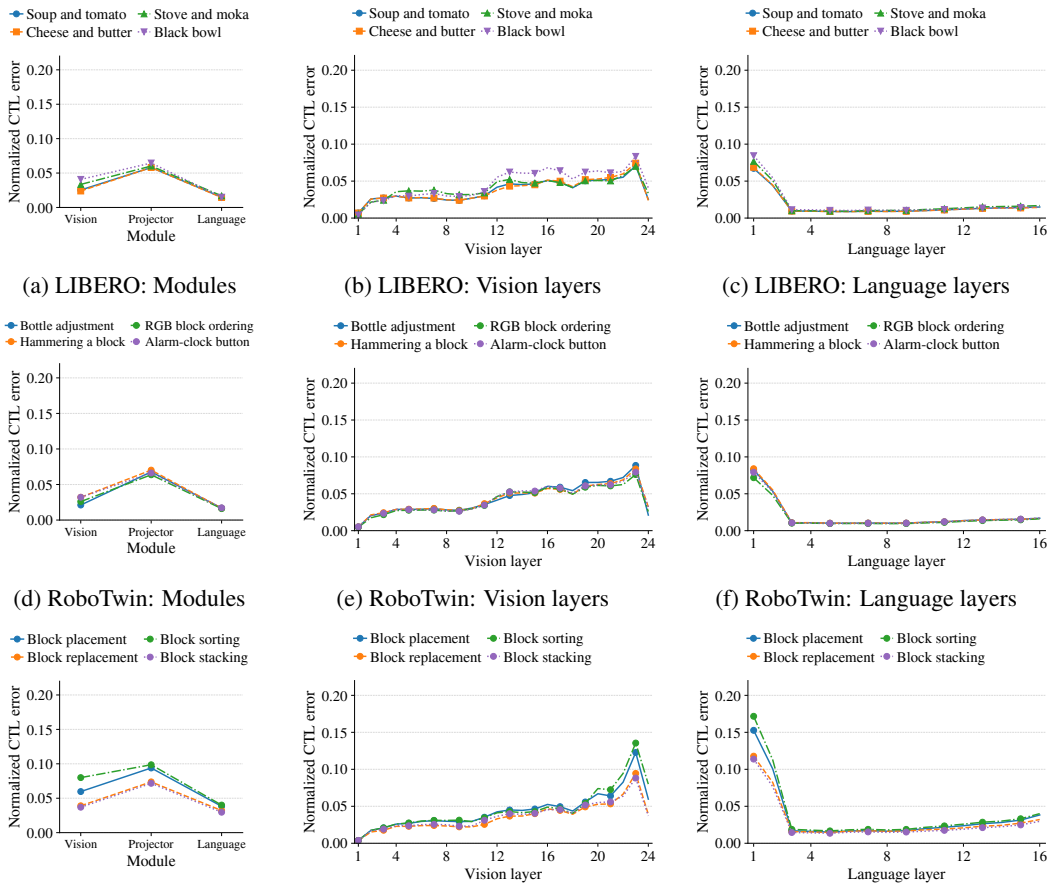
(d) RoboTwin: Modules (e) RoboTwin: Vision layers (f) RoboTwin: Language layers

(g) Real-world: Modules (h) Real-world: Vision layers (i) Real-world: Language layers

Figure B.7: Internal CKA of the equal-weight model on three datasets, comparing representations across modules and layers. Rows correspond to datasets; columns show CKA matrices across modules, vision layers, and language layers. Heatmaps retain diagonal self-comparisons of 1, while **Table B.6** reports only off-diagonal module pairs. Blue, orange, and green denote LIBERO, RoboTwin, and real-world data, respectively, with a separately labeled color scale for each panel.

Table B.6: Cross-module CKA within the equal-weight model on three datasets, excluding diagonal self-comparisons.

Dataset	Output pair	CKA
LIBERO	vision–projector	0.62
	vision–language	0.59
	projector–language	0.99
RoboTwin	vision–projector	0.16
	vision–language	0.10
	projector–language	0.53
Real-world	vision–projector	0.65
	vision–language	0.71
	projector–language	0.85



(a) LIBERO: Modules (b) LIBERO: Vision layers (c) LIBERO: Language layers

(d) RoboTwin: Modules (e) RoboTwin: Vision layers (f) RoboTwin: Language layers

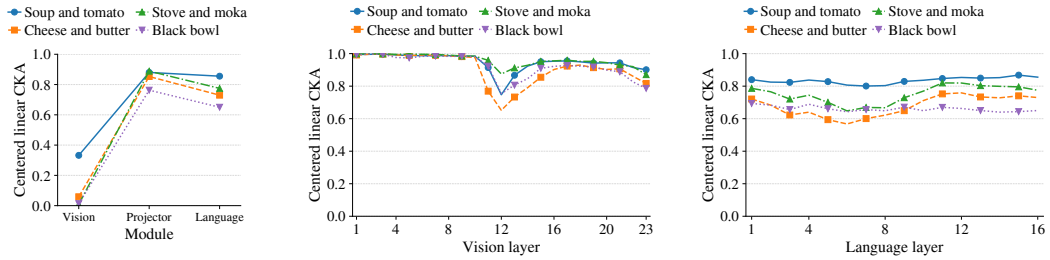
(g) Real-world: Modules (h) Real-world: Vision layers (i) Real-world: Language layers

Figure B.8: Task-conditioned CTL under equal-weight merging on three datasets, at module outputs, vision layers, and language layers. Each row corresponds to one dataset, with curves distinguishing its task groups. Values are within-task sample means; module-level means and standard deviations are reported in **Table B.7**.

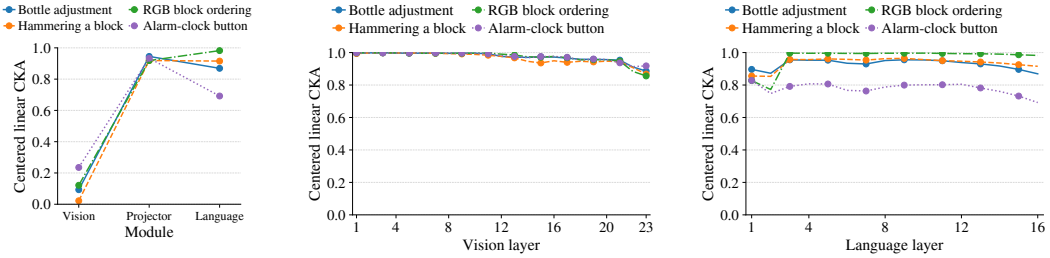
Table B.7: CTL under equal-weight merging by task group on three datasets (%, mean ± sample standard deviation).

Dataset	Task group	Samples	Vision	Projector	Language
LIBERO	Overall	64	3.09 ± 2.34	6.02 ± 0.51	1.53 ± 0.13
	Soup and tomato	16	2.54 ± 1.62	5.80 ± 0.37	1.49 ± 0.06
	Cheese and butter	16	2.39 ± 1.61	5.80 ± 0.23	1.44 ± 0.06
	Stove and moka	16	3.35 ± 2.70	6.02 ± 0.68	1.69 ± 0.13
	Black bowl	16	4.08 ± 2.94	6.46 ± 0.37	1.49 ± 0.08
RoboTwin	Overall	64	2.78 ± 2.30	6.70 ± 0.73	1.67 ± 0.24
	Bottle adjustment	16	2.11 ± 1.66	6.78 ± 0.59	1.69 ± 0.16
	Hammering a block	16	3.20 ± 2.59	7.02 ± 0.82	1.67 ± 0.15
	RGB block ordering	16	2.59 ± 1.60	6.37 ± 0.75	1.61 ± 0.28
	Alarm-clock button	16	3.21 ± 3.04	6.62 ± 0.64	1.72 ± 0.34
Real-world	Overall	64	5.39 ± 3.27	8.45 ± 1.52	3.50 ± 0.60
	Block placement	16	5.97 ± 2.83	9.39 ± 1.20	3.83 ± 0.25
	Block replacement	16	3.90 ± 2.48	7.37 ± 0.90	3.21 ± 0.51
	Block sorting	16	8.00 ± 2.45	9.86 ± 0.70	4.01 ± 0.30
	Block stacking	16	3.70 ± 3.41	7.17 ± 0.94	2.97 ± 0.57

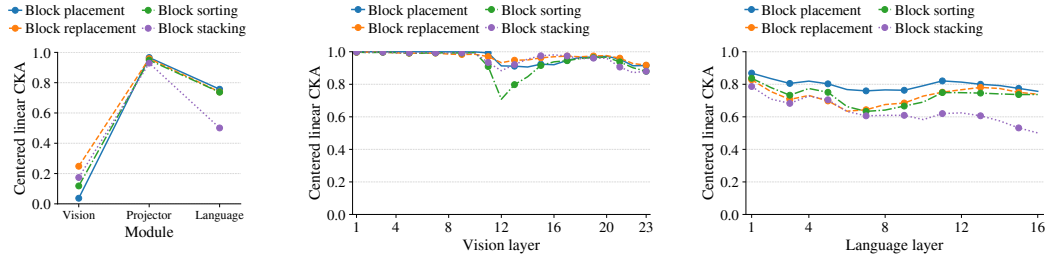
B.4.2 TASK-CONDITIONED CKA



(a) LIBERO: Modules (b) LIBERO: Vision layers (c) LIBERO: Language layers



(d) RoboTwin: Modules (e) RoboTwin: Vision layers (f) RoboTwin: Language layers



(g) Real-world: Modules (h) Real-world: Vision layers (i) Real-world: Language layers

Figure B.9: Task-conditioned Qwen–GR00T CKA on three datasets, at module outputs, vision layers, and language layers. Each row corresponds to one dataset, with curves distinguishing its task groups. CKA is computed with independent centering within each task group.

Table B.8: Qwen–GR00T CKA by task group on three datasets.

Dataset	Task group	Samples	Vision	Projector	Language
LIBERO	Overall	64	0.51	0.98	0.99
	Soup and tomato	16	0.33	0.88	0.86
	Cheese and butter	16	0.06	0.85	0.73
	Stove and moka	16	0.01	0.89	0.77
	Black bowl	16	0.01	0.76	0.65
RoboTwin	Overall	64	0.08	0.93	0.99
	Bottle adjustment	16	0.09	0.95	0.87
	Hammering a block	16	0.02	0.92	0.92
	RGB block ordering	16	0.12	0.92	0.98
	Alarm-clock button	16	0.24	0.93	0.69
Real-world	Overall	64	0.39	0.97	1.00
	Block placement	16	0.04	0.97	0.76
	Block replacement	16	0.25	0.96	0.74
	Block sorting	16	0.12	0.95	0.74
	Block stacking	16	0.17	0.93	0.50

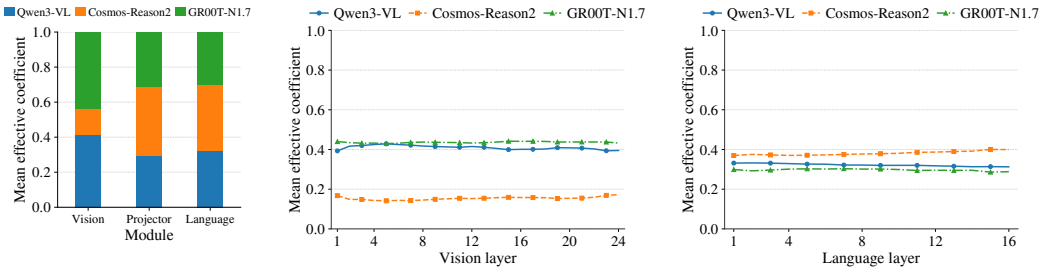
Figure B.9 shows Qwen–GR00T CKA under different task conditions on the three datasets, and **Table B.8** summarizes module-level similarities by task group. After grouping by task, Projector CKA remains generally high, while Vision and Language similarities vary more with the task. Task-

specific Language CKA ranges from 0.65–0.86 on LIBERO, 0.69–0.98 on RoboTwin, and 0.50–0.76 on real-world data. Vision CKA also varies markedly: it is 0.01 for Stove and moka on LIBERO, 0.02 for Hammering a block on RoboTwin, and 0.04 for real-world Block placement, indicating the weakest visual representation agreement under these specific task conditions.

C Complete Merging and Calibration Results

This appendix expands Section 5.6. Initial coefficients are shared across datasets because initialization uses only source parameters; calibrated coefficients and scale diagnostics are reported separately for LIBERO, RoboTwin, and the real-world dataset.

C.1 Initialization Before Calibration



(a) Initial module coefficients (b) Initial vision-layer coefficients (c) Initial language-layer coefficients

Figure C.1: Effective source coefficients before calibration, at the module, vision-layer, and language-layer levels. This initialization depends only on source weights, so all three datasets share the same initial coefficients.

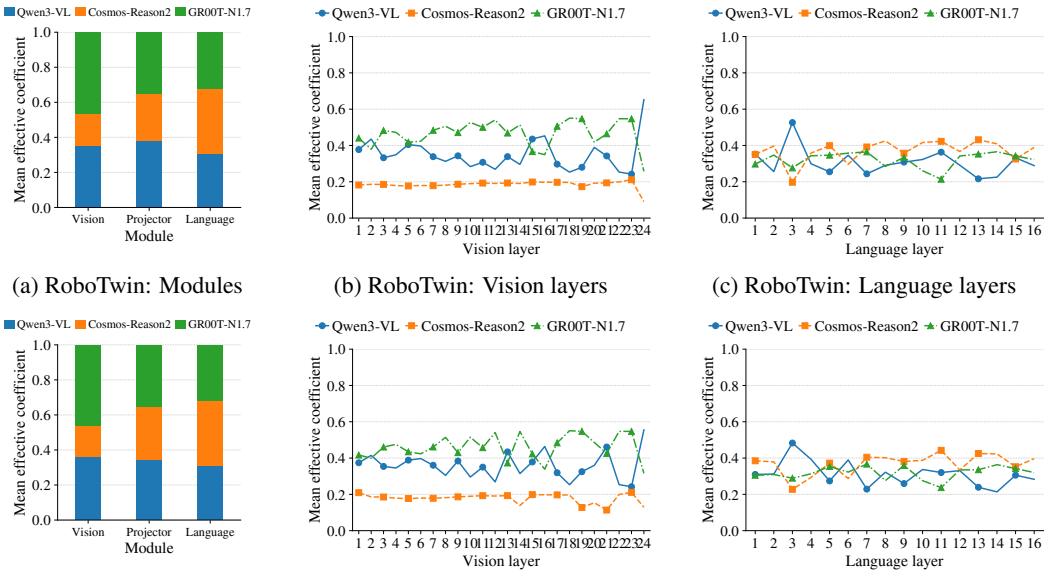
Table C.1: Effective module-level source coefficients before calibration, aggregated with weights proportional to parameter coordinate counts.

Module	Qwen	Cosmos	GR00T
Vision	0.41	0.15	0.43
Projector	0.29	0.40	0.31
Language	0.32	0.38	0.30

Figure C.1 shows module-level and layerwise effective source coefficients before calibration. **Table C.1** summarizes module coefficients weighted by parameter coordinate counts. **Panels (a), (b), and (c)** show module-level, vision-layer, and language-layer results, respectively. This WIDEN-SCE initialization (Yu et al., 2024; Wan et al., 2025) is determined solely by source weights without target samples, making it independent of the target dataset. LIBERO, RoboTwin, and real-world experiments therefore start calibration from the same merged parameters, and we report one shared set of results. GR00T has the largest Vision coefficient (0.43), while Cosmos has the largest Projector and Language coefficients (0.40 and 0.38). These coefficients describe algebraic parameter proportions.

C.2 Source Allocation After Calibration

Figure C.2 shows effective source coefficients after 20,000 calibration steps on RoboTwin and real-world data; Figure 3(d)–(f) gives the corresponding LIBERO results. **Figure C.1** gives the shared initialization, and **Table C.2** summarizes module coefficients at step 20,000, with signed changes relative to initialization in parentheses. Red denotes increases and green decreases; changes were computed from unrounded coefficients. Each dataset used its own training images and instructions for calibration without action annotations. Source weights remained frozen, and only the 986 tensor scales for Cosmos and GR00T were optimized. Scales started at 1, were constrained to 0–1.25, and were taken from step 20,000. All three datasets exhibit the same module-level trends: Vision coefficients increase for Cosmos and GR00T and decrease for Qwen; Projector coefficients decrease for Cosmos and increase for Qwen and GR00T; Language undergoes only small adjustments. On real-world data, Projector shows the clearest source reallocation, with a decrease in the Cosmos coefficient and increases in the Qwen and GR00T coefficients.



(a) RoboTwin: Modules (b) RoboTwin: Vision layers (c) RoboTwin: Language layers

(d) Real-world: Modules (e) Real-world: Vision layers (f) Real-world: Language layers

Figure C.2: Effective source coefficients after calibration on RoboTwin (top) and real-world data (bottom), at step 20,000. Columns show modules, vision layers, and language layers. LIBERO results are shown in Figure 3(d)–(f). **Figure C.1** gives the shared initialization, and **Table C.2** reports changes in module coefficients.

Table C.2: Effective module-level source coefficients after calibration on three datasets (step 20,000). Parentheses indicate changes from the initial coefficients: red plus signs denote increases and green minus signs denote decreases. Final values and changes are independently rounded to two decimal places.

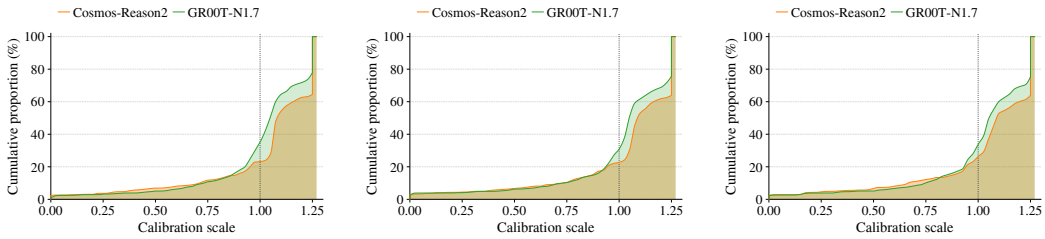
Dataset	Module	Qwen	Cosmos	GR00T
LIBERO	Vision	0.34 (−0.07)	0.19 (+0.03)	0.47 (+0.04)
	Projector	0.34 (+0.05)	0.30 (−0.10)	0.36 (+0.05)
	Language	0.31 (−0.01)	0.37 (−0.01)	0.32 (+0.02)
RoboTwin	Vision	0.35 (−0.06)	0.18 (+0.03)	0.46 (+0.03)
	Projector	0.38 (+0.09)	0.27 (−0.13)	0.35 (+0.04)
	Language	0.31 (−0.01)	0.37 (−0.01)	0.32 (+0.02)
Real-world	Vision	0.36 (−0.05)	0.18 (+0.02)	0.46 (+0.03)
	Projector	0.34 (+0.05)	0.31 (−0.09)	0.35 (+0.04)
	Language	0.31 (−0.01)	0.37 (−0.01)	0.32 (+0.02)

C.3 Calibration Scale Distributions

Figure C.3 shows cumulative distributions of calibration scales, and **Table C.3** provides summary statistics. **Panels (a), (b), and (c)** correspond to LIBERO, RoboTwin, and real-world data, respectively. Each dataset has 493 calibrated scales for each of Cosmos and GR00T. All means are close to 1, but population standard deviations are approximately 0.25–0.29, and some scales accumulate at both bounds, indicating nonuniform scaling across tensors. The mean Cosmos scales are 1.04, 1.04, and 1.04 on LIBERO, RoboTwin, and real-world data, respectively, while the GR00T means are 1.01, 1.02, and 1.02. More scales exceed 1 on all three datasets; the corresponding counts for Cosmos are 379, 381, and 363.

C.4 Joint Adjustment of Two Sources

Figure C.4 shows joint changes in Cosmos and GR00T scales for the same tensor, and **Table C.4** summarizes counts and proportions by adjustment type. **Panels (a), (b), and (c)** correspond to LIBERO, RoboTwin, and real-world data, respectively. Amplifying both sources is the most frequent joint adjustment, accounting for 52.13%, 55.58%, and 49.90% on the three datasets. Attenuating both accounts for 10.34%, 9.13%, and 10.14%. Opposite adjustments account for 37.53%,

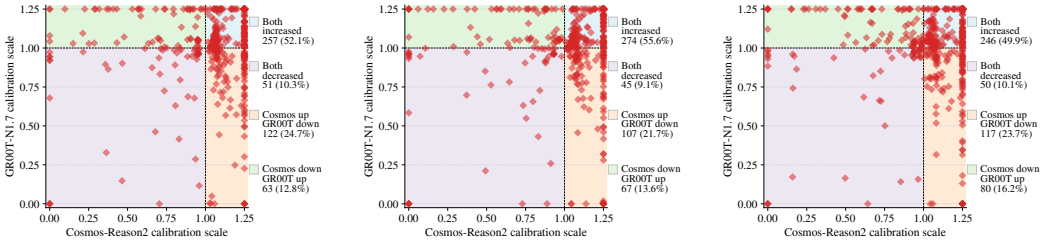


(a) LIBERO (b) RoboTwin (c) Real-world

Figure C.3: Cumulative distributions of calibration scales on three datasets. Each panel compares the distributions of 493 tensor scales each for Cosmos and GR00T. The horizontal axis is the scale, and the vertical axis is the cumulative proportion. Curves are smoothed visualizations of empirical cumulative distributions.

Table C.3: Calibration scale distribution statistics on three datasets.

Dataset	Source	Count	Mean ± pop. std.	Median	> 1	< 1	Lower bound 0	Upper bound 1.25
LIBERO	Cosmos	493	1.04 ± 0.28	1.08	379	114	13	176
	GR00T	493	1.01 ± 0.25	1.05	320	173	10	116
RoboTwin	Cosmos	493	1.04 ± 0.29	1.08	381	112	16	179
	GR00T	493	1.02 ± 0.28	1.05	341	152	17	122
Real-world	Cosmos	493	1.04 ± 0.29	1.08	363	130	14	180
	GR00T	493	1.02 ± 0.27	1.05	326	167	13	129



(a) LIBERO (b) RoboTwin (c) Real-world

Figure C.4: Joint adjustment of two source calibration scales for the same tensor on three datasets. Each panel pairs the Cosmos and GR00T scales; a scale of 1 preserves the initial residual magnitude.

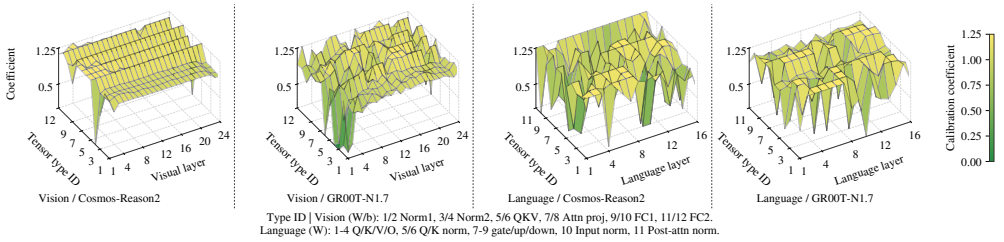
Table C.4: Joint adjustment of two source scales for the same tensor on three datasets.

Dataset	Tensor pairs	Both amplified	Both attenuated	Cosmos ↑ GR00T ↓	Cosmos ↓ GR00T ↑	Opposite adjustments
LIBERO	493	257 (52.13%)	51 (10.34%)	122 (24.75%)	63 (12.78%)	185 (37.53%)
RoboTwin	493	274 (55.58%)	45 (9.13%)	107 (21.70%)	67 (13.59%)	174 (35.29%)
Real-world	493	246 (49.90%)	50 (10.14%)	117 (23.73%)	80 (16.23%)	197 (39.96%)

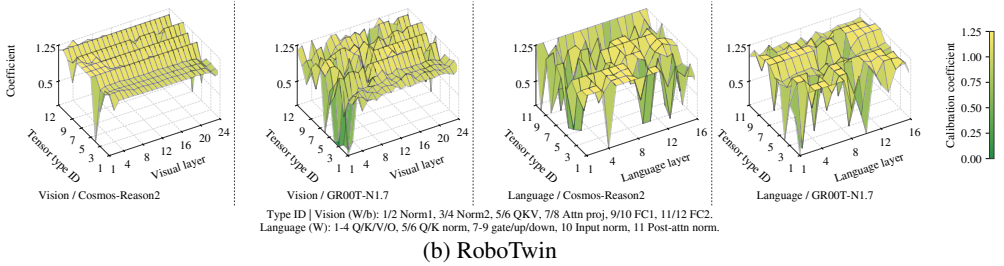
35.29%, and 39.96%, showing that calibration both amplifies sources together and reallocates a substantial share of their contributions.

C.5 TENSOR-LEVEL CALIBRATION SURFACES

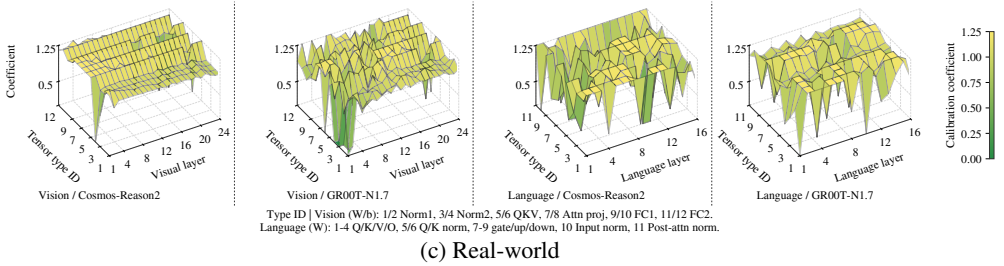
Figure C.5 shows tensor-level calibration surfaces, and **Table C.5** summarizes the scales included in these surfaces by module. **Panels (a), (b), and (c)** correspond to LIBERO, RoboTwin, and real-world data. The surfaces contain 288 vision-block and 176 language-block scales per source, totaling 928 scales. Mean Cosmos scales in vision exceed 1 on all three datasets, at 1.12, 1.11, and 1.10. Mean language scales for Cosmos are 0.97, 0.98, and 1.00, compared with 1.03, 1.04, and 1.04 for GR00T. Variation across layers and tensor types indicates substantial positional differences in calibration strength. These surfaces exclude the remaining scales for the stem, Projector, token embeddings, and final language normalization.



(a) LIBERO



(b) RoboTwin



(c) Real-world

Figure C.5: Tensor-level calibration scale surfaces on three datasets. Each panel shows how Cosmos and GR00T scales vary with layer position and tensor type in vision and language blocks. Module-level summary statistics are reported in **Table C.5**.

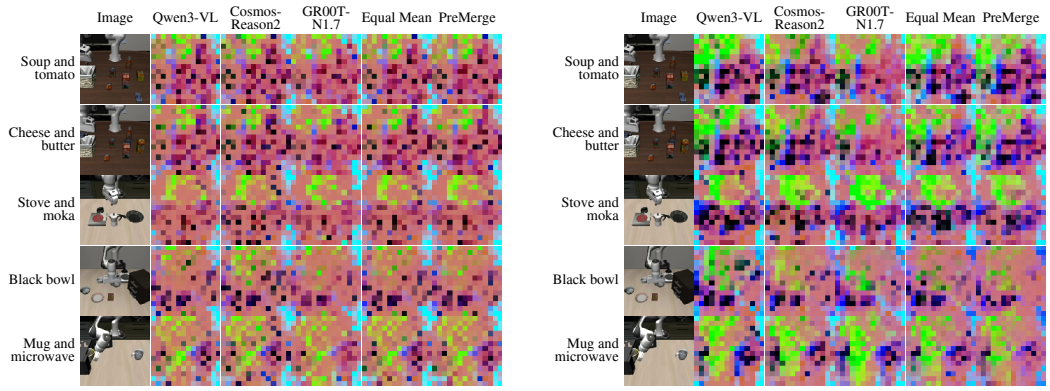
Table C.5: Tensor-level calibration surfaces and module statistics on three datasets.

Dataset	Module	Source	Mean ± pop. std.	Lower-bound count	Upper-bound count
LIBERO	Vision	Cosmos	1.12 ± 0.15	3	99
	Vision	GR00T	1.01 ± 0.24	7	46
	Language	Cosmos	0.97 ± 0.36	6	74
	Language	GR00T	1.03 ± 0.28	3	65
RoboTwin	Vision	Cosmos	1.11 ± 0.16	3	99
	Vision	GR00T	1.01 ± 0.26	11	47
	Language	Cosmos	0.98 ± 0.37	8	76
	Language	GR00T	1.04 ± 0.30	6	68
Real-world	Vision	Cosmos	1.10 ± 0.19	5	98
	Vision	GR00T	1.01 ± 0.27	10	54
	Language	Cosmos	1.00 ± 0.35	5	78
	Language	GR00T	1.04 ± 0.27	3	68

D Additional Qualitative Results

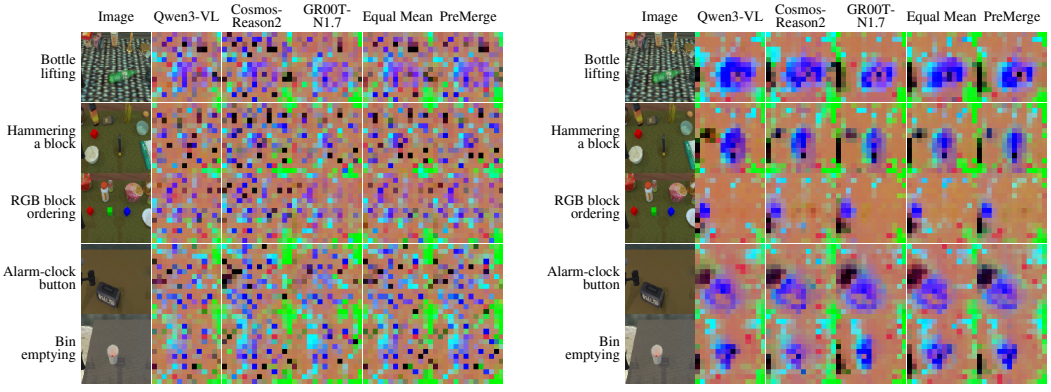
These qualitative comparisons supplement Section 5.6 for LIBERO and RoboTwin only. They visualize changes after action training, whereas **Appendix C** analyzes coefficient calibration before action training.

D.1 Visual Representations Before and After Training



(a) LIBERO: Before training

(b) LIBERO: After training



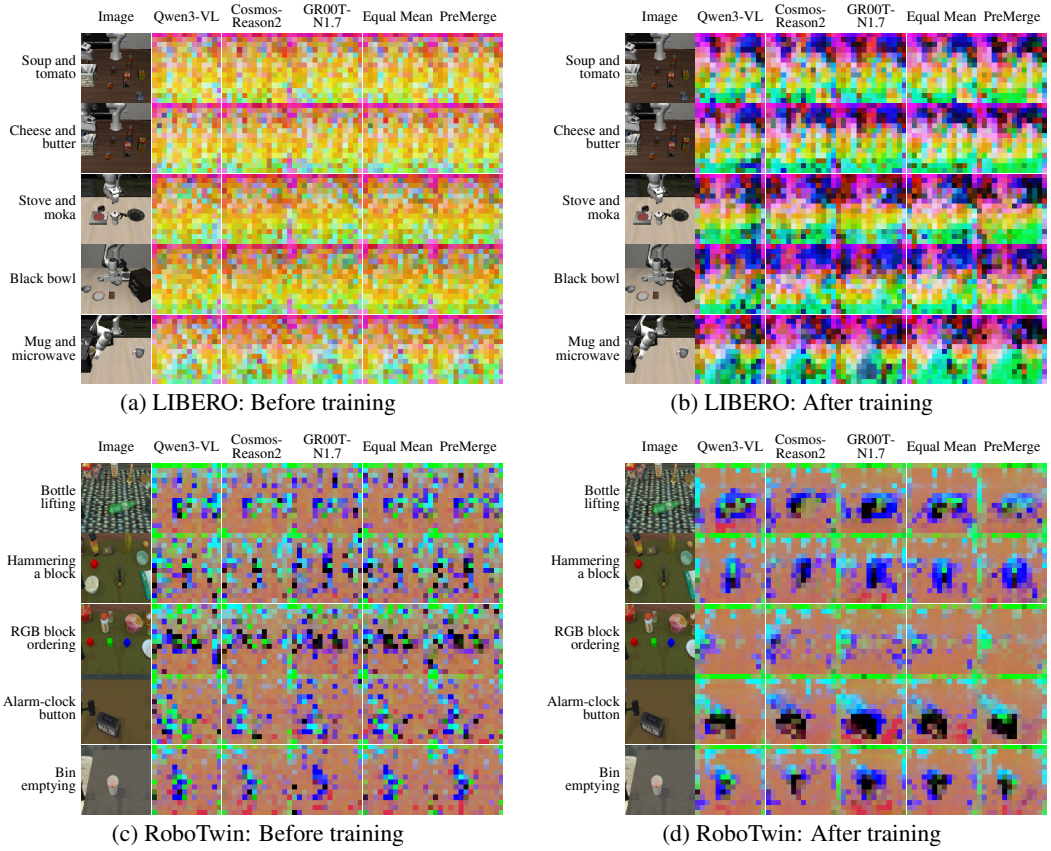
(c) RoboTwin: Before training

(d) RoboTwin: After training

Figure D.1: PCA visualization of visual representations before and after training. The top and bottom rows show LIBERO and RoboTwin, respectively; the left and right columns show representations before and after training. One frame is displayed per task. For each dataset, the post-training reference basis is fitted using ten frames from five tasks, and features before training are projected onto the same basis.

Figure D.1 compares visual representations before and after training on LIBERO and RoboTwin. LIBERO results appear in **panels (a) and (b)**, and RoboTwin results in **panels (c) and (d)**. Both datasets used 224×224 inputs. The post-training reference basis was fitted using 9,800 L2-normalized patches from five models and ten frames; representations before training were projected directly into the same coordinate system. The first three reference principal components explain 26.96% and 34.26% of variance on LIBERO and RoboTwin, respectively. The same scene uses the same reference basis, and the spatial organization of color regions changes after training. Different initializations still exhibit different local responses to the same input. PCA colors encode projection coordinates.

D.2 Final-Layer VLM Representations Before and After Training



(a) LIBERO: Before training

(b) LIBERO: After training

(c) RoboTwin: Before training

(d) RoboTwin: After training

Figure D.2: PCA visualization of final-layer VLM representations before and after training. The top and bottom rows show LIBERO and RoboTwin, respectively; the left and right columns show representations before and after training. One frame is displayed per task. For each dataset, the post-training reference basis is fitted using ten frames from five tasks, and features before training are projected onto the same basis.

Figure D.2 shows final-layer VLM representations before and after training: **panels (a) and (b)** correspond to LIBERO, and **panels (c) and (d)** to RoboTwin. The analysis used 448×448 inputs and fitted a post-training reference basis from 9,800 normalized patches across five models and ten frames. Results before training used the same center, principal-component directions, and whitening variances. The first three components explain 14.98% and 24.80% of variance, below the 26.96% and 34.26% for visual representations. PreMerge and single sources exhibit different local responses in some regions.