ROCKET: Residual-Oriented Multi-Layer Alignment for Spatially-Aware Vision-Language-Action Models

Guoheng Sun[1*] **Tingting Du**[2] **Kaixi Feng**[1] **Chenxiang Luo**[3] **Xingguo Ding**[4]
Zheyu Shen[1] **Ziyao Wang**[1] **Yexiao He**[1] **Ang Li**[1†]
[1]University of Maryland, College Park [2]University of Wisconsin, Madison
[3]City University of Hong Kong [4]University of California, Santa Barbara

Abstract

Vision-Language-Action (VLA) models enable instruction-following robotic manipulation, but they are typically pretrained on 2D data and lack 3D spatial understanding. An effective approach is representation alignment, where a strong vision foundation model is used to guide a 2D VLA model. However, existing methods usually apply supervision at only a single layer, failing to fully exploit the rich information distributed across depth; meanwhile, naïve multi-layer alignment can cause gradient interference. We introduce ROCKET, a residual-oriented multi-layer representation alignment framework that formulates multi-layer alignment as aligning one residual stream to another. Concretely, ROCKET employs a shared projector to align multiple layers of the VLA backbone with multiple layers of a powerful 3D vision foundation model via a layer-invariant mapping, which reduces gradient conflicts. We provide both theoretical justification and empirical analyses showing that a shared projector is sufficient and outperforms prior designs, and further propose a Matryoshka-style sparse activation scheme for the shared projector to balance multiple alignment losses. Our experiments show that, combined with a training-free layer selection strategy, ROCKET achieves a 98.5% state-of-the-art success rate on LIBERO and outperforms single-layer alignment under the same training budget. We further demonstrate the superior performance of ROCKET across LIBERO-Plus, RoboTwin, multiple VLA models, and a real-world industrial warehouse picking task. The code and model weights are available at `https://github.com/CASE-Lab-UMD/ROCKET-VLA`.

1 Introduction

Robotic manipulation requires models to integrate semantic intent with precise spatial reasoning about objects, contacts, and constraints in the physical world. Recent vision-language-action (VLA) models achieve strong instruction-following by leveraging large vision-language backbones, yet these backbones are predominantly pretrained on 2D imagery and often fail to produce representations with stable 3D structure Zitkovich et al. [2023], Kim et al. [2024], Li et al. [2025a]. As a result, VLAs may succeed on familiar scenarios but generalize poorly when manipulation depends on accurate geometry, viewpoint changes, or spatial relations Shridhar et al. [2022], Huang et al. [2023].

To bridge this gap, prior work has pursued three main strategies to equip VLAs with spatial reasoning: (1) augmenting inputs with explicit 3D signals such as depth maps or point clouds Shridhar et al. [2023]; (2) recovering 3D structure from 2D observations via depth estimation Ranftl et al. [2021]; and (3) implicitly aligning intermediate VLA representations to spatial features produced by pretrained 3D foundation models Li et al. [2025a]. Compared with the first two approaches, which rely on additional

*ghsun@umd.edu
†Corresponding author: angliece@umd.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

sensors or external depth estimators, implicit alignment is typically more inference-efficient and easier to scale, making it a particularly practical direction for improving spatial grounding in VLAs.

In *representation alignment* for improving spatial reasoning, a common practice is to supervise a VLA model's visual-token representations to match those of a strong 3D vision foundation model. Most existing methods apply alignment at a single layer Yu et al. [2024], Huang et al. [2025b], Li et al. [2025a], Wang et al. [2024]. However, prior work has found that performance is highly sensitive to the chosen layer, and the optimal layer often varies across tasks and data distributions (as shown in Table 1). Since it is difficult to predict which layer contains the most appropriate visual representations *a priori*, single-layer alignment typically relies on inefficient post-hoc search. Meanwhile, representations in deep residual networks evolve with depth and encode information at multiple levels Skean et al. [2025], Lee et al. [2025], and prior feature distillation studies similarly suggest benefits from leveraging teacher signals across depths Sun et al. [2019], Chen et al. [2021], Gong et al. [2025]. A natural alternative is multi-layer alignment, which can exploit hierarchical spatial cues from shallow to deep layers. However, in the VLA setting, directly extending classic distillation by aligning multiple depths with *independent* projectors per layer often backfires: separate projectors tend to learn inconsistent mappings between representation spaces, leading to severe gradient interference and ultimately causing performance collapse.

To address gradient interference, we propose ROCKET, a stable and high-performance multi-layer alignment strategy that formulates multi-layer alignment as aligning one residual stream to another. From a residual-dynamics perspective, we attribute the failure of naive multi-projector designs to gradient conflicts, and show that using a shared projector promotes gradient coherence. Concretely, ROCKET learns a cross-layer shared mapping via a lightweight projector that aligns residual streams from multiple layers to a target residual stream. This shared alignment helps reduce gradient interference, enabling fast convergence while continuing to improve late-stage performance. We further observe that, in multi-layer alignment, shallower layers converge more easily; to prevent shallow-layer alignment from dominating the shared projector, ROCKET introduces a Matryoshka-style Kusupati et al. [2022] sparse activation scheme in which deeper layers activate more projector parameters. Such a scheme balances alignment losses across depths, encouraging shallow layers to quickly capture common local cues while allowing deeper layers to further refine global information. We also analyze layer selection and show that a simple training-free rule yields stable gains. Experiments across multiple models and datasets demonstrate that ROCKET improves spatial reasoning in VLA models and consistently outperforms the single-layer alignment paradigm. Our contributions are summarized as follows:

- We propose ROCKET, a multi-layer alignment framework that injects 3D spatial reasoning into 2D-pretrained VLA models, featuring a layer-invariant shared projector to avoid gradient conflicts and a Matryoshka-style sparse activation scheme to balance alignment losses across depths.
- We develop theoretical and empirical analyses that explain when and why prior multi-layer alignment fails due to gradient interference, and show that a single shared projector is both sufficient and consistently superior to layer-wise projectors.
- By leveraging complementary spatial cues across depths, ROCKET achieves state-of-the-art performance on LIBERO and generalizes across multiple datasets and VLA backbones, and demonstrates consistent improvements on a real-world industrial warehouse picking task.
- ROCKET is compute-efficient and converges quickly, reaching state-of-the-art (SOTA) performance with $\sim$4% of the training compute required by prior SOTA methods, while remaining effective in data-limited regimes relevant to embodied settings.

2 Related Work

Vision-Language-Action Models and Spatial Grounding. VLA models integrate visual perception and language understanding to generate embodied robot actions. Representative works include **OpenVLA** Kim et al. [2024] and **OpenVLA-OFT** Kim et al. [2025] built upon Prismatic-7B Karamcheti et al. [2024], as well as **PI0** Black et al. [2024] and **PI0.5** Black et al. [2025] based on PaliGemma Beyer et al. [2024], which primarily rely on 2D Vision-Language Models (VLMs). These models often lack geometric understanding of the 3D world. To address this, several works augment VLA models with 3D information, including point-cloud inputs Bhat et al. [2025], Sun et al. [2025] or depth-aware encodings Qu et al. [2025], and concurrent works further integrate VGGT-based 3D features Yu et al. [2026], Li et al. [2026], Xiao et al. [2026]. Meanwhile, recent 3D foundation models

such as **VGGT** Wang et al. [2025a], **Depth Anything** Lin et al. [2025a], and π^3 Wang et al. [2025b] provide strong spatial representations that can offer rich supervision signals for 2D vision backbones.

