GeoLACE: Geometric Layerwise Action-Conditioned Enhancement for Vision-Language-Action Models

Anonymous authors
Paper under double-blind review

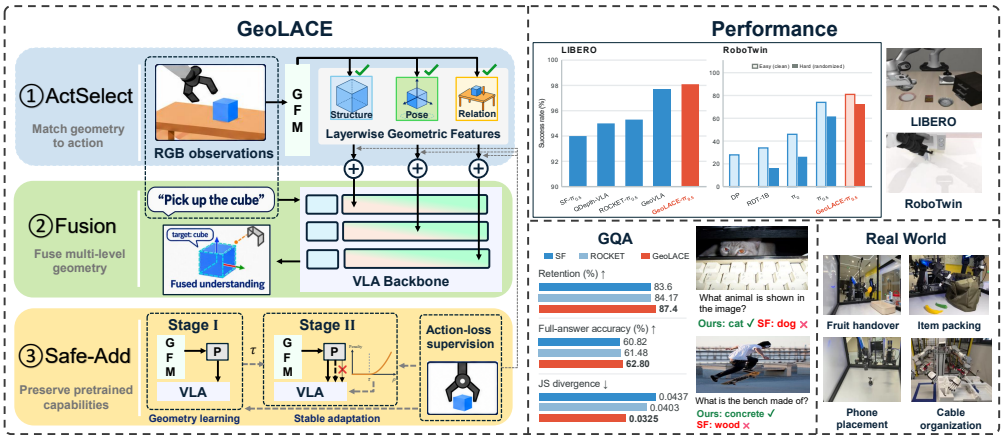


Figure 1: **Overview of GeoLACE.** Left: (1) **ActSelect** selects complementary GFM source layers and action-relevant VLA target layers; (2) **Fusion** injects multi-level geometric features into visual tokens at the selected VLA layers; (3) **Safe-Add** combines Stage I geometry learning with Stage II stable adaptation to preserve pretrained vision–language capabilities. Right: manipulation performance on LIBERO and RoboTwin, quantitative and qualitative GQA evaluations of vision–language capability retention, and real-world task examples. SF denotes Spatial Forcing.

Abstract

Generalist vision–language–action (VLA) models benefit from vision–language pretraining but have limited geometric understanding for precise manipulation. Promising approaches use geometric foundation models (GFMs) to provide VLA policies with geometric priors from RGB observations. However, these methods often lack a systematic layer-selection strategy that jointly considers geometric complementarity and relevance to action prediction. This can lead to redundant or underutilized geometric features, limiting gains in manipulation performance. Incorporating geometric knowledge can also disrupt pretrained VLA representations, potentially compromising vision–language capabilities. To address these challenges, we propose GeoLACE, a plug-and-play framework that selects complementary GFM layers and action-relevant VLA layers. It performs layerwise geometric fusion using multi-level features extracted from current RGB observations. We further introduce a two-stage training strategy that enables the policy to better exploit these priors while preserving its pretrained capabilities. We integrate GeoLACE into representative VLA backbones with different architectures. Experiments on LIBERO, RoboTwin, and two distinct real-world bimanual platforms demonstrate improved manipulation performance, with particularly substantial gains in real-world manipulation. GQA evaluation further supports better retention of pretrained vision–language capabilities than the compared methods.

1 Introduction

Vision–language–action (VLA) models (Brohan et al., 2022; 2023; Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025) map visual observations and language instructions to robot

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.

actions, leveraging large-scale vision–language knowledge to generalize across diverse manipulation tasks. This knowledge supports object recognition and instruction following, helping robots interact with the physical world. However, precise interaction with this world also requires geometric understanding of object poses, spatial relations, and 3D structure. Such geometric understanding is often insufficiently developed in VLAs pretrained on RGB-only datasets. Representative geometry-augmented VLA methods (Qu et al., 2025; Sun et al., 2025; Bhat et al., 2025; Li et al., 2026a) mainly incorporate 3D information as an explicit input. These methods require geometric preprocessing, which may involve inaccurate depth estimates or additional sensors. Geometric foundation models (GFMs) extract geometric priors directly from RGB observations, reducing the need for explicit 3D data acquisition. Consequently, GFMs have attracted growing interest for their potential to enhance geometric understanding in VLA policies.

Building on the geometric priors provided by GFMs, methods have enhanced VLA policies by injecting geometric features into the policy network (Lin et al., 2025; Yu et al., 2026) or aligning VLA representations with them (Li et al., 2026b). Despite these different mechanisms, these methods draw on features from a single GFM layer. Different GFM layers can capture complementary geometric information, from fine-grained object structure to spatial relations across the scene (Zhang et al., 2026). This layerwise complementarity motivates multi-layer GFM–VLA alignment (Sun et al., 2026; Ding et al., 2026). However, similar geometric features across the selected layers can introduce redundancy (Li et al., 2026c), while their correspondence with VLA layers can overlook the requirements of action prediction. Incorporating geometric knowledge may also disrupt pretrained VLA representations, potentially compromising vision–language capabilities (Hancock et al., 2026). These limitations raise three challenges: (1) selecting complementary GFM source layers and action-relevant VLA target layers; (2) integrating the selected geometric features into VLA policies to support precise manipulation; and (3) improving geometric understanding during task adaptation while preserving pretrained vision–language capabilities.

To address these challenges, we propose GeoLACE, a plug-and-play framework that augments pretrained VLA policies through layerwise fusion of complementary geometric features. First, GeoLACE introduces ActSelect, which combines CKA-based selection of complementary frozen GFM source layers with short-run native-action-loss probes to select action-relevant VLA target layers. Second, layerwise geometric fusion injects multi-level features from current RGB observations into the visual-token representations of selected VLA layers, preserving camera-view and image-patch correspondence for action generation. Third, GeoLACE uses projectors with zero-initialized output layers and a two-stage Safe-Add training scheme to incorporate geometric knowledge while limiting disruption to pretrained representations. Zero initialization preserves the pretrained policy at attachment, while two-stage training first learns task-relevant residuals and then controls their contribution through stochastic scaling and soft regularization. Together, these components unify source–target selection, layerwise geometric fusion, and controlled residual adaptation.

We evaluate GeoLACE on the LIBERO and RoboTwin simulation benchmarks and two distinct real-world bimanual platforms: ALOHA and a custom dual-arm robot. As a plug-and-play geometry adapter for different VLA backbones, GeoLACE raises average LIBERO success from 96.9% to 98.1% on $\pi_{0.5}$ and from 97.0% to 98.6% on GR00T N1.6, the highest average among the methods in our comparison. Frozen-model GQA evaluation further shows a 1.98-percentage-point accuracy gain over direct implicit feature alignment; together with the representation analyses, this supports better retention of the evaluated pretrained vision–language capabilities while incorporating geometry.

Our contributions are:

- **Geometric complementarity and layer-dependent contributions to action prediction.** We find complementary geometric information across GFM layers and show that its action utility varies with the VLA injection layer. Both the GFM source and VLA injection location shape effective geometric augmentation.
- **A plug-and-play framework for hierarchical geometric augmentation.** We propose GeoLACE, combining ActSelect source–target layer selection, layerwise geometric fusion, and two-stage Safe-Add training to incorporate geometry while preserving pretrained vision–language capabilities.
- **Improved manipulation performance and vision–language capability retention.** Experiments on LIBERO, RoboTwin, and two real-world platforms show manipulation gains across two VLA backbones. Frozen GQA evaluation and representation analyses support better retention than direct implicit feature alignment.

2 Related Work

Explicit Geometric Inputs for VLA Policies. Geometric understanding is essential for precise manipulation but remains a challenge for generalist VLA models. One line of work makes geometry explicit in the policy input. SpatialVLA (Qu et al., 2025) augments visual tokens with egocentric 3D position encodings derived from predicted depth, PointVLA (Li et al., 2026a) incorporates point-cloud inputs, GeoVLA (Sun et al., 2025) processes RGB and point-cloud streams in parallel, and 3D-CAVLA (Bhat et al., 2025) incorporates depth-derived point-cloud features. These designs make structured spatial cues directly available to action prediction. However, they require explicit construction of depth, point clouds, or 3D coordinates and may introduce additional sensing, preprocessing, or cross-modal calibration. This motivates RGB-only approaches that leverage geometric priors from pretrained models without requiring explicit 3D inputs.

Single-Level Geometric Integration with GFMs. Geometric foundation models (GFMs) provide learned geometric priors directly from RGB observations. DUSt3R (Wang et al., 2024) regresses point maps from uncalibrated image pairs, VGGT (Wang et al., 2025) jointly predicts camera parameters, depth, and point maps from multi-view RGB observations, and Depth Anything 3 (DA3) (Lin et al., 2026) predicts spatially consistent geometry from single- or multi-view RGB inputs. VLA methods incorporate these priors through inference-time feature integration or training-time supervision. Evo-0 (Lin et al., 2025) fuses final-layer VGGT features with visual tokens through lightweight cross-attention, while 3D-Mix (Yu et al., 2026) adaptively combines VGGT geometry with VLA semantic features through semantic-conditioned gating. Spatial Forcing (Li et al., 2026b) aligns one intermediate VLA representation with VGGT features, whereas QDepth-VLA (Li et al., 2025) introduces auxiliary quantized-depth prediction. These methods generally use a single geometric representation or auxiliary target at manually specified injection or supervision sites, without explicitly exploiting complementary geometric information across GFM layers.

Multi-Level Geometric Integration with GFMs. Recent work explores geometric information across network depths. SpatialStack (Zhang et al., 2026) injects layer-specific geometric features into successive language-model layers for spatial reasoning in VLMs. ROCKET (Sun et al., 2026) uses a shared projector to align VGGT features with multiple VLA residual streams as training-time supervision. Mind-VLA (Ding et al., 2026) makes multi-level supervision instruction-aware by aligning VLA representations with VGGT features of canonical target-object views, without requiring online VGGT feature extraction at inference. However, exploiting multiple layers alone does not ensure action-relevant source–target correspondences or preservation of pretrained vision–language capabilities. GeoLACE combines complementary GFM source selection and action-conditioned VLA target selection with layerwise geometric fusion and controlled residual adaptation, making current geometric observations available to action prediction while preserving pretrained representations.

3 Method

3.1 Geometric Feature Analysis and Method Overview

To examine how geometric information varies across GFM layers, we analyze frozen VGGT features on LIBERO, a robot manipulation benchmark.

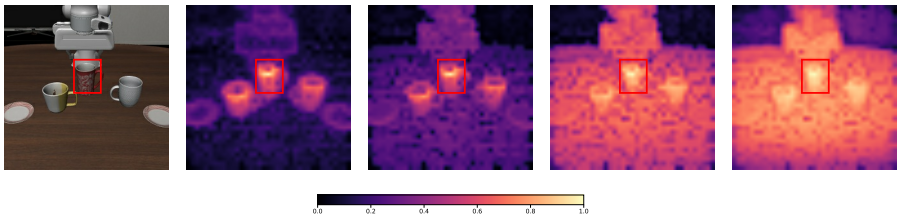


Figure 2: **Layer-wise VGGT feature similarities.** Patch-wise similarity maps for the same red-boxed reference patch across progressively deeper VGGT layers on a LIBERO observation.

We fix the reference patch and compare its similarity to other image patches across frozen VGGT layers. As shown in Figure 2, the maps shift from localized patterns at shallower layers to broader, more diffuse patterns at deeper layers. This qualitative shift is consistent with geometric representa-

tions spanning fine-grained object structure and broader scene-level context, motivating the selection of complementary GFM source layers.

However, the contribution of these features to action prediction varies across VLA injection layers, and their integration may disrupt pretrained vision–language capabilities. We therefore propose GeoLACE, which uses ActSelect to select source–target layer pairs, injects current geometric features into visual-token representations, and jointly fine-tunes the VLA and projectors through two-stage Safe-Add training (Figure 3). We use VGGT as the frozen GFM in all experiments.

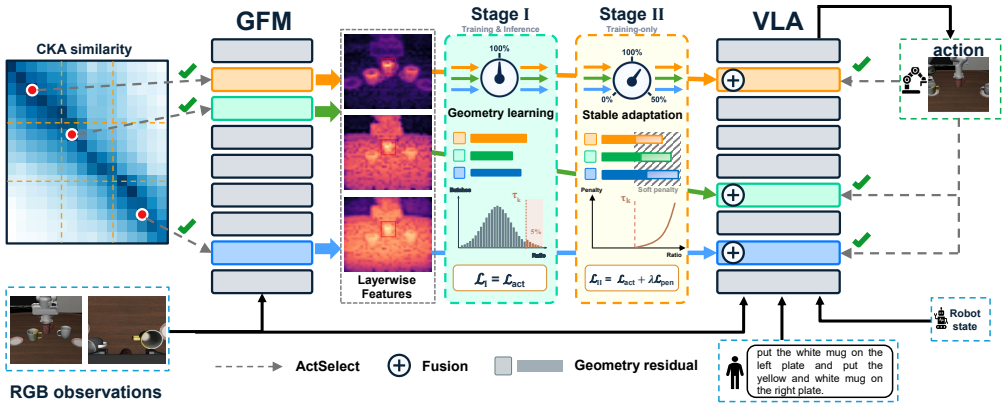


Figure 3: **GeoLACE framework. ActSelect** selects complementary GFM source layers and action-relevant VLA target layers. **Fusion** uses a geometric residual branch for each selected source–target pair at its VLA target layer. **Safe-Add** has two training stages: Stage I for geometry learning and Stage II for stable adaptation.

At time t, a pretrained VLA receives RGB views $\{I_t^c\}_{c=1}^C$, robot state s_t, and language instruction q, and predicts an H-step action chunk. GeoLACE preserves the input interface and native action prediction objective of the VLA.

For the flow-matching backbone (Black et al., 2024) used in our main experiments, the native action prediction objective is

$$x_u = u\epsilon + (1-u)a, \qquad \mathcal{L}_{\text{act}} = \mathbb{E}\big[\|v_\theta(x_u, o_t, u) - (\epsilon - a)\|_2^2\big]. \tag{1}$$

Here, a is the demonstration action chunk, $o_t = (\{I_t^c\}_{c=1}^C, s_t, q)$ is the policy input, $u \in [0,1]$ is the flow time, and $\epsilon \sim \mathcal{N}(0, I)$ is Gaussian noise with the same shape as a. The noisy action chunk is denoted by x_u, and $v_\theta(x_u, o_t, u)$ is the policy's velocity-field prediction parameterized by θ. For other VLA backbones, GeoLACE uses their native supervised action prediction objectives.

3.2 ActSelect: Offline Source–Target Selection

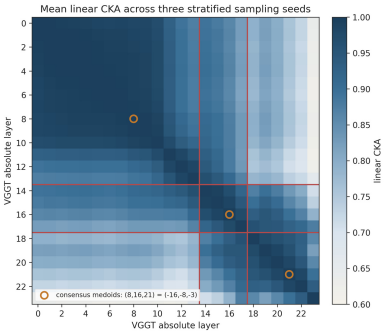
ActSelect selects complementary GFM source layers and maps them monotonically to action-relevant VLA target layers using a lightweight offline calibration procedure. First, we keep VGGT frozen and use linear CKA (Kornblith et al., 2019) on calibration observations to organize its intermediate features into complementary contiguous stages. We select a representative layer from each stage as a GFM source. Figure 4(a) visualizes the inter-layer CKA structure and the resulting source selection. This CKA-based step fixes the GFM sources, while choosing the VLA layers that can best use these features for action prediction requires action-conditioned evaluation.

We next perform action-conditioned target selection using the native action loss. With both the VLA and VGGT frozen, we briefly train three temporary projectors, one per GFM source and shared across candidate VLA layers, under sampled monotonic assignments. We then introduce a scalar gate $g_{k\ell}$ on the residual from source k to target layer ℓ. Its local utility is the reduction in action loss predicted at the zero-residual point:

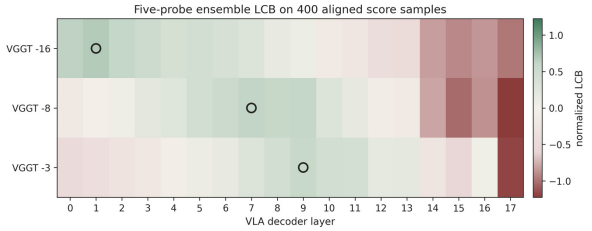
$$q_{k\ell} = -\mathbb{E}_{\mathcal{B},u}\left[\frac{\partial \mathcal{L}_{\text{act}}}{\partial g_{k\ell}}\bigg|_{\mathbf{g}=\mathbf{0}}\right]. \tag{2}$$

Here, $\mathbf{g}$ denotes all source–target gates. Scores are accumulated over held-out calibration batches and flow times. Within each probe, scores are centered over candidate layers for each source and standardized over sample–layer entries. Let $\tilde{q}_{k\ell}^{(p)}$ denote the mean normalized score for source–target pair (k, ℓ) in probe p. We aggregate these scores across probes using the lower-confidence-bound (LCB) score $q_{k\ell}^{\mathrm{LCB}} = \bar{\tilde{q}}_{k\ell} - \mathrm{SE}_p(\tilde{q}_{k\ell}^{(p)})$, where the mean and standard error are computed across probes. Figure 4(b) visualizes the resulting source–target LCB scores, with circles marking the selected layer pairs. The final mapping maximizes the summed score subject to $\ell_1 < \ell_2 < \ell_3$. We further validate the selected mapping by measuring changes in action loss on held-out batches at different residual RMS levels (Figure 10(b)).

By considering both geometric complementarity and action relevance, ActSelect adapts source–target selection to different VLA backbones and task datasets. This enables hierarchical geometry integration without prescribing a fixed layer mapping across settings. GFM source-layer selection and VLA target-layer selection are detailed in Appendices A.1 and A.2, respectively, with additional selection analyses in Appendix B.



(a) CKA-based GFM source selection.



(b) Action-conditioned target selection.

Figure 4: **CKA-based source and action-conditioned target selection.** (a) Inter-layer CKA reveals contiguous stages for GFM source selection. (b) LCB scores quantify the action-loss utility of source–target pairs; circles indicate the selected monotonic mapping.

3.3 Online Layerwise Geometric Fusion

Given the selected pairs, the current camera images are processed along two paths: the native VLA visual encoder and a frozen VGGT encoder. For a source feature G_k^c from camera c, we remove the camera and register tokens and reshape the remaining patch tokens into a regular two-dimensional grid. When the source and target grids have different resolutions, we resize the source feature map using two-dimensional bilinear interpolation with corner alignment disabled; when their resolutions coincide, no resampling is performed. The aligned feature map is then flattened in raster-scan order and mapped by an independent projector:

$$R_k^c = W_{k,2}\,\mathrm{GELU}(W_{k,1}\mathrm{LN}_k(G_k^c))\,, \qquad H_{\ell_k}^{\mathrm{img},c} \leftarrow H_{\ell_k}^{\mathrm{img},c} + \alpha R_k^c. \tag{3}$$

Here, G_k^c denotes the spatially aligned patch tokens, k indexes the selected source–target pairs, and ℓ_k denotes the VLA target layer paired with source k. $H_{\ell_k}^{\mathrm{img},c}$ denotes the visual-token features associated with camera c at layer ℓ_k, and R_k^c is the projected geometric residual with the same shape. LN_k denotes layer normalization, and $W_{k,1}$ and $W_{k,2}$ are the learnable projection matrices for branch k. Bias terms are omitted for brevity. The scalar α controls the contribution of the geometric residual, as specified in the two-stage training scheme below. Equation (3) is applied independently to each camera span. Camera slots marked as invalid by the input mask are excluded from geometric processing, and their original VLA image tokens are restored after fusion. Thus geometric features are updated online during execution and participate directly in closed-loop action generation. At inference, camera-aligned VGGT features can be reused across policy requests (Appendix D.4). In the $\pi_{0.5}$ implementation, the features are reused across action-denoising steps for the same observation (Appendix A.3).

All VGGT parameters remain frozen. The output linear layer of every geometry projector, including its bias, is initialized to zero. Hence $R_k^c = 0$ at attachment time, preserving the pretrained policy's velocity prediction:

$$v_{\theta_0,\phi_0}(x_u, o_t, u) = v_{\theta_0}(x_u, o_t, u), \tag{4}$$

where θ_0 denotes the pretrained VLA parameters and ϕ_0 denotes the initial geometry-projector parameters. The projectors subsequently learn task-specific residuals. This construction requires only camera-token spans, callbacks at candidate VLA blocks, and access to the model's native action loss; residuals are added only to visual tokens, with no direct addition to language or action tokens.

3.4 Two-Stage Safe-Add Training

Additive fusion introduces online geometric residuals into the pretrained VLA's visual-token representations. Safe-Add uses two training stages to learn task-relevant residuals while controlling their contribution to the pretrained policy.

Stage I (geometry learning). We jointly fine-tune the VLA and geometry projectors with zero-initialized output layers. The Stage-I objective and the relative residual magnitude recorded for branch k, camera c, and sample b are

$$\mathcal{L}_{\mathrm{I}} = \mathcal{L}_{\mathrm{act}}, \qquad \rho_{kbc} = \frac{\mathrm{RMS}(R_k^{bc})}{\mathrm{RMS}(H_{\ell_k}^{\mathrm{img},bc}) + \varepsilon}. \tag{5}$$

For each branch, τ_k is the 95th percentile of batch-mean residual ratios collected near the end of Stage I, with each mean computed across all valid sample–camera pairs. This branch-specific reference is shared across camera views.

Stage II (stable adaptation). We resume the Stage-I model and optimizer state and retain direct-add fusion. The Stage-II training objective softly penalizes the excess of each sample–camera ratio above the fixed Stage-I batch-mean percentile τ_k:

$$\mathcal{L}_{\mathrm{II}} = \mathcal{L}_{\mathrm{act}} + \lambda \mathcal{L}_{\mathrm{pen}}, \qquad \mathcal{L}_{\mathrm{pen}} = \frac{1}{KC} \sum_{k,c} \mathbb{E}_b \big[\max(0, \rho_{kbc} - \tau_k)^2\big]. \tag{6}$$

In $\mathcal{L}_{\mathrm{pen}}$, ρ_{kbc} is evaluated with $\mathrm{sg}(\cdot)$ applied to its base-feature RMS denominator. During Stage II, one shared scale $\alpha \in \{1, 0.5, 0\}$ is sampled per forward pass, producing full, attenuated, and geometry-off paths under the same optimization objective. At inference, $\alpha = 1$ and the auxiliary penalty is removed. Together, residual scaling and soft regularization enable the policy to exploit task-relevant geometry while discouraging excessive residual contributions, promoting stable adaptation of pretrained representations.

4 Experiments

Base models and implementation details. We instantiate GeoLACE with $\pi_{0.5}$ (Physical Intelligence et al., 2025) as the primary VLA backbone for both simulation and real-world experiments. To evaluate the architectural generality of our method, we additionally integrate GeoLACE into GR00T N1.6 (NVIDIA GEAR Team, 2025). In both instantiations, VGGT is kept frozen, while the VLA backbone and geometry projectors are jointly fine-tuned. Each model is trained on 4 NVIDIA H20 GPUs.

Benchmarks and metrics. We evaluate GeoLACE on LIBERO (Liu et al., 2023), RoboTwin 2.0 (Chen et al., 2025), and two physical bimanual platforms. On LIBERO, we follow the respective official evaluation protocols, using 50 rollouts per task for $\pi_{0.5}$ and 20 for GR00T N1.6. On RoboTwin, policies are trained separately for six tasks, each evaluated over 100 trials in both easy and hard settings.

4.1 Overall Manipulation Performance

LIBERO. As shown in Table 1, GeoLACE improves the average success rate of $\pi_{0.5}$ from 96.9% to 98.1%, a gain of 1.2 percentage points. Improvements are observed on Object, Goal, and Long, while performance on Spatial remains unchanged at 98.8%. The improvement on LIBERO-Long suggests that geometric augmentation can also benefit long-horizon manipulation, where successful execution requires maintaining accurate interactions over multiple steps. With GR00T N1.6, GeoLACE achieves an average success rate of 98.6%, exceeding the officially reported baseline of 97.0% by 1.6 percentage points, and achieves 100% success on both Spatial and Goal. The gains on both backbones support the applicability of GeoLACE beyond a single VLA architecture.

Table 1: LIBERO success rates (%) across four task suites and their average. Best and second-best results are shown in bold and underlined, respectively.

Method	Spatial	Object	Goal	Long	Average
Basic 2D VLA					
Octo (Octo Model Team et al., 2024)	78.9	85.7	84.6	51.1	75.1
OpenVLA (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5
GR00T N1.6 (NVIDIA GEAR Team, 2025)	97.7	98.5	97.5	94.4	97.0
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
π_0 (Black et al., 2024; Kim et al., 2025)	96.8	98.8	95.8	85.2	94.2
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	<u>98.8</u>	98.2	98.0	92.4	96.9
Single-level 3D VLA					
SpatialVLA (Qu et al., 2025)	88.2	89.9	78.6	55.5	78.1
GeoVLA (Sun et al., 2025)	98.4	99.0	96.6	**96.6**	97.7
3D-CAVLA (Bhat et al., 2025)	98.2	**99.8**	98.2	<u>96.1</u>	<u>98.1</u>
QDepth-VLA (Li et al., 2025)	95.4	96.8	95.8	91.8	95.0
Spatial Forcing-$\pi_{0.5}$ (Li et al., 2026b)	97.8	97.8	94.4	85.8	94.0
Multi-level 3D VLA					
ROCKET-$\pi_{0.5}$ (Sun et al., 2026)	96.4	98.8	96.6	89.2	95.3
Mind-VLA (Ding et al., 2026)	98.0	98.0	92.0	89.5	94.4
GeoLACE-$\pi_{0.5}$	<u>98.8</u>	99.0	<u>99.2</u>	95.2	<u>98.1</u>
GeoLACE-GR00T N1.6	**100.0**	<u>99.5</u>	**100.0**	95.0	**98.6**

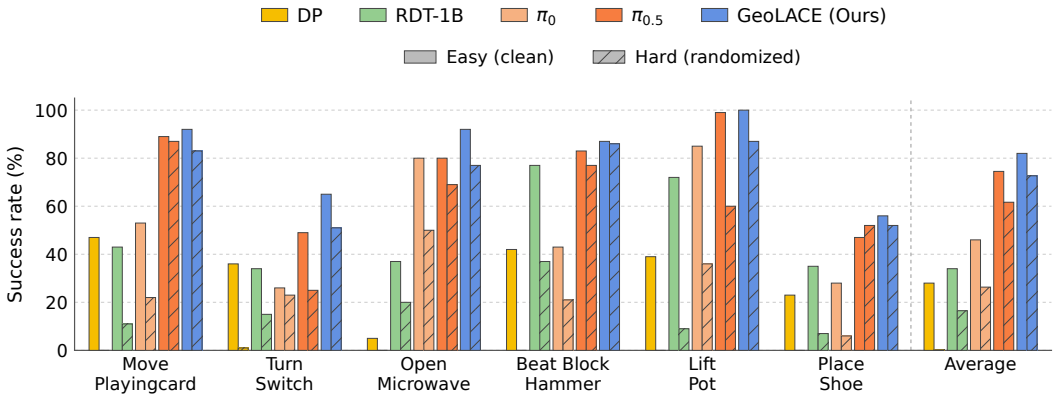


Figure 5: RoboTwin success rates across six bimanual manipulation tasks. Colors identify methods; solid and hatched bars denote easy and hard evaluation, respectively.

RoboTwin. Figure 5 compares GeoLACE with Diffusion Policy (DP) (Chi et al., 2025), RDT-1B (Liu et al., 2025), π_0, and $\pi_{0.5}$ on six RoboTwin tasks. GeoLACE exceeds $\pi_{0.5}$ by 7.5 and 11.0 percentage points in the easy and hard settings, respectively. The largest gains occur on Turn Switch under easy conditions (+16 points) and Lift Pot under hard conditions (+27 points). The improvements in success rate demonstrate the effectiveness of GeoLACE for bimanual manipulation.

4.2 Ablation Studies

We evaluate the effects of the number of source–target branches and the selection of VLA target layers. All variants use the same LIBERO training settings. Table 2 reports per-suite and average success rates. Detailed ablation settings and fusion-operator implementations are provided in Appendix A.4.

Number of source–target layer pairs. We compare the three ActSelect-selected source–target pairs in $L179$ (denoted by their VLA target layers $1, 7, 9$) with the reduced $L17$ and $L1$ configurations. The reduced variants remove branches while preserving the remaining source–target pairings. The average success rate increases from 97.2% with one branch to 97.5% with two and 98.1% with three, indicating that the full configuration benefits from branches omitted in the reduced variants.

Target-layer selection. We compare ActSelect's selected VLA targets with three random three-layer configurations. All retain the same GFM source layers and projector architectures, varying only the VLA target layers while preserving source–target order. The selected targets achieve an average success rate of 98.1%, compared with 97.3%–97.5% for the random configurations. This result supports action-conditioned selection over random selection of VLA target layers in the evaluated setting.

Table 2: **Geometry-injection ablations on LIBERO.** We compare the number of source–target branches and the choice of VLA target layers. Success rates (%) are reported for each suite and their average.

Ablation	Sites / configuration	Spatial	Object	Goal	Long	Average
Number of source–target pairs	$L1$	98.2	98.2	99.0	93.4	97.2
	$L17$	**99.2**	98.4	98.6	93.6	97.5
	$L179$	98.8	**99.0**	**99.2**	**95.2**	**98.1**
Random layer selection	Random layer selection 1	98.2	98.0	98.0	**95.2**	97.4
	Random layer selection 2	**99.2**	98.6	98.2	93.2	97.3
	Random layer selection 3	99.0	98.2	98.0	94.8	97.5

Training schedule. Table 3(a) compares three schedules with the same source–target configuration and 30k training steps. One-stage training uses only the native action loss, while two-stage training switches to Stage II after 20k or 25k steps. All other settings are identical. Our 25k+5k schedule achieves the highest average success rate (98.1%), exceeding one-stage training (97.5%) and the 20k+10k schedule (96.4%). These results suggest that introducing regularization too early or maintaining it for too long may limit adaptation, supporting a longer residual-learning stage followed by a shorter regularization stage.

Table 3: Training-schedule and fusion-operator ablations on LIBERO. (a) Training schedules with residual addition. (b) Fusion operators under one-stage 30k training.

(a) Two-stage schedule

Schedule	Spatial	Object	Goal	Long	Average
One-stage	98.4	98.8	98.6	94.0	97.5
20k+10k	98.6	97.8	97.6	91.6	96.4
25k+5k	**98.8**	**99.0**	**99.2**	**95.2**	**98.1**

(b) Fusion operator

Operator	Spatial	Object	Goal	Long	Average
Add	98.4	**98.8**	**98.6**	94.0	**97.5**
Cross-attn.	98.0	98.0	97.8	**94.4**	97.1
Gated	95.8	94.8	96.4	85.0	93.0
Weighted	**99.0**	97.8	98.0	92.8	96.9

Fusion operator. Table 3(b) compares fusion operators under one-stage, action-loss-only training for 30k steps. The 98.1% result additionally uses the 25k+5k two-stage schedule. Residual addition achieves the highest average success rate (97.5%), compared with cross-attention (97.1%), weighted fusion (96.9%), and gated fusion (93.0%). These results support residual addition as our geometry-fusion operator.

4.3 Vision–Language Capability Retention

We compare GeoLACE with the $\pi_{0.5}$ fine-tuning baseline ($\pi_{0.5}$-FT), Spatial Forcing (SF), and ROCKET, all based on the same pretrained $\pi_{0.5}$ model and fine-tuned on LIBERO. We evaluate them on 4,000 held-out questions from balanced GQA 1.2 (Hudson & Manning, 2019) without further parameter updates. Retention and acquisition measure first-token candidate accuracy on questions the pretrained reference answers correctly and incorrectly, while full-answer accuracy scores complete candidate answer sequences.

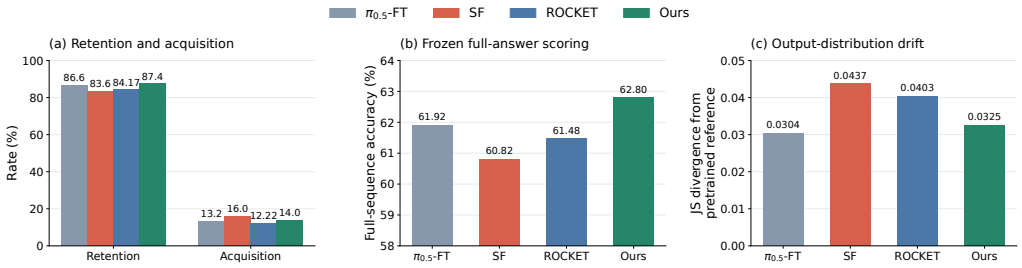


Figure 6: **Vision–language capability retention on GQA.** $\pi_{0.5}$-FT denotes the $\pi_{0.5}$ baseline fine-tuned on LIBERO. (a) Retention and acquisition under first-token candidate scoring. (b) Accuracy under full-answer candidate scoring. (c) Jensen–Shannon (JS) divergence from the pretrained reference using family-restricted first-token distributions.

Figure 6(a) shows that SF has lower retention than $\pi_{0.5}$-FT, whereas GeoLACE improves retention by 3.84 percentage points over SF, indicating better preservation of pretrained vision–language capabilities. In Figure 6(b), GeoLACE achieves the highest full-answer accuracy among the compared methods (62.80%, versus 60.82% for SF and 61.48% for ROCKET), showing that this retention is accompanied by stronger answer accuracy. Figure 6(c) shows lower JS divergence for GeoLACE than for SF and ROCKET, indicating less output-distribution drift from the pretrained reference. Additional experiments on representation retention and geometric residuals are provided in Appendix C.

4.4 Real-World Manipulation



(a) Starfruit handover **(b)** Rag pickup **(c)** Item packing **(d)** Phone placement **(e)** Cable organization

Figure 7: **Real-world tasks.** (a)–(d) Setting A (ALOHA-style); (e) Setting B (custom dual-arm).

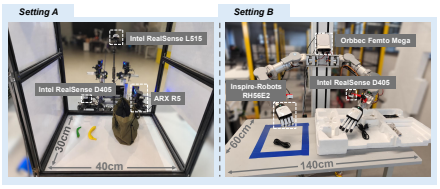


Figure 8: **Real-world robot settings.** Approximate tabletop workspaces: (a) ALOHA-style Setting A (40×30 cm); (b) custom dual-arm Setting B (140×60 cm). Labels identify cameras and arms.

Table 4: **Real-world success and inference latency.** Each policy has 20 trials per task; latency is per action chunk. Overall averages the four Setting A tasks equally.

		Success ↑		Latency (ms) ↓	
Set.	Task	$\pi_{0.5}$	Ours	$\pi_{0.5}$	Ours
A	Starfruit handover	10/20	**16/20**	78.2	101.1
A	Rag pickup	16/20	**18/20**	77.6	81.6
A	Item packing	9/20	**15/20**	77.7	85.2
A	Phone placement	6/20	**10/20**	87.2	90.4
Overall (Setting A)		51.25%	**73.75%**	80.2	89.6
B	Cable organization	6/20	**13/20**	235.1	285.0

We evaluate GeoLACE on two bimanual robot platforms, as shown in Figure 8. Setting A is an ALOHA-style platform with two ARX R5 arms, while Setting B is a custom dual-arm robot with Inspire-Robots RH56E2 dexterous hands. Both settings use three RGB views from one head-mounted and two wrist-mounted cameras. Policy inference runs on an NVIDIA RTX 4090 for Setting A and an NVIDIA Thor platform for Setting B. Detailed platform configurations are provided in Appendix D.1.

We report quantitative results on five tasks: starfruit handover, rag pickup, item packing, and phone placement on Setting A (Figure 7(a)–(d)), and cable organization on Setting B (Figure 7(e)). Task definitions and evaluation protocols are detailed in Appendices D.2 and D.3, respectively.

For each task, both policies are fine-tuned on the same 50 demonstrations and evaluated over 20 trials per policy under the same conditions. To limit the inference overhead of online geometry, we reuse cached multi-level VGGT features across policy requests and refresh them asynchronously, as detailed in Appendix D.4. As shown in Table 4, GeoLACE raises mean success across the four Setting A tasks from 51.25% to 73.75%, while mean latency rises from 80.2 to 89.6 ms. The largest success gain occurs in cable organization in Setting B (30% to 65%), where latency rises from 235.1 to 285.0 ms. GeoLACE improves success on all five tasks with modest added latency.

5 Conclusion

Our analyses reveal complementary geometric information across GFM layers and show that its utility for action prediction varies with the VLA injection layer. Building on these findings, we propose GeoLACE, which combines ActSelect-guided layerwise fusion with Safe-Add residual control. Experiments show gains in simulated and real-world manipulation, while GQA and representation analyses support retention of pretrained vision–language capabilities. Future work will extend the selected residual branches with temporal geometric memory to improve robustness to temporary occlusion. We will also explore lightweight geometric encoding to reduce online inference latency for real-time robot manipulation.

AI USE STATEMENT

In this work, we did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, design experiments, implement methods, or analyze data. Proof-related, translation, and qualitative-analysis tasks are not applicable to this work. Additionally, we used generative AI tools solely as general-purpose writing aids for light proofreading and language polishing, limited to minor improvements in grammar, phrasing, and style. The research ideas, problem formulation, methodology, experimental design, data analysis, code development, and substantive manuscript drafting were carried out by the authors. No LLM generated original research content or independently drafted any part of the manuscript. We reviewed and verified all AI-assisted edits and take responsibility for the final content of this work, including its text, claims, and artifacts. No LLM is an author or contributor to this work.

REPRODUCIBILITY STATEMENT

To support reproducibility, Sections 3 and 4 describe the GeoLACE method, training procedure, and evaluation protocols. The appendix provides implementation and hyperparameter settings, further details of the experimental analyses, and the hardware, task, and evaluation settings for the real-world experiments. We also provide source code and demonstration videos in the supplementary materials to facilitate replication and inspection of the reported results.

REFERENCES

Vineet Bhat, Yu-Hsiang Lan, Prashanth Krishnamurthy, Ramesh Karri, and Farshad Khorrami. 3d cavla: Leveraging depth and 3d context to generalize vision language action models for unseen tasks. arXiv preprint arXiv:2505.05800, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818, 2023.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research, 44(10-11):1684–1704, 2025.

Xingyu Ding, Yuzhong Zhao, Yang Wu, Chaoyang Zhao, Chunhai Zhao, Yifan Zhang, and Jian Cheng. Mind-vla: Instruction-aware spatial representation alignment for vision-language-action models. arXiv preprint arXiv:2608.04633, 2026.

Asher Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. In International Conference on Learning Representations, volume 2026, pp. 75333–75360, 2026.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6693–6702. IEEE, 2019.

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

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. arXiv preprint arXiv:2502.19645, 2025.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In International conference on machine learning, pp. 3519–3529. PMlR, 2019.

Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. IEEE Robotics and Automation Letters, 11(3):2506–2513, 2026a.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In International Conference on Learning Representations, volume 2026, pp. 132324–132345, 2026b.

Haoyuan Li, Qihang Cao, Tao Tang, Kun Xiang, Zihan Guo, Jianhua Han, JiaWang Bian, Hang Xu, and Xiaodan Liang. Thinking with geometry: Active geometry integration for spatial reasoning. arXiv preprint arXiv:2602.06037, 2026c.

Yixuan Li, Yuhui Chen, Mingcai Zhou, Haoran Li, Zhengtao Zhang, and Dongbin Zhao. Qdepth-vla: Quantized depth prediction as auxiliary supervision for vision-language-action models. arXiv preprint arXiv:2510.14836, 2025.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In International Conference on Learning Representations, 2026.

Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, Yuxin Du, Jiting Liu, Encheng Gu, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding. arXiv preprint arXiv:2507.00416, 2025.

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

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, volume 2025, pp. 29982–30009, 2025.

NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots. NVIDIA Research, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. arXiv preprint arXiv:2405.12213, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. arXiv preprint arXiv:2501.15830, 2025.

Guoheng Sun, Tingting Du, Kaixi Feng, Chenxiang Luo, Xingguo Ding, Zheyu Shen, Ziyao Wang, Yexiao He, and Ang Li. Rocket: Residual-oriented multi-layer alignment for spatially-aware vision-language-action models. arXiv preprint arXiv:2602.17951, 2026.

Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. Geovla: Empowering 3d representations in vision-language-action models. arXiv preprint arXiv:2508.09071, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5294–5306. IEEE, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20697–20709. IEEE, 2024.

Bin Yu, Shijie Lian, Xiaopeng Lin, Zhaolong Shen, Yuliang Wei, Haishan Liu, Changti Wu, Hang Yuan, Bailing Wang, Cong Huang, et al. 3d-mix for vla: A plug-and-play module for integrating vggt-based 3d information into vision-language-action models. arXiv preprint arXiv:2603.24393, 2026.

Jian Zhang, Shijie Zhou, Bangya Liu, Achuta Kadambi, and Zhiwen Fan. Spatialstack: Layered geometry-language fusion for 3d vlm spatial reasoning. arXiv preprint arXiv:2603.27437, 2026.

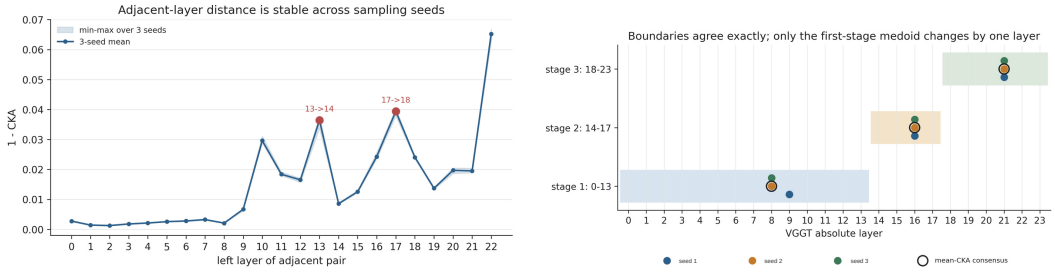
A Implementation and Experimental Details

A.1 GFM Source Selection

For each of three random seeds, we sample 256 frames from 256 distinct episodes, with 64 frames per LIBERO suite, 6–7 frames per task across all 40 tasks, and two camera views per frame. After removing one camera token and four register tokens, we mean-pool patch tokens from all 24 frozen VGGT aggregator layers. For centered feature matrices X_i and X_j,

$$\mathrm{CKA}(i,j) = \frac{\|X_i^\top X_j\|_F^2}{\|X_i^\top X_i\|_F \, \|X_j^\top X_j\|_F}. \tag{7}$$

Adjacent distance $d_i = 1 - \mathrm{CKA}(i, i+1)$ defines three contiguous stages of at least three layers.



(a) Adjacent-layer distance. **(b)** Per-seed medoids and consensus.

Figure 9: **Stratified CKA stability.** Three independently sampled, suite/task-balanced manifests recover identical stage boundaries; only the first-stage medoid shifts from absolute layer 8 to 9 in one seed.

All three seeds give cuts $13 \to 14$ and $17 \to 18$, hence stages 0–13, 14–17, and 18–23. The medoids of the equally averaged CKA matrices are $(8, 16, 21)$, corresponding to negative indices $(-16, -8, -3)$. One seed chooses layer 9 rather than 8 for the first stage; the other medoids and both boundaries agree exactly.

A.2 Action-Conditioned Target Selection

For $\pi_{0.5}$, ActSelect's target selection starts from the pretrained model without prior LIBERO fine-tuning. The VLA and VGGT remain frozen. The stratified probe-training and scoring sets contain 800 and 200 frames, respectively, with 20 and 5 frames per task across all 40 LIBERO tasks. Each frame comes from a distinct episode, with no episode overlap between the two sets. Five independently initialized probes share these data splits.

Each probe contains three temporary projectors, one per GFM source, shared across candidate VLA target layers. The projectors use a hidden dimension of 256 and small nonzero output-layer initialization to enable gate-gradient scoring. The probes are trained for 400 steps using AdamW. At

each step, three target layers are sampled uniformly without replacement from the 18 VLA decoder blocks (indices 0–17), sorted, and paired in order with the three GFM sources. The same three projectors are updated throughout this process.

After training, the projectors are frozen. All source–target pairs are scored using negative action-loss gradients with respect to their gates, evaluated with all gates set to zero. We evaluate 100 scoring batches of size two at flow times $\{0.25, 0.5, 0.75, 1\}$, yielding 400 batch–flow-time evaluations per probe. Scores are centered over candidate layers for each source, standardized over sample–layer entries, and aggregated across probes with an LCB coefficient of one. All $\binom{18}{3} = 816$ monotonic target triples are ranked by their summed source–target LCB scores, selecting $(1, 7, 9)$. Selection stability is assessed using 1000 bootstrap resamples.

For joint action-loss verification, the five highest-ranked mappings reuse the frozen probe projectors on the same 10 held-out scoring batches, with matched flow times and noise. Each residual is rescaled so that its RMS is a prescribed fraction r of the corresponding base-feature RMS. We measure

$$\Delta\mathcal{L}_{\text{act}}(r) = \mathcal{L}_{\text{act}}^{\text{off}} - \mathcal{L}_{\text{act}}^{(r)}, \qquad r \in \{0.5, 1, 2, 5\}\%, \tag{8}$$

where $\mathcal{L}_{\text{act}}^{\text{off}}$ denotes the action loss with residuals disabled, and $\mathcal{L}_{\text{act}}^{(r)}$ denotes the action loss with residuals rescaled to ratio r. Positive values indicate reduced action loss. Appendix B reports the ranking, bootstrap stability, and verification results. The temporary probes are discarded before two-stage fusion training, which starts from the same pretrained VLA parameters with zero-initialized projector output layers.

GR00T N1.6 source–target mapping. For GR00T N1.6, ActSelect selects VLA targets using the native flow-matching loss and searches all 16 decoder layers of the Eagle vision–language backbone (indices 0–15). Flow-time sampling, probe-training settings, and LCB aggregation follow the $\pi_{0.5}$ protocol above. The selected VGGT source layers $(-16, -8, -3)$ map to target layers $(4, 11, 15)$, respectively.

A.3 Two-Stage Safe-Add Training

Each geometry projector uses a two-layer MLP with GELU activation, with an intermediate hidden dimension of 4096 for $\pi_{0.5}$ and 2048 for GR00T N1.6. The output dimension is 2048 for both backbones, matching the hidden size of the target VLA layers.

Stage I jointly fine-tunes the VLA and geometry projectors for 25k steps using the native action loss, with $\alpha = 1$ and no residual penalty. The VGGT encoder remains frozen throughout both stages. The branch reference τ_k is the 95th percentile of batch-mean ratios logged during the final 5k Stage-I steps. At each logging step t, the mean is computed over the set $\mathcal{V}_t$ of all valid sample–camera pairs in the batch:

$$\bar{\rho}_k(t) = \frac{1}{|\mathcal{V}_t|} \sum_{(b,c)\in\mathcal{V}_t} \rho_{kbc}(t), \qquad \tau_k = Q_{0.95}(\{\bar{\rho}_k(t) : t \in \mathcal{T}_{\log}\}),$$

where $\mathcal{T}_{\log}$ contains the logging steps in this window. The same τ_k is applied to individual residual ratios across all valid camera views. It provides a soft reference calibrated from batch means, rather than a sample-level 95th-percentile cutoff; the fraction of penalized samples is therefore not fixed at 5%. Stage II resumes model and Adam states for 5k steps with $\lambda = 0.01$ and $\alpha \sim \text{Cat}(\{1, 0.5, 0\}; \{0.7, 0.2, 0.1\})$. The same scale is shared across the minibatch, cameras, and branches; the regularization penalty is computed from unscaled residuals and backpropagated even when $\alpha = 0$. The VLA and projectors continue to be jointly trained in Stage II. At inference, $\alpha = 1$ and no auxiliary penalty is evaluated.

A.4 Controlled Ablation Settings

The layer-selection ablations in Table 2 use the same base VLA model, LIBERO training data, initialization rules, optimizer, training budget, and two-stage schedule. The full $L179$ configuration uses the ActSelect-selected target layers $\{1, 7, 9\}$. The $L17$ and $L1$ variants remove the corresponding source–target branches while retaining the remaining pairings. Each random three-layer configuration retains the same three GFM sources and projector architecture, varying only the VLA targets with an order-preserving one-to-one mapping.

All schedules in Table 3(a) use 30k training steps and the same source–target configuration. One-stage training uses the native action loss throughout. The 20k+10k and 25k+5k schedules change only the switch from Stage I to Stage II, with all other training settings held fixed. All four fusion operators in Table 3(b) use one-stage training for 30k steps with only the native action loss, holding the source–target layers, training data, and evaluation settings fixed. These controls support separate comparisons of branch selection, training schedule, and fusion design.

Fusion-operator implementations. For branch k and valid camera c, let $H = H_{\ell_k}^{\mathrm{img},c}$ and $R = R_k^c$ follow Equation (3). Table 3(b) compares alternatives to residual addition using the same selected source–target pairs and projector architecture. Each operator replaces only the corresponding camera's image-token states after block ℓ_k. The additive baseline uses $\alpha = 1$.

Cross-attention fusion. After separate layer normalizations, H provides queries and R provides keys and values to a multi-head attention block. Its output is added to H, followed by a second residual MLP.

Gated fusion. The separately normalized H and R are concatenated and passed through a sigmoid projection to produce a gate for each token and channel. The gate weights the normalized VLA features, while its complement weights the normalized geometry features.

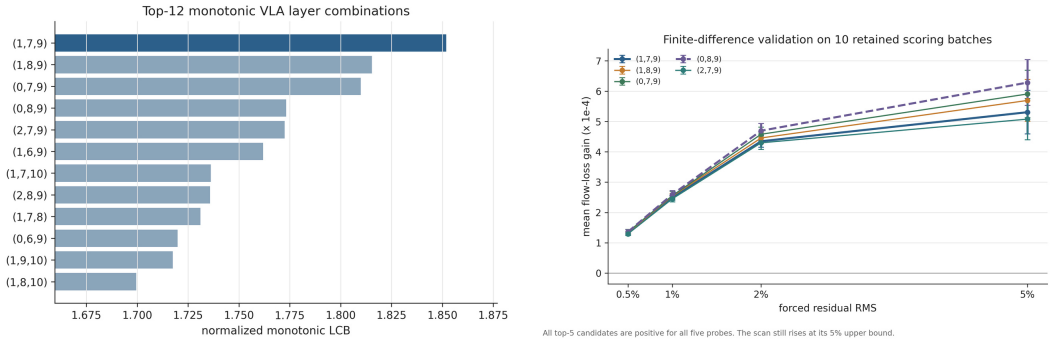
Weighted fusion. At each fusion depth, two learned scalar logits are softmax-normalized to weight H and R. The weights are shared across samples, cameras, tokens, and channels. They are initialized to 0.95 for the VLA features and 0.05 for the geometry features.

B Additional Analysis of Layer Selection

B.1 Candidate Ranking

Table 5: Top monotone ActSelect mappings.

VLA target layers	Normalized LCB	Bootstrap selection frequency
$(1, 7, 9)$	**1.8519**	**99.1%**
$(1, 8, 9)$	1.8153	0.6%
$(0, 7, 9)$	1.8099	0.3%
$(0, 8, 9)$	1.7733	0.0%
$(2, 7, 9)$	1.7724	0.0%



(a) Top-12 monotone mappings. **(b)** Action loss across residual RMS ratios.

Figure 10: **ActSelect ranking and action-loss verification.** The leading LCB mapping is $(1, 7, 9)$. All five leading candidates reduce held-out action loss across the tested RMS ratios. The candidate $(0, 8, 9)$ yields the largest loss reduction on the 10-batch verification subset.

B.2 Selection Stability Across Probes

ActSelect aggregates five independently initialized probes to reduce sensitivity to temporary-projector initialization and short-run optimization. The LCB correction favors source–target pairs with consistently high action-loss utility across probes. Across 1,000 bootstrap resamples, $(1, 7, 9)$

is selected in 99.1% of cases, supporting the stability of the selected mapping under the evaluated calibration protocol.

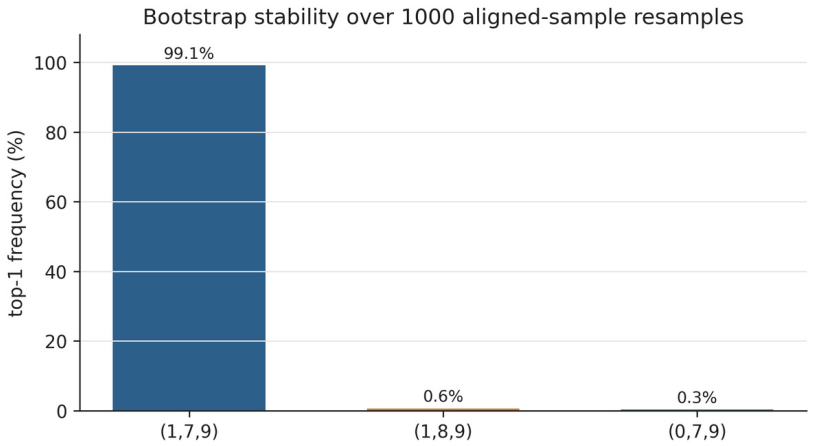


Figure 11: One thousand bootstrap resamples of the five-probe ensemble. $(1, 7, 9)$ is selected in 99.1% of resamples; $(1, 8, 9)$ and $(0, 7, 9)$ account for the remaining 0.6% and 0.3%.

B.3 Action-Loss Verification Across Residual Scales

Table 6: Action-loss reduction for the selected mapping and a comparison candidate.

Mapping	Residual RMS ratio	Mean loss reduction ($\times 10^{-4}$)	Probe SE ($\times 10^{-6}$)	Positive probes
$(1, 7, 9)$	0.5%	+1.32	4.33	5/5
	1%	+2.49	9.33	5/5
	2%	+4.35	19.86	5/5
	5%	+5.31	71.89	5/5
$(0, 8, 9)$	0.5%	+1.36	7.34	5/5
	1%	+2.59	12.70	5/5
	2%	+4.69	23.66	5/5
	5%	+6.28	75.50	5/5

The selected mapping $(1, 7, 9)$ reduces action loss at every tested RMS ratio, with positive gains in all five probes. These results support the action relevance of the selected source–target pairs. The candidate $(0, 8, 9)$ has larger gains on this 10-batch verification subset, whereas $(1, 7, 9)$ leads the LCB ranking computed using all 400 score samples per probe. The full LCB ranking determines the mapping, while finite-scale joint verification checks its action-loss effect. Nonlinear cross-layer interactions can cause the joint ranking to differ from the additive first-order ranking.

For both mappings, the measured gain continues to increase up to the 5% scan boundary. The scan verifies beneficial residual contributions over the tested range; the residual scale used by the adapted policy is learned during two-stage fusion training.

C Representation Retention and Geometric Residual Analysis

C.1 Compared Models and Evaluation Settings

The representation and action-prediction diagnostics compare four checkpoints: $\pi_{0.5}$-FT, SF, ROCKET, and GeoLACE. All four start from the same $\pi_{0.5}$ policy and are trained on LIBERO for 30k steps. GQA is evaluated only after training, with every checkpoint frozen. The $\pi_{0.5}$-FT checkpoint serves as the matched action-training control, while the pretrained $\pi_{0.5}$ checkpoint is the reference used to define retention and acquisition.

Table 7: Model definitions for representation analysis and GQA evaluation. VGGT is frozen in all three geometry-augmented variants.

Model	LIBERO objective	Geometry interface
$\pi_{0.5}$-FT	Native action loss	None; matched action fine-tuning.
SF	Action loss $+ 0.5L_{\text{align}}$	VLA L12 image hidden aligned to the final VGGT hidden through a trainable projector.
ROCKET	Action loss $+ 0.5L_{\text{align}}$	Ten VGGT–VLA layer pairs aligned through a shared projector.
GeoLACE	Native action loss + Stage-II Safe-Add penalty	VGGT sources injected as independent residuals at L1/L7/L9 in this analysis.

The action-only baseline controls for adaptation to LIBERO. Comparing SF, ROCKET, and GeoLACE with this baseline evaluates how their geometry-augmentation mechanisms affect representations and vision–language capabilities.

C.2 Representation and Action-Prediction Changes

The representation study uses the same 512 LIBERO observations for all models, balanced across 40 tasks and four suites. We extract image, language, and action tokens at VLM residual indices L0–L18. Relative representation difference (RRD) measures scale-normalized paired activation change; cross-layer CKA is computed directly from VLA hidden states.

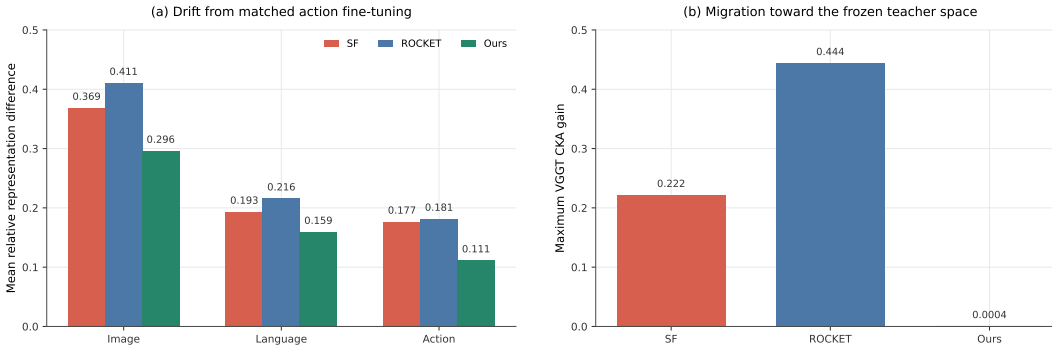


Figure 12: **Representation changes under geometric augmentation.** GeoLACE has the least representation drift and VGGT CKA gain, indicating better preservation of action-adapted VLA features.

We also compare the action velocity-field predictions. After normalizing each action dimension by its dataset standard deviation, both SF and ROCKET show greater drift from $\pi_{0.5}$-FT than GeoLACE at every evaluated flow time (Table 8).

Table 8: Normalized action-vector-field drift from $\pi_{0.5}$-FT.

Method / flow time	0.25	0.50	0.75	1.00
SF	0.384	0.257	0.183	0.141
ROCKET	0.378	0.422	0.373	0.299
GeoLACE	0.177	0.144	0.115	0.089

Together, the representation and action-prediction comparisons show that GeoLACE remains closer to the action-only baseline than both SF and ROCKET, supporting geometric augmentation with less disruption to the adapted policy.

C.3 GQA Evaluation and Visual-Input Controls

We use a fixed, image-disjoint evaluation split from balanced GQA 1.2 (Hudson & Manning, 2019). The evaluation set contains 4,000 questions over 1,864 validation images, with at most three ques-

tions per image. The questions cover yes/no, color, material, position, and category answers. Category candidates are selected from answers occurring at least 30 times in the balanced training partition, without using evaluation labels.

Each GQA image is provided to both the VLA SigLIP encoder and the frozen VGGT encoder through the base-camera input. The wrist-camera slots are masked, and the proprioceptive state is set to zero. Each model receives the question with its standard beginning-of-sequence marker and a trailing newline, without an answer, family label, task instruction, or few-shot example. Inputs and candidate ordering are identical across checkpoints.

Answer scoring and frozen evaluation. We distinguish first-token candidate scoring from full-answer candidate scoring. For an image x, question q, and candidate answer $a = (a_1, \ldots, a_m)$ in its answer family $\mathcal{A}_f$, first-token scoring selects $\hat{a}_{\text{first}} = \arg\max_{a \in \mathcal{A}_f} \ell(a_1 \mid x, q)$, where ℓ denotes a vocabulary logit. Candidate first tokens are unique within each family. Full-answer scoring evaluates the complete candidate sequence:

$$s(a) = \sum_{i=1}^{m} \log p(a_i \mid x, q, a_{<i}), \qquad \hat{a}_{\text{full}} = \arg\max_{a \in \mathcal{A}_f} s(a). \tag{9}$$

Each conditional probability is normalized over the full vocabulary. EOS is excluded, and no length normalization is applied. Replacing the first-token log-probability in Equation (9) with its raw logit adds the candidate-independent constant $\log Z(x, q)$, where $Z(x, q) = \sum_{v \in \mathcal{V}} \exp \ell(v \mid x, q)$ and $\mathcal{V}$ is the vocabulary. The two expressions therefore yield identical candidate rankings for a fixed model and input; we use the uniformly normalized form for clarity. Full-answer accuracy compares $\hat{a}_{\text{full}}$ with the ground-truth candidate and is distinct from free-generation exact match. A separate fixed, family-stratified subset of 500 questions checks free generation with at most five answer tokens using normalized exact match. No GQA example updates the checkpoint, and no VQA head, linear/MLP probe, or LoRA adapter is trained for this evaluation.

Retention and acquisition in Figure 6(a) use first-token scoring for both the pretrained reference and the evaluated model. Let $\mathcal{D}_+$ and $\mathcal{D}_-$ denote questions answered correctly and incorrectly by the reference under this scoring rule. Retention and acquisition are the evaluated model's first-token accuracies on $\mathcal{D}_+$ and $\mathcal{D}_-$, respectively. Thus

$$\text{Acc}_{\text{first}} = p_{\text{ref,first}} \, \text{Retention} + (1 - p_{\text{ref,first}}) \, \text{Acquisition},$$

where $p_{\text{ref,first}} = |\mathcal{D}_+|/(|\mathcal{D}_+| + |\mathcal{D}_-|)$. This identity concerns first-token accuracy, whereas Figure 6(b) reports full-answer accuracy under Equation (9). The action-only $\pi_{0.5}$-FT model provides the matched adaptation baseline for the reported differences.

Visual-input controls. The visual-input controls and fixed-trunk residual interventions use first-token candidate scoring. Each question is evaluated with the correct image, a black image, a randomly shuffled image, and an answer-matched shuffled image. The accuracy benefit of correctly paired visual input is $\Delta_{\text{visual}} = \text{Acc}(\text{correct}) - \text{Acc}(\text{shuffled})$.

Quantitative results and uncertainty. The 95% confidence intervals use 2,000 paired image-cluster bootstrap resamples, jointly resampling all questions associated with each image. They quantify test-image sampling uncertainty, not variation across training seeds. Under first-token scoring, SF reduces retention by 3.05 percentage points relative to $\pi_{0.5}$-FT; the 95% confidence interval for this SF-minus-$\pi_{0.5}$-FT difference is $[-4.47, -1.71]$ percentage points. GeoLACE improves retention by 3.84 percentage points over SF while maintaining an acquisition gain over $\pi_{0.5}$-FT, and achieves 62.80% frozen-model full-answer accuracy (Figure 6(a,b)). Under the visual-input controls, GeoLACE achieves a correct-versus-shuffled accuracy advantage of 23.40 percentage points (Figure 13).

Probability-based diagnostics use first-token logits normalized over the candidates in each question family, rather than full-answer scores:

$$P_{\text{first}}(a \mid x, q, \mathcal{A}_f) = \frac{\exp \ell(a_1 \mid x, q)}{\sum_{b \in \mathcal{A}_f} \exp \ell(b_1 \mid x, q)}. \tag{10}$$

Table 9 reports negative log-likelihood (NLL), the Brier score, Jensen–Shannon (JS) divergence from the pretrained reference, and top-1 prediction agreement with that reference. NLL is the negative log-probability of the correct candidate; Brier is the sum of squared errors against the one-hot

label, without division by the number of candidates. JS uses natural logarithms. These metrics are computed per question and averaged over the evaluation set. The complete JS values underlying Figure 6(c) are included in this table. The GQA evaluation uses frozen model parameters throughout.

Table 9: GQA probability and reference-agreement diagnostics based on family-restricted first-token distributions. Lower is better for NLL, Brier, and JS; higher is better for agreement.

Model	NLL	Brier	JS to reference	Top-1 agreement
$\pi_{0.5}$-FT	1.4131	0.5215	0.0304	0.8175
SF	1.4414	0.5432	0.0437	0.7758
ROCKET	1.4615	0.5524	0.0403	0.7895
GeoLACE	**1.4013**	0.5238	0.0325	**0.8195**

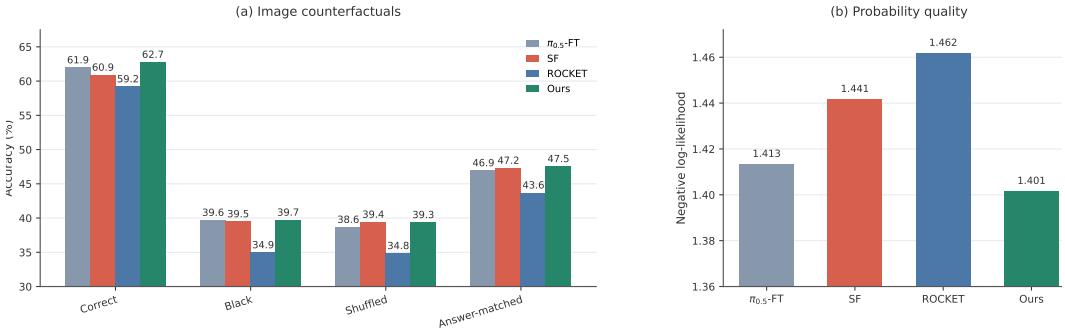


Figure 13: **GQA input controls and probability quality.** Compared with SF, GeoLACE shows a larger accuracy benefit from correctly paired images and achieves lower NLL than both SF and ROCKET.

C.4 Residual Subspace and Rank Analysis

Diagnostic data and residual magnitude. These diagnostics use 500 programmatically constructed paired images with cached 2,048-dimensional activations. Semantic pairs vary color, material, or category while preserving layout; relation pairs vary left/right, front/behind, or inside/touching while preserving object semantics. At each injection site we record the base activation h_{base}, the added geometric residual r_{vggt}, and $h_{\text{fused}} = h_{\text{base}} + r_{\text{vggt}}$. The reported magnitude is the global-pooled residual-to-base RMS ratio,

$$\rho_\ell = \frac{\text{RMS}(r_{\text{vggt},\ell})}{\text{RMS}(h_{\text{base},\ell})}. \tag{11}$$

The denominator is the base activation at the same site before residual addition. Expressed as percentages, the three layerwise measurements range from 1.03% to 2.40% in this diagnostic experiment; this is not a confidence interval or a range over training seeds. Effective-rank summaries additionally average over five pools: global, objects A/B/C, and container.

Linear-readout protocol. These geometric diagnostics are separate from the frozen GQA evaluation. They share the same activation cache and fixed data split. We first fit linear probes on fused features in the training split. We then use their semantic and relation discriminant directions to project components out of the residual; the original probe remains frozen during the intervention. The conditional-information comparison uses the same linear-readout training budget for base, residual, their sum, and their concatenation, and reports balanced accuracy. The rank intervention applies SVD truncation to residual activations at inference and reuses the original frozen fused-feature probe.



Q1 (yes/no). Is the man holding a racket?
$\pi_{0.5}$-FT, GeoLACE: yes.
SF, ROCKET: no.



Q2 (color). Which color is the helmet at the top of the picture?
$\pi_{0.5}$-FT, GeoLACE: black.
SF, ROCKET: red.



Q3 (material). What is the bench made of?
$\pi_{0.5}$-FT, GeoLACE: concrete.
SF, ROCKET: wood.



Q4 (position). Is the white toilet to the left or to the right of the towel?
$\pi_{0.5}$-FT, GeoLACE: right.
SF, ROCKET: left.



Q5 (category). What animal is shown in the image?
$\pi_{0.5}$-FT, GeoLACE: cat.
SF, ROCKET: dog.

Figure 14: **Qualitative GQA comparisons (Q1–Q5).** The examples cover yes/no, color, material, position, and category questions. $\pi_{0.5}$-FT and GeoLACE answer correctly in all five cases, while SF and ROCKET answer incorrectly.

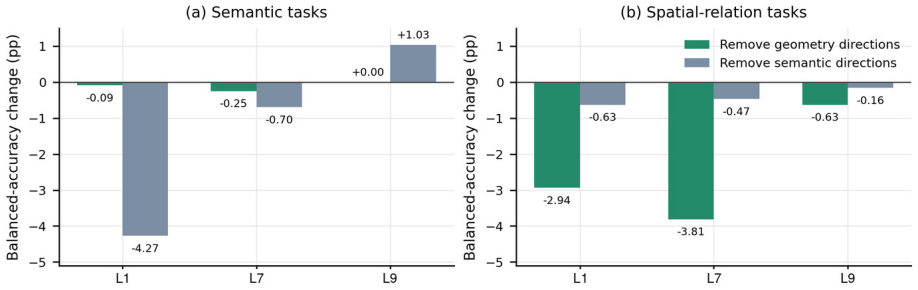


Figure 15: **Probe-relative residual subspace deletion.** At L1/L7, deleting relation directions harms relation tasks more than semantic tasks; deleting semantic directions shows the converse pattern. The directions are defined by linear probes on the diagnostic tasks.

On 500 programmatically constructed paired scenes, frozen linear probes define semantic and spatial-relation directions. We delete only the corresponding component of the residual while leaving the base and readout unchanged. L1/L7 show the expected double selectivity: removing spatial-relation directions preferentially harms relation tasks, whereas removing semantic directions preferentially harms semantic tasks. The effect is most pronounced at L1/L7.

Finally, matched linear readouts show that the base retains the dominant unique information at L9. After conditioning on the residual, the base contributes 28.5/16.4 balanced-accuracy points on semantic/relation tasks; after conditioning on the base, the residual contributes 0.2/0.8 points. Inference-time SVD further shows that most usable relation gain is concentrated in a few residual directions, with a non-monotonic response at L1. This analysis evaluates low-rank approximations at inference while keeping the trained model fixed.

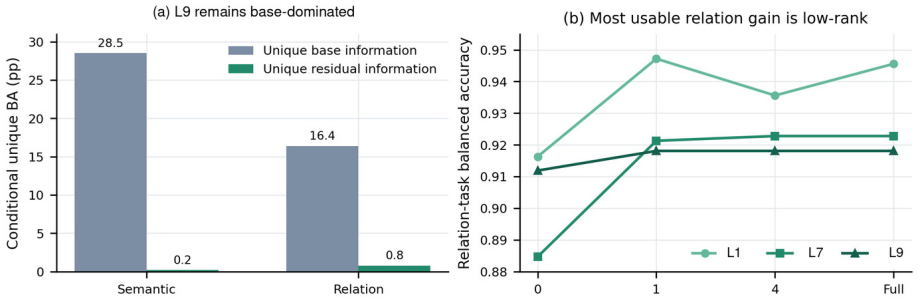


Figure 16: **Base dominance and residual bottleneck.** Conditional unique information remains concentrated in the base, while much of the residual's relation-task effect survives a low-rank inference-time approximation.

D Real-World Experimental Details

D.1 Robot Platforms and Policy Interfaces

Setting A (ALOHA-style). Setting A has an approximately 40×30 cm tabletop working area (Figure 8(a)). The platform in Figure 7(a)–(d) uses two ARX R5 arms, one Intel RealSense L515 head camera, and two D405 wrist cameras. Three 224×224 RGB views and 14-dimensional proprioception condition 50-step bimanual action chunks. During evaluation, the low-level controller publishes action targets at 30 Hz and executes the first 15 steps of each predicted chunk.

Setting B (custom dual-arm robot). Setting B has an approximately 140×60 cm tabletop working area (Figure 8(b)). The platform in Figure 7(e) comprises two serial manipulators, each equipped with an Inspire-Robots RH56E2 five-fingered dexterous hand. Visual input comprises RGB images from a head-mounted Orbbec Femto Mega camera and two wrist-mounted Intel RealSense D405 cameras. The policy receives a 20-dimensional state input, comprising seven arm-state variables and three hand-state variables per side. The policy predicts 50-step action chunks with 20 effective action dimensions, ordered as left arm (7), left hand (3), right arm (7), and right hand (3); the model internally pads this representation to 32 dimensions.

For cable organization, the controller dispatches all 50 right-side action steps from each chunk and waits for the scheduled trajectory duration before requesting the next chunk. Additional interpolation points are inserted when needed to maintain continuity between trajectories.

D.2 Tasks and Success Criteria

The real-world task set includes starfruit handover, item packing, cable organization, rag pickup, and phone placement. The task descriptions are as follows.

Starfruit handover. The robot grasps the starfruit with one arm, hands it over to the other arm, and then places it in the basket.

Rag pickup. The robot grasps the rag with one arm, and then places it in the basket.

Item packing. The robot first uses both arms to pull open the backpack, then picks up all the items from the table and places them into the backpack.

Phone placement. The robot grasps the phone, transports it to the designated stand, aligns it with the stand, and releases it after stable placement.

Cable organization. The robot grasps the flexible cable, lifts and transports it to the foam tray, aligns it with the designated compartment, and releases it entirely inside the compartment.

D.3 Training and Evaluation Protocol

For each of the five real-world tasks, we use 50 demonstration trajectories collected through bimanual teleoperation on the corresponding platform. The reproduced $\pi_{0.5}$ baseline and GeoLACE are fine-tuned on the same demonstrations and evaluated under matched conditions and initial-state distributions. Before each evaluation trial, object positions and orientations are randomized. Each policy is evaluated over 20 trials per reported task. Table 4 reports successful trials for each task and aggregate success rates for Setting A. Across all five tasks, the overall success rates are 47% for $\pi_{0.5}$ and 72% for GeoLACE, based on 20 trials per task.

D.4 Inference-Efficiency Optimizations

Online geometric encoding provides additional spatial information but makes each synchronous policy request wait for features from the current observations. We reduce repeated encoding and blocking through geometric feature reuse and asynchronous refresh, bringing ordinary-request latency closer to that of the base policy.

Temporal reuse of geometric features. The first request synchronously caches camera-aligned, multi-level VGGT features for fusion with current RGB observations and proprioception. The cache refreshes at set intervals or when input shape, camera validity, or scene appearance changes. Appearance changes are detected using per-camera 8-by-8 average-pooled RGB signatures; a refresh is triggered when the largest mean absolute difference from the cached signatures reaches a preset threshold.

Asynchronous geometric refresh. In Setting A, policy requests are issued at approximately 10 Hz, and VGGT refreshes at 2 Hz. Scheduled refreshes encode a snapshot of current observations in the background while requests use cached features. After completion, the next request adopts the new features, visual signature, and completion timestamp; at most one refresh runs at a time. For visual-change triggers, the system waits for any ongoing update and rechecks the signature, recomputing synchronously only if the threshold is still reached.

Table 4 reports means across four Setting A tasks: 89.6 ms for GeoLACE versus 80.2 ms for the base policy (+9.4 ms). The averages cover all recorded requests, including the initial request and those during geometric refreshes. GeoLACE takes 101.1 ms on starfruit handover (RTX 4090) and 285.0 ms on cable organization (NVIDIA Thor). During starfruit handover, changes in camera views and occlusions as the fruit passes between arms may trigger additional geometric refreshes. During cable organization, changes in the flexible cable's configuration may have a similar effect.