Representation Alignment for Spatial Supervision. Representation alignment (feature distillation) is widely used to transfer spatial knowledge from pretrained teachers to VLM/VLA models. Most prior work aligns student visual-token features from one layer to a fixed teacher layer. For example, **Spatial Forcing (SF)** Li et al. [2025a] and **GLaD** Guo et al. [2025] align *deep* vision representations to VGGT features; **REPA** Yu et al. [2024] aligns *shallow* diffusion-model Peebles and Xie [2022] representations to DINOv2-Large Oquab et al. [2023]; and **3DRS** Huang et al. [2025b] supervises the *final-layer* visual tokens using VGGT. Although effective, these methods typically supervise a single layer and do not fully exploit the representations across depth. Moreover, as shown in Table 1, the choice of which layer to align is often inconsistent and depends on post-hoc empirical results.

Table 1: Comparison of representation alignment methods for spatial supervision. OpenVLA – **24 / 32**: 32-layer backbone with 24th-layer features for alignment.

Method	Student	Teacher
SF	OpenVLA – **24 / 32**	VGGT – 24 / 24
REPA	DiT – **8 / 24**	DINOv2 – 24 / 24
3DRS	LLaVA – **32 / 32**	VGGT – 24 / 24

Multi-layer Distillation and Optimization Challenges. Early studies on knowledge distillation have shown that supervising multiple intermediate layers can lead to more effective knowledge transfer. Methods such as **Patient Knowledge Distillation** Sun et al. [2019], **ReviewKD** Chen et al. [2021], and recent feature-dynamics-based approaches Gong et al. [2025] demonstrate the benefits of multi-layer alignment in deep networks. To enable supervision at intermediate layers, prior methods commonly attach layer-wise *projectors* Chen et al. [2022], Miles and Mikolajczyk [2024], Miles et al. [2024] that reconcile mismatched feature spaces before computing alignment losses. However, extending multi-layer distillation to VLA models introduces additional challenges: gradient interference can arise among the separate projectors, which can degrade alignment quality.

3 Theoretical Framework: Residual Dynamics and Gradient Coherence

3.1 Background

VLA as student and manipulation objective. For ease of understanding, we view representation alignment from the perspective of feature knowledge distillation. We consider a pretrained VLA policy as the *student* model. Given an observation (e.g., multi-view images) I and an instruction P, the student induces an action policy π_θ. One possible instantiation is an autoregressive policy Kim et al. [2024] that predicts an action sequence $A = \{a_t\}_{t=1}^{K}$: $a_t \sim \pi_\theta(\cdot \mid I, P, a_{<t})$, while our method is agnostic to the specific action parameterization (e.g., continuous regression or discretized action tokens). The student is trained with an action learning loss: $\mathcal{L}_{\text{action}} = \mathcal{L}(A, A^{gt})$, where A^{gt} denotes the ground-truth action sequence from demonstrations.

3D vision foundation model as teacher. We assume access to a *frozen* 3D-aware vision foundation model as the *teacher*, which provides geometry-grounded *patch token* representations. Such models (e.g., VGGT Wang et al. [2025a], π^3 Wang et al. [2025b], Depth Anything 3 Lin et al. [2025a]) learn spatially consistent features from monocular or multi-view images and can serve as strong supervision signals for injecting 3D cues into a VLA student's visual representations.

Single-layer representation alignment. Single-layer alignment applies supervision on the *visual tokens* of a student model by matching them to a teacher's intermediate representations at a selected depth to improve spatial awareness. There has been a line of prior work adopting single-layer intermediate feature matching, which is simple yet effective Li et al. [2025a], Guo et al. [2025]. Let $h_s^S(\theta) \in \mathbb{R}^{d_S}$ and $h_\tau^T \in \mathbb{R}^{d_T}$ denote the student/teacher vision representations at student layer s and teacher layer τ. SF aligns one student layer $s^\star$ to one teacher layer $\tau^\star$ (i.e., $\mathcal{S} = \{s^\star\}$, $\mathcal{T} = \{\tau^\star\}$) via a lightweight projector p_ϕ and optimizes: $\mathcal{L}_{\text{align}} = \ell\big(p_\phi\big(h_{s^\star}^S(\theta)\big)\,, h_{\tau^\star}^T\big)$, where $\ell(\cdot,\cdot)$ can be any similarity-based loss. Accordingly, most representation-alignment methods optimize: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} + \lambda\,\mathcal{L}_{\text{align}}$, where λ balances task learning and representation alignment.

3.2 The Gradient-Conflict of Multi-Layer Alignment

From single-layer to multi-layer alignment: richer cues, but no gains in practice. Despite its effectiveness, single-layer alignment has limitations: (i) *single-point* alignment requires post-hoc layer selection, and (ii) it fails to use complementary cross-layer signals. A natural extension is to align *multiple* layers simultaneously, since spatial cues may be distributed across depth and multi-layer distillation is effective in other settings Skean et al. [2025], Lee et al. [2025]. Concretely, given ordered layer lists $\mathcal{S} = \{s_i\}_{i=1}^N$ (student) and $\mathcal{T} = \{\tau_i\}_{i=1}^N$ (teacher), previous multi-layer alignment methods use layer-specific projectors $\{p_{\phi_i}\}$: $\mathcal{L}_{\text{align}} = \frac{1}{N}\sum_{i=1}^N \ell\big(p_{\phi_i}\big(h_{s_i}^S(\theta)\big),\, h_{\tau_i}^T\big)$. However, this prior-style design Sun et al. [2019], Chen et al. [2021], Gong et al. [2025] *hurts* downstream VLA manipulation performance (as shown in Fig. 3).

Residual-dynamical view: multi-layer alignment should be cone-to-cone, yet learned projectors decouple. To understand why multi-layer alignment fails, we view both student and teacher backbones as residual dynamical systems E [2017], Chang et al. [2017]. Let $h_l \in \mathbb{R}^d$ be the student residual representation at layer l:

$$h_{l+1} \;=\; h_l + F_l(h_l;\theta_l)\,, \qquad l = 0,\ldots,L-1, \tag{1}$$

with $\Delta_l := F_l(h_l;\theta_l)$. Empirically, the hidden states within each model tend to converge into the same feature space Gromov et al. [2024], He et al. [2024] (Fig. 4), consistent with the "cone effect" Gao et al. [2019]. This suggests multi-layer alignment should learn a consistent *cone-to-cone* mapping from the student residual stream to the teacher residual stream. Yet, under naive multi-layer alignment, the learned projectors $\{p_{\phi_i}\}$ *diverge* and become nearly orthogonal in parameter space (Please refer to Appendix G), indicating that different depths follow inconsistent alignment paths rather than sharing a coherent mapping.

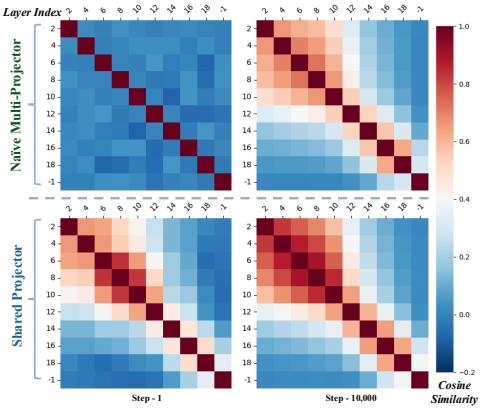


Figure 1: Cosine similarity between gradients induced by different alignment losses in the VLA's shallow layers. **Red** indicates gradient coherence, while **Blue** indicates gradient interference. Detailed results are provided in Fig. 19 and Fig. 20.

Corollary: early-layer updates are a superposition of future local distillation gradients. Why do inconsistent projectors matter for optimization? Our key observation is that in a Pre-LN residual stream, back-propagation through depth is close to identity when residual updates are small, so an early-layer representation receives an *additive* mixture of gradient signals coming from *all* aligned future layers. Formally, Appendix J derives (Corollary 1) that for an early layer i,

$$\nabla_{h_i}\mathcal{L}_{\text{align}} = \sum_{l\in\mathcal{S}:l\geq i} \alpha_l \underbrace{J_{p_{\phi_l}}(h_l)^\top g_l}_{=:\,v_l} + \mathcal{E}_i, g_l \quad := \nabla_{z_l}\ell(z_l,t_l), \qquad z_l = p_{\phi_l}(h_l). \tag{2}$$

where $\mathcal{E}_i$ is a *transport error* controlled by residual-smallness and α denotes the layer-wise weight. Thus, *whether multi-layer alignment helps hinges on whether* $\{v_l\}$ *add constructively or cancel.*

Proposition: Jacobian-induced gradient interference under multiple independent projectors. Let $G \approx \sum_{l\geq i}\alpha_l v_l$ denote the superposed gradient (ignoring $\mathcal{E}_i$). Its squared norm expands as

$$\|G\|_2^2 = \sum_l \alpha_l^2\|v_l\|_2^2 + \sum_{a\neq b}\alpha_a\alpha_b\,\langle v_a, v_b\rangle, \tag{3}$$

where the cross terms determine constructive vs. destructive interference. Crucially, each cross term is a bilinear form on teacher-side error signals:

$$\langle v_a, v_b\rangle = g_a^\top \underbrace{\Big(J_{p_{\phi_a}}(h_a)\,J_{p_{\phi_b}}(h_b)^\top\Big)}_{=:\,\mathcal{M}_{ab}} g_b. \tag{4}$$

With *multiple independent* projectors, $\mathcal{M}_{ab}$ is *not structurally coupled* across (a,b): the interaction matrix $J_{p_{\phi_a}}(h_a)J_{p_{\phi_b}}(h_b)^\top$ need not be symmetric or positive semidefinite (PSD), and can vary

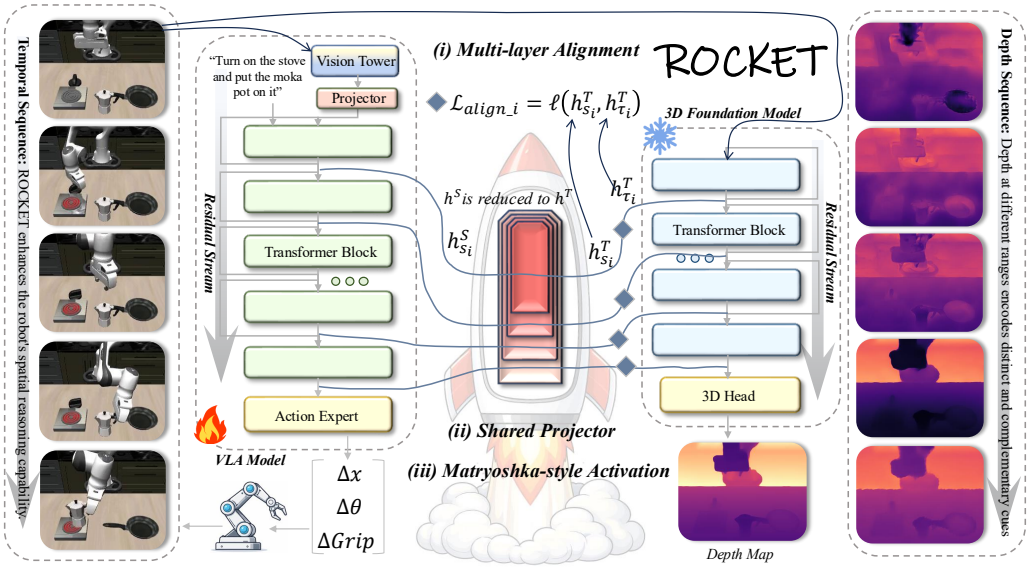


Figure 2: Overview of ROCKET. On the **left**, we select a sample from LIBERO to showcase ROCKET's performance. On the **right**, we use the outputs from VGGT layers $\{1, 3, 9, 15, 21\}$ to directly predict the depth map, demonstrating that different layers contain rich 3D information. Some of the illustrations were generated by GPT-4o Hurst et al. [2024].

arbitrarily across layers. As a result, $\langle v_a, v_b \rangle$ may be weak, sign-unstable, or even destructive. Empirically, we observe that cross-layer alignment gradients remain poorly aligned (small pairwise cosine similarity) throughout training (Shown in the upper part of Fig. 1), consistent with gradient conflict under unconstrained cross terms.

3.3 Motivation

Shared projector yields a structured decomposition and improves gradient coherence in practice. Multi-layer alignment with multiple *independent* projectors often underperforms due to gradient interference across layers. In contrast, if we *share a single projector* across layers ($\phi_l \equiv \phi$), Appendix J shows that although $\mathcal{M}_{ab} = J_{p_\phi}(h_a) J_{p_\phi}(h_b)^\top$ is not necessarily symmetric or PSD, it can be decomposed into a *common PSD reference* plus a controlled deviation:

$$\langle v_a, v_b \rangle_{\text{share}} = g_a^\top \mathcal{M}_{ab} g_b = g_a^\top M g_b + \Delta_{ab}, \qquad M := JJ^\top \succeq 0. \tag{5}$$

Moreover, on the error-signal subspace $\mathcal{G} = \text{span}\{g_l\}$ where M is near-isometric, we obtain the signal-aligned lower bound (Theorem 1):

$$\langle v_a, v_b \rangle_{\text{share}} \;\geq\; c\, g_a^\top g_b \;-\; \eta c \cdot \frac{\|g_a\|_2^2 + \|g_b\|_2^2}{2} \;-\; |\Delta_{ab}| \tag{6}$$

for constants $c > 0$ and $\eta \in [0, 1)$. When $g_a^\top g_b$ is typically positive and the deviation terms are small, sharing biases cross-layer gradients toward constructive interference. We validate this prediction by visualizing gradient angles: with a shared projector, cross-layer gradient cosine similarities increase substantially compared to the independent-projector baseline (Shown in the lower part of Fig. 1), and this improved coherence translates into significant downstream gains on VLA manipulation tasks (Fig. 3). This motivates ROCKET's design choice: *share the projector* to reduce gradient conflict and make multi-layer alignment effective. We note that this analysis explains the observed optimization behavior rather than fully describing the training dynamics: gradient interference is an important contributing factor, while other factors, such as optimization difficulty and supervision imbalance, may also contribute. Full proofs and additional empirical analysis are provided in Appendix J.

4 Methodology

Overview of ROCKET. We propose ROCKET, a multi-layer representation alignment method on the residual stream. As shown in Fig. 2, ROCKET has three key designs:

(i) Multi-layer representation alignment. Single-layer alignment cannot fully leverage the rich semantic information across layers and often relies on post-hoc layer selection. ROCKET aligns multiple layers, and Sec. 7 shows that with a simple layer selection strategy, ROCKET outperforms the baselines.

(ii) Shared projector to reduce gradient conflict. Naive multi-layer alignment may cause gradient conflicts across layers. ROCKET uses a *shared* projector to learn a common mapping between two residual streams.

(iii) Matryoshka-style activation for balancing depths. As indicated by Eq. 2, gradients from multiple alignment losses approximately add up, so shallow layers receive denser supervision. Our experiments also show that shallow layers are easier to align (Please refer to Appendix H for more details). To prevent shallow layers from dominating the shared projector, ROCKET introduces a Matryoshka-style Kusupati et al. [2022] parameter-sparse activation mechanism, which progressively increases the activated parameter fraction from shallow to deep layers.

4.1 Multi-layer Alignment with Shared Projector

We follow the multi-layer alignment setting in Sec. 3.2, with N aligned layer pairs $\{(s_i, \tau_i)\}_{i=1}^{N}$ and the corresponding hidden states $\{(h_{s_i}^S, h_{\tau_i}^T)\}_{i=1}^{N}$. Crucially, we replace the layer-specific projectors $\{p_{\phi_i}\}_{i=1}^{N}$ in Eq. 3.2 with a single shared projector p_Φ, which learns a *shared* student-to-teacher mapping that is consistent across depth. For the detailed layer selection strategy, please refer to Sec.7.

4.2 Matryoshka-style Sparse Activation

We parameterize a single shared projector $p_\Phi : \mathbb{R}^{d_S} \to \mathbb{R}^{d_T}$ with a maximum internal width m. Specifically, we present a two-layer MLP projector: $p_\Phi(x) = W_2\,\sigma(W_1 x), W_1 \in \mathbb{R}^{m\times d_S},\ W_2 \in \mathbb{R}^{d_T\times m}$, where $\sigma(\cdot)$ is a pointwise nonlinearity (e.g., GELU), and $\Phi = \{W_1, W_2\}$.

For each aligned layer s_i, we activate only the *first* m_i hidden channels of the shared projector, where m_i increases monotonically with depth. By default, we use a simple linear width schedule:

$$\rho_i = \rho_{\min} + (\rho_{\max} - \rho_{\min}) \cdot \frac{i-1}{\max(N-1,\,1)}, m_i \;=\; \lceil \rho_i\, m \rceil\,, \tag{7}$$

where $\rho_{\min} = 0.2$ and $\rho_{\max} = 1.0$, m is the maximum hidden width of the shared projector. When $N = 1$, this reduces to $\rho_1 = \rho_{\max}$ (i.e., activating the full projector). Equivalently, define a binary gating vector $g_i \in \{0,1\}^m$ as a prefix mask: $g_i[j] \;=\; \mathbb{I}(j \le m_i)\,, j = 1, \ldots, m$, and apply it to the hidden activation: $p_\Phi^{(i)}(x) = W_2(g_i \odot \sigma(W_1 x))\,$, where $\odot$ denotes element-wise multiplication. This construction yields a *Matryoshka* (nested) family of projectors $\{p_\Phi^{(i)}\}_{i=1}^{N}$ that share parameters, but expose increasing effective capacity for deeper layers.

Training objective. Our final training objective combines the task loss for action prediction with the proposed multi-layer alignment regularizer: $\mathcal{L}_{\texttt{ROCKET}} = \mathcal{L}_{\text{action}} + \lambda\, \frac{1}{N} \sum_{i=1}^{N} \ell\Big(p_\Phi^{(i)}\big(h_{s_i}^S\big)\,,\, h_{\tau_i}^T\Big)\,,$ where $\mathcal{L}_{\text{action}}$ is defined in Eq. (3.1), and λ balances task learning and representation alignment. By default, we use $\ell(a,b) = 1 - \cos(a,b)$ and set $\lambda = 0.5$.

5 Main Result

5.1 Experimental Setup

We evaluate on three benchmarks: LIBERO Liu et al. [2023], LIBERO-PLUS Fei et al. [2025], and ROBOTWIN 2.0 Chen et al. [2025a]. Unless otherwise specified, we fine-tune OPENVLA-7B with LoRA. We also evaluate under full fine-tuning with PI0.5 on LIBERO, and with PI0 on ROBOTWIN 2.0. For all benchmarks, we report task execution success rate (SR, %). Full dataset descriptions and training details are in Appendix A.1.

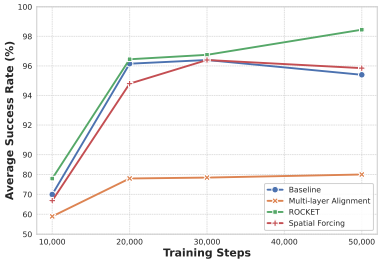


Figure 3: Performance on LIBERO across different training stages. See Table 17 for details.

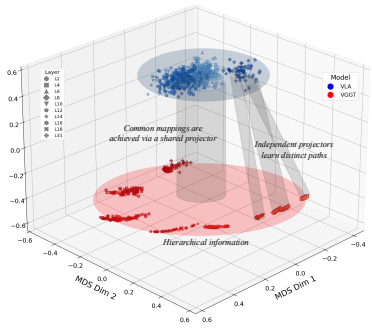


Figure 4: A cone-effect perspective on alignment.

Table 2: Comparisons with SOTA VLA models on the LIBERO benchmark.

Method	Spatial	Object	Goal	Long	Avg.
Basic 2D VLA					
Octo Ghosh 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
WorldVLA Cen et al. [2025]	87.6	96.2	83.4	60.0	81.8
Dita Hou et al. [2025]	84.2	96.3	85.4	63.8	82.4
PI0 Black et al. [2024]	96.8	98.8	95.8	85.2	94.2
InternVLA Chen et al. [2025b]	98.0	99.0	93.8	92.6	95.9
GR00T N1.6 Bjorck et al. [2025]	97.7	98.5	97.5	94.3	97.0
VOTE Lin et al. [2025b]	98.8	**99.8**	97.6	95.6	98.0
TraceVLA Zheng et al. [2024]	84.6	85.2	75.1	54.1	74.8
ThinkAct Huang et al. [2025a]	88.3	91.4	87.1	70.9	84.4
UniVLA Bu et al. [2025]	96.5	96.8	95.6	92.0	95.2
MemoryVLA Shi et al. [2025]	98.4	98.4	96.4	93.4	96.7
CronusVLA Li et al. [2025b]	97.3	99.6	96.9	94.0	97.0
OpenVLA-OFT Kim et al. [2025]	97.6	98.4	97.9	94.5	97.1
3D VLA with explicit 3D inputs					
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	96.1	98.1
3D VLA with representation alignment					
GLaD Guo et al. [2025]	95.0	97.4	94.4	89.4	94.1
Spatial Forcing Li et al. [2025a]	**99.4**	99.6	**98.8**	96.0	**98.5**
ROCKET (Ours)	98.2	**99.8**	**98.8**	**97.0**	**98.5**

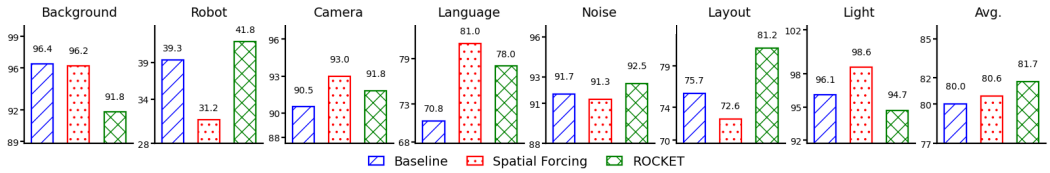


Figure 5: Success rate (%) on LIBERO-Plus. Results are reported under the seven perturbations defined by LIBERO-Plus. Each column shows the average success rate over four task groups: Spatial, Object, Goal, and Long. Detailed results are provided in Appendix I.

5.2 Performance of ROCKET Across Different Settings

LIBERO with OpenVLA-7B. As shown in Table 2, we compare representative approaches on LIBERO, including: (i) standard 2D VLA models, covering both direct fine-tuning of the base VLA and variants that boost performance via additional modules; (ii) 3D VLA methods with explicit 3D inputs, including augmenting inputs with explicit 3D signals and recovering 3D structure via depth estimation; and (iii) two single-layer representation alignment baselines. Results show that ROCKET substantially outperforms prior models. Even compared to Spatial Forcing, ROCKET matches its performance. Since the results of prior methods are taken from their original papers with different training budgets, we further compare them under the same training budget in Table 3, where ROCKET outperforms Spatial Forcing by 2.6% (98.5% vs. 95.9%).

Table 3: Success rate (%) of OpenVLA-7B and PI0.5 with different representation alignment strategies on LIBERO under the same training budget.

Method	Spatial	Object	Goal	Long	Avg.
OpenVLA-7B (LoRA, bs 32, 50k steps)					
Baseline	94.0	97.2	98.0	92.4	95.4
Spatial Forcing	97.8	98.8	96.2	90.6	95.9
ROCKET	**98.2**	**99.8**	**98.8**	**97.0**	**98.5**
PI0.5 (full fine-tuning, bs 64, 30k steps)					
Baseline	96.4	98.2	95.0	82.2	93.0
Spatial Forcing	**97.8**	97.8	94.4	85.8	94.0
ROCKET	96.4	**98.8**	**96.6**	**89.2**	**95.3**

LIBERO with PI0.5. On PI0.5, we compare ROCKET with the direct fine-tuning baseline and Spatial Forcing under the full fine-tuning setting. As reported in Table 3, under the same setting, ROCKET

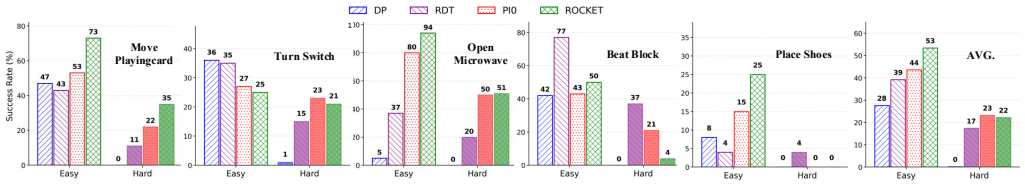


Figure 6: Success rate (%) on RoboTwin 2.0 (ALOHA) under Easy and Hard settings across five bimanual tasks, comparing ROCKET with DP Chi et al. [2025], RDT Liu et al. [2024], and PI0 Black et al. [2024].

improves performance by 2.3%. Overall, this improvement suggests that even for smaller-scale models under full fine-tuning, ROCKET can effectively inject spatial reasoning capability.

LIBERO-Plus with OpenVLA-7B. We further evaluate ROCKET on LIBERO-Plus, a robustness benchmark with seven perturbation dimensions, and find that ROCKET is more robust under challenging spatial perturbations. As shown in Fig. 5, ROCKET achieves the best average success rate (81.7%), outperforming the baseline (80.0%). It also achieves a higher average success rate than Spatial Forcing (80.6%), although it does not outperform Spatial Forcing on every perturbation type. Notably, ROCKET shows the most pronounced gains under *Robot* and *Layout* shifts, which are strongly tied to spatial geometry rather than appearance. These improvements suggest that ROCKET enhances spatial reasoning instead of relying on positional shortcuts.

RoboTwin 2.0 with PI0. For bimanual tasks, we evaluate on RoboTwin 2.0 using ALOHA assets Aldaco et al. [2024]. As shown in Fig. 6, ROCKET achieves a clear advantage in the Easy setting, while in the Hard setting it is only slightly behind the strongest baseline (finetuned).

RoboTwin 2.0 Hard with PI0.5. The models in Fig. 6 are trained only on Clean data, so the Hard setting evaluates out-of-distribution generalization. We further train PI0.5 and ROCKET directly on Hard data (50 demonstrations and 100 evaluation trials per task). As shown in Table 4, with one model per task, ROCKET achieves the highest average success rate (29.2% vs. 27.6% for the baseline and 25.6% for Spatial Forcing), although no method dominates every task. With one policy for all five tasks, ROCKET improves the average success rate from 17.2% to 28.2%.

Table 4: RoboTwin 2.0 Hard SR (%) with PI0.5 trained on Hard data.

Task	One model per task			One model for all	
	Baseline	SF	ROCKET	Baseline	ROCKET
Move Playingcard	61	**84**	63	14	**56**
Turn Switch	**19**	11	17	21	**30**
Open Microwave	27	22	**37**	11	**15**
Beat Block	**31**	11	27	**40**	**40**
Place Shoes	0	0	**2**	0	0
Average	27.6	25.6	**29.2**	17.2	**28.2**

6 Real-World Experiments

We validate ROCKET on a challenging real-robot warehouse parcel picking task that requires precise 3D spatial understanding under industrial conditions. Full experimental details, per-run results, and additional analyses are provided in Appendix B.

Setup. We use a Ufactory xArm 850 with a UFACTORY xArm Gripper G2 (Fig. 7-a,c). The task requires picking deformable, visually identical parcels from loaded containers, with 20 consecutive grasps per round without scene reset. We evaluate on two environments (**Env1 (Fig. 7-a,b)**: wire-mesh container; **Env2 (Fig. 7-c,d)**: box) under nighttime and daytime (OOD) lighting. All methods use identical training data and compute (see Appendix B.1 for full details).

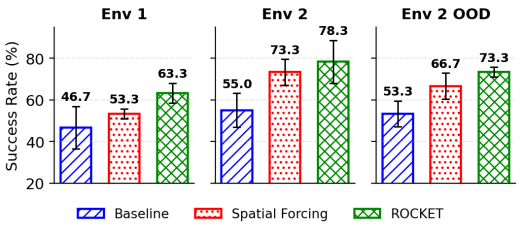
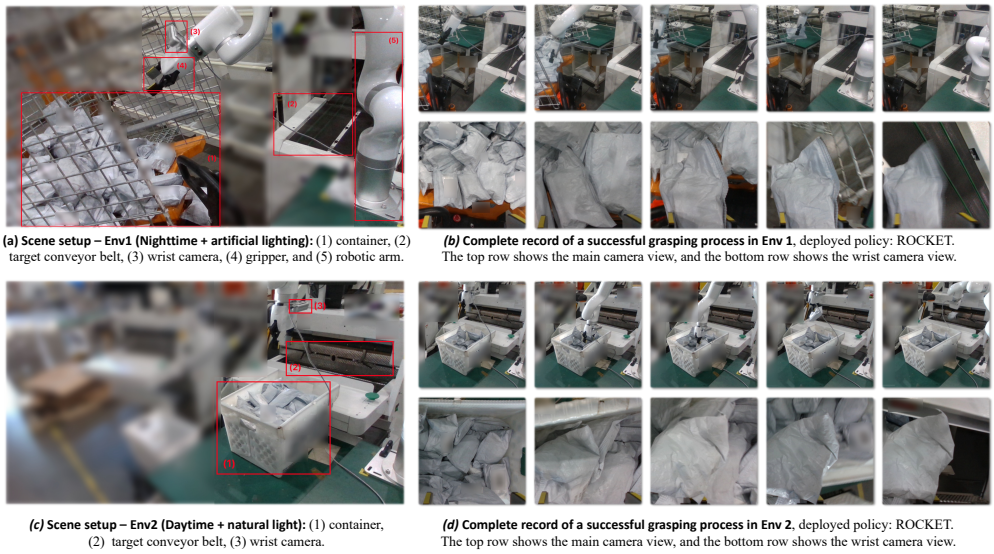


Figure 8: Success rate across real-world settings.



(a) Scene setup – Env1 (Nighttime + artificial lighting): (1) container, (2) target conveyor belt, (3) wrist camera, (4) gripper, and (5) robotic arm.

***(b)* Complete record of a successful grasping process in Env 1**, deployed policy: ROCKET. The top row shows the main camera view, and the bottom row shows the wrist camera view.

***(c)* Scene setup – Env2 (Daytime + natural light):** (1) container, (2) target conveyor belt, (3) wrist camera.

***(d)* Complete record of a successful grasping process in Env 2**, deployed policy: ROCKET. The top row shows the main camera view, and the bottom row shows the wrist camera view.

Figure 7: Real-world setup. **Left**: Env1, wire-mesh container to conveyor belt. **Right**: Env2, box to adjacent slots. Parcels are soft-body, visually identical, and densely stacked.

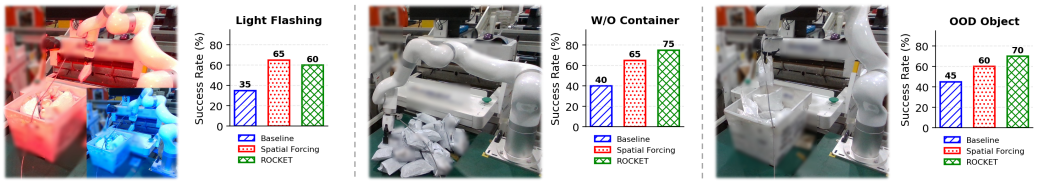


Figure 9: Robustness to environmental perturbations. ROCKET achieves the best overall robustness, with strong generalization to unseen objects and simplified environments.

Main Results. As shown in Fig. 8, ROCKET achieves the highest average success rate of **71.7%**, outperforming SF (64.4%) and the 2D VLA baseline (51.7%), with the largest gain in the most challenging Env1. ROCKET also shows the smallest performance drop under OOD lighting. Per-run breakdowns are in Appendix B.2 (Table 10).

Failure Analysis. We categorize failures into four types: boundary perception, distance estimation, intra-object reasoning, and obstacle awareness (details in Appendix B.3). ROCKET reduces total failures from 87 to 51 (−41%), with the largest gains in distance estimation (−83%) and boundary perception (−52%). Intra-object reasoning (Case 3) shows no clear difference across methods, suggesting it remains an open challenge independent of 3D alignment.

Robustness. We further test under three perturbation conditions (light flashing, OOD objects, without container; 20 grasps each). As shown in Fig. 9, ROCKET achieves the best average robustness (68.3% vs. SF 63.3% vs. baseline 40.0%), with particularly strong generalization to unseen objects and simplified environments. Full results are in Table 12.

Table 5: Success rate (%) of ROCKET with different teacher models on LIBERO.

Teacher	OpenVLA-7B					PI0.5				
	Spa.	Obj.	Goal	Long	Avg.	Spa.	Obj.	Goal	Long	Avg.
Baseline	**97.2**	98.4	95.6	88.6	95.0	96.4	98.2	95.0	82.2	93.0
DA3	93.6	**99.8**	97.0	94.0	96.1	95.8	97.6	94.6	86.4	93.6
π^3	94.4	99.4	96.4	93.0	95.8	**97.6**	98.4	95.8	87.8	94.9
DINOv3	91.0	**99.8**	**98.0**	**94.4**	95.8	96.6	97.8	96.2	88.6	94.8
VGGT	96.2	98.0	97.2	93.6	**96.3**	96.4	**98.8**	**96.6**	**89.2**	**95.3**

7 Discussion

Robustness to Teacher Choice. We replace VGGT with Depth Anything 3 Lin et al. [2025a], π^3 Wang et al. [2025b], and DINOv3 Siméoni et al. [2025], keeping all other settings the same as ROCKET-VGGT. As shown in Table 5, different teachers consistently improve over the baseline on both OpenVLA-7B and PI0.5, while VGGT gives the largest gain, showing that ROCKET is not limited to a specific teacher model.

Optimal Number of Alignment Layers. A key design choice in multi-layer alignment is the number of aligned layer pairs. We fix the projector capacity and uniformly sample N layers from both the VLA LLM backbone and VGGT, aligning them as N pairs. As shown in Table 7, aligning 8–16 layer pairs yields the best trade-off; ROCKET uses $N = 10$ by default.

Table 6: Comparison of layer selection strategies on LIBERO.

Strategy	Spa.	Obj.	Goal	Long	Avg.
Baseline	97.2	98.4	95.6	88.6	95.0
Uniform-8	**98.2**	99.8	96.4	**95.0**	**97.4**
E2M-Last1	96.2	98.0	**97.2**	93.6	96.3
Shallow	97.8	99.8	95.4	85.6	94.7
Middle	89.4	99.8	94.2	91.2	93.7
Deep	92.0	99.8	95.6	93.6	95.3
Sim-1-Top	92.8	99.8	95.0	86.8	93.6
Sim-1-Last	93.8	**100.0**	95.6	92.8	95.6
Sim-2-Top	91.6	**100.0**	96.0	94.0	95.4
Sim-2-Last	91.8	**100.0**	93.6	94.8	95.1

Criteria for Layer Selection. We investigate layer selection strategies: (i) Uniform sampling; (ii) selecting from Shallow, Middle, and Deep stages; (iii) E2M-Last1: early-to-middle layers plus the last layer; and (iv) similarity-based heuristics (Sim-1/2-Top/Last) motivated by Gromov et al. [2024], He et al. [2024]. Table 6 shows that most reasonable strategies provide consistent gains, suggesting robustness to layer choice. We adopt E2M-Last1 by default. Details are in Appendix K.1, K.2.

Table 7: Impact of # aligned layers.

#	Spa.	Obj.	Goal	Long	Avg.
4	98.0	98.8	93.0	**95.2**	96.3
8	**98.2**	**99.8**	96.4	95.0	**97.4**
16	97.8	99.6	**97.6**	91.6	96.7
24	93.2	95.6	97.4	90.0	94.1

Table 8: Limited-data training.

Ratio	Spa.	Obj.	Goal	Long	Avg.
1%	75.0	88.8	65.0	65.6	73.6
5%	94.2	95.0	75.2	69.0	83.4
10%	94.0	91.8	78.6	72.6	84.3
100%	**96.2**	**98.0**	**97.2**	**93.6**	**96.3**

Data and Computational Efficiency. We evaluate efficiency from three perspectives. **(i) Limited-data training.** Table 8 shows that ROCKET achieves strong performance even with 1%–10% of LIBERO data (each trained for 30k steps), indicating high data efficiency. **(ii) Convergence speed.** In Fig. 3, ROCKET outperforms the baseline by ∼10% at around 10k steps and continues to improve (see Table 17 for complete results). **(iii) Same training budget.** Under the same training budget, ROCKET outperforms Spatial Forcing on both OpenVLA-7B and PI0.5 (Table 3).

Ablation Study. We ablate the key components of ROCKET in Table 9 (Full results in Table 17.). Introducing multi-layer alignment with *independent* projectors degrades performance to 80.0%, indicating gradient interference across layers. Replacing with the proposed *shared* projector improves to 98.2%. Adding Matryoshka-style sparse activation further increases to 98.5%.

Table 9: Ablation study of ROCKET on LIBERO.

Method	Spa.	Obj.	Goal	Long	Avg.
Baseline	94.0	97.2	98.0	92.4	95.4
+ Multi-layer	93.6	99.2	42.2	85.0	80.0
+ Shared	**99.0**	**99.8**	97.0	96.8	98.2
+ Matryoshka	98.2	**99.8**	**98.8**	**97.0**	**98.5**

8 Conclusion

We presented ROCKET, a residual-oriented multi-layer representation alignment framework that transfers 3D spatial priors from a strong vision foundation model into 2D-pretrained VLA models. By using a *layer-invariant shared projector*, ROCKET mitigates gradient interference that arises in naïve multi-layer distillation, and our Matryoshka-style sparse activation further balances supervision across depths. Extensive experiments show that ROCKET consistently improves spatial grounding and generalization across datasets and VLA backbones, achieving near state-of-the-art performance on LIBERO with only a small fraction of the compute. We hope ROCKET offers a simple and scalable route to more reliable 3D-aware robotic manipulation.

Acknowledgments and Disclosure of Funding

This work was supported by the Cisco Research Award and the Qualcomm Innovation Fellowship. The authors declare no competing interests.

References

Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, et al. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

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.

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. In *ArXiv Preprint*, March 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael 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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL `https://arxiv.org/abs/2410.24164`.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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 (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40. PMLR, 2025.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Bo Chang, Lili Meng, Eldad Haber, Frederick Tung, and David Begert. Multi-level residual networks from dynamical systems view. *arXiv preprint arXiv:1710.10348*, 2017.

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5008–5017, 2021.

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*, 2025a.

Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025b.

Yudong Chen, Sen Wang, Jiajun Liu, Xuwei Xu, Frank de Hoog, and Zi Huang. Improved feature distillation via projector ensemble. *Advances in Neural Information Processing Systems*, 35: 12084–12095, 2022.

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.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*, 2019.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090.

Guoqiang Gong, Jiaxing Wang, Jin Xu, Deping Xiang, Zicheng Zhang, Leqi Shen, Yifeng Zhang, Junhua Shu, Zhaolong Xing, Zhen Chen, Pengzhang Liu, and Ke Zhang. Beyond logits: Aligning feature dynamics for effective knowledge distillation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23067–23077, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1125. URL `https://aclanthology.org/2025.acl-long.1125/`.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024.

Minghao Guo, Meng Cao, Jiachen Tao, Rongtao Xu, Yan Yan, Xiaodan Liang, Ivan Laptev, and Xiaojun Chang. Glad: Geometric latent distillation for vision-language-action models. *arXiv preprint arXiv:2512.09619*, 2025.

Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*, 2024.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, and Yuntao Chen. Dita: Scaling diffusion transformer for generalist vision-language-action policy. *arXiv preprint arXiv:2503.19757*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, et al. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025a.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D value maps for robotic manipulation with language models. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 540–562. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/huang23b.html.

Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. Mllms need 3d-aware representation supervision for scene understanding. *arXiv preprint arXiv:2506.01946*, 2025b.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

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 Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 06–09 Nov 2024. URL https://proceedings.mlr.press/v270/kim25c.html.

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

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249, 2022.

Byung-Kwan Lee, Ryo Hachiuma, Yu-Chiang Frank Wang, Yong Man Ro, and Yueh-Hua Wu. Vlsi: Verbalized layers-to-interactions from large to small vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29545–29557, 2025.

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. *arXiv preprint arXiv:2510.12276*, 2025a.

Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, et al. Cronusvla: Towards efficient and robust manipulation via multi-frame vision-language-action modeling. *arXiv preprint arXiv:2506.19816*, 2025b.

Wei Li, Jizhihui Liu, Li Yixing, Junwen Tong, Rui Shao, and Liqiang Nie. ConsisVLA-4D: Advancing spatiotemporal consistency in efficient 3D-perception and 4D-reasoning for robotic manipulation. *arXiv preprint arXiv:2605.05126*, 2026.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025a.

Juyi Lin, Amir Taherin, Arash Akbari, Arman Akbari, Lei Lu, Guangyu Chen, Taskin Padir, Xiaomeng Yang, Weiwei Chen, Yiqian Li, et al. Vote: vision-language-action optimization with trajectory ensemble voting. *arXiv preprint arXiv:2507.05116*, 2025b.

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, pages 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. *arXiv preprint arXiv:2410.07864*, 2024.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Roy Miles and Krystian Mikolajczyk. Understanding the role of the projector in knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4233–4241, 2024.

Roy Miles, Ismail Elezi, and Jiankang Deng. Vkd: Improving knowledge distillation using orthogonal projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15720–15730, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021.

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. *arXiv preprint arXiv:2508.19236*, 2025.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of The 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 894–906. PMLR, 08–11 Nov 2022. URL `https://proceedings.mlr.press/v164/shridhar22a.html`.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 785–799. PMLR, 14–18 Dec 2023. URL `https://proceedings.mlr.press/v205/shridhar23a.html`.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

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.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Haochen Wang, Anlin Zheng, Yucheng Zhao, Tiancai Wang, Zheng Ge, Xiangyu Zhang, and Zhaoxiang Zhang. Reconstructive visual instruction tuning. *arXiv preprint arXiv:2410.09575*, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025a.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025b.

Junjin Xiao, Dongyang Li, Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Feng Xiong, Mu Xu, Xing Wei, Zhiheng Ma, Qing Zhang, and Wei-Shi Zheng. Learning action manifold with multi-view latent priors for robotic manipulation. *arXiv preprint arXiv:2605.11832*, 2026.

Bin Yu, Shijie Lian, Xiaopeng Lin, Zhaolong Shen, Yuliang Wei, Haishan Liu, Changti Wu, Hang Yuan, Bailing Wang, Cong Huang, and Kai Chen. 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.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 06–09 Nov 2023. URL `https://proceedings.mlr.press/v229/zitkovich23a.html`.

A Experimental Setup and Training Details

A.1 Datasets and Training Details

LIBERO and LIBERO-Plus. We conduct experiments on LIBERO Liu et al. [2023] and LIBERO-PLUS Fei et al. [2025]. Following the official protocol of LIBERO-PLUS as well as the common practice used in PI0/PI0.5, we mix the four sub-datasets during training and evaluate a *single* model on the four corresponding test suites. Unless otherwise specified, we fine-tune the OPENVLA-7B pretrained weights on LIBERO and LIBERO-PLUS. We adopt the training setting of SF and OPENVLA-OFT. Specifically, we perform LoRA fine-tuning for $50{,}000$ steps with batch size 32. In addition, to compare multiple methods under the *full fine-tuning* regime as in PI0.5, we run full fine-tuning on LIBERO for $30{,}000$ steps with batch size 64.

RoboTwin 2.0. We follow the official training protocol and perform LoRA fine-tuning for $30{,}000$ steps with a batch size of 16. For evaluation, we follow the ROBOTWIN 2.0 Chen et al. [2025a] evaluation procedure with a random seed of 0, and run 100 trials for both the easy and hard task sets. We use the task subset specified in SF for both training and evaluation. For all benchmarks, we report task execution success rate (SR, %) by default. For the RoboTwin 2.0 Hard experiments in Table 4, all methods are trained and evaluated on the Hard setting, using 50 demonstrations per task from the official Hard data and 100 evaluation trials per task. `ROCKET` follows the hyperparameters in Fig. 6, while Spatial Forcing follows the settings in its original paper. We report two settings: one model trained for each task, and one policy trained for all five tasks.

Projector and Alignment Hyperparameters. To ensure a fair comparison, we keep the projector architecture *identical across all experiments*. In particular, we use a *single* projector whose structure and dimensionalities exactly match SF. For the alignment-loss weight λ and the number of layers used by the single-layer alignment baseline in SF, we directly use the best configurations reported by their hyperparameter search.

A.2 Training Hyperparameters

All experiments are conducted on NVIDIA H200 GPUs.

Teacher Freezing. Throughout the entire training pipeline, we keep the teacher model frozen and only fine-tune the student VLA model.

Learning Rate Schedule and Batch Size. We train OPENVLA-7B for a total of $50{,}000$ steps, following its default learning-rate decay strategy with a minor adjustment: we keep the learning rate at 5×10^{-4} for the first $10{,}000$ steps, and then decay it to 5×10^{-5} for the remainder of training. We use a batch size of 32.

For the search-oriented experiments in Sec. 7 and Table 8, to reduce computational cost we decrease the batch size to 16. Meanwhile, to avoid under-training, we keep the learning rate at 5×10^{-4} for the first $30{,}000$ steps and then decay it to 5×10^{-5} afterwards.

PI-Model Experiments. For experiments related to PI0.5, we set the batch size to 64 and train the model in PyTorch; all other settings follow the default configuration. For PI0 experiments on ROBOTWIN 2.0, we use a batch size of 16 and train with the JAX implementation.

We also observe that in such single-task settings, the action loss typically converges quickly. Therefore, we reduce the coefficient of the alignment loss from its default value of 0.5 to 0.125, which yields strong empirical performance.

B Real-World Experiment Details

This appendix provides the full details of the real-world experiments summarized in Sec. 6.

Table 10: Real-world success rate (%). Each setting is evaluated over 3 runs of 20 consecutive grasps (60 grasps per cell). ROCKET achieves the best performance across all settings.

Environment	Method	Run 1	Run 2	Run 3	Total	SR (%)
Env1 (Night + Artif.)	2D VLA	12	7	9	28	46.7
	SF	11	11	10	32	53.3
	ROCKET	12	14	12	38	**63.3**
Env2 (Night + Artif.)	2D VLA	9	13	11	33	55.0
	SF	16	13	15	44	73.3
	ROCKET	16	18	13	47	**78.3**
Env2 OOD (Day + Natural)	2D VLA	11	12	9	32	53.3
	SF	13	15	12	40	66.7
	ROCKET	14	15	15	44	**73.3**
Average		2D VLA: 51.7			SF: 64.4	**71.7**

B.1 Experimental Setup

Hardware and Data Collection. We use a Ufactory xArm 850 with a UFACTORY xArm Gripper G2, as shown in Fig. 7. Data is collected via teleoperation using Meta Quest 3. The operator repeatedly moves packages from a loaded container to a target location until the container is nearly empty. Data is sampled at 10 Hz, totaling **110,657 frames** ($\sim$600 grasps; 80% from Env 1, 20% from Env 2). Training follows the LIBERO setting (Appendix A.1) with a batch size of 16.

Task and Environments. The target scenario is warehouse package picking and placing. We consider two environments: **Env1**, picking from a wire-mesh container to a right-side conveyor belt; and **Env2**, picking from a box to adjacent empty slots. The default lighting condition is *nighttime with artificial lighting*; we additionally evaluate under *daytime with natural light* as an out-of-distribution (OOD) condition for Env2.

Task Difficulty. This task is challenging due to three factors: (i) *long-horizon sequential operation*: the robot must complete 20 consecutive grasps per round without scene reset, and the parcel layout changes irreversibly after each grasp; (ii) *deformable objects with high visual ambiguity*: all parcels are soft-body with identical appearance (solid color, same size), and when densely stacked their boundaries become extremely ambiguous; (iii) *strict success criteria*: parcels contain medicine bottles with only $\sim$30% graspable area, each grasp must pick exactly one parcel, and the cluttered scene requires collision-free operation. The 2D VLA baseline achieves only 51.7% average success rate, confirming the inherent difficulty.

Evaluation Protocol and Baselines. For each setting, we perform 20 consecutive grasps from a fully loaded container, repeated 3 times (60 grasps per setting). After each placement, the robot resets to its initial pose. We compare three methods: (i) the **2D VLA** baseline (without spatial grounding); (ii) **Spatial Forcing (SF)** Li et al. [2025a], the current SOTA implicit single-layer alignment method; and (iii) ROCKET. All methods use *identical training data and compute budget*.

B.2 Main Results

Table 10 reports the per-run success rates across all environments and lighting conditions. ROCKET achieves the highest **average success rate of 71.7%**, outperforming SF (64.4%) and the 2D VLA baseline (51.7%).

The improvement is most pronounced in the most challenging setting. In **Env1**, where the semi-transparent wire-mesh container poses significant collision risk, ROCKET reaches 63.3%, surpassing SF by **+10.0 pp** and the baseline by **+16.6 pp**. In **Env2** under in-distribution lighting, ROCKET achieves 78.3%, the highest across all settings. Under the **OOD lighting condition** (Env2, daytime + natural light), ROCKET maintains 73.3% with only a -5.0 pp drop from the in-distribution setting (78.3%$\rightarrow$73.3%), compared to SF's -6.6 pp drop (73.3%$\rightarrow$66.7%), demonstrating stronger robustness to lighting variation.



***(a)* Successful grasp:** The deployed VLA model is Baseline.



***(b)* Failure Case 1:** Grasped multiple packages



***(c)* Failure Case 2:** Grasped too early



***(d)* Failure Case 3:** Incorrect grasp position



***(e)* Failure Case 4:** Potential collision / safety-triggered termination / failed placement

Figure 10: Visualization of four diagnostic failure types. **Case 1**: grasped multiple parcels (boundary perception). **Case 2**: premature grasp (distance estimation). **Case 3**: suboptimal grasp position (intra-object reasoning). **Case 4**: collision or safety termination (obstacle awareness).

B.3 Fine-Grained Failure Analysis

To understand *which* spatial capabilities benefit from 3D alignment, we categorize all failures into four diagnostic types, each targeting a distinct perceptual dimension (see Fig. 10 for visual examples):

- **Case 1 (Boundary Perception):** Grasped multiple parcels simultaneously, reflecting failure in perceiving object boundaries among visually identical, densely stacked items.
- **Case 2 (Distance Estimation):** Gripper closed prematurely before reaching the target parcel, reflecting incorrect depth estimation. This is especially challenging inside containers where only the wrist camera provides useful viewpoint, easily confused by ubiquitous solid-color parcels.
- **Case 3 (Intra-Object Reasoning):** Grasped at a suboptimal position within the parcel (e.g., the bottle region instead of the corner), reflecting the ability to identify which positions are easier for grasping.
- **Case 4 (Obstacle Awareness):** Potential collision detected, safety-triggered termination, or failed placement due to environmental interference, reflecting spatial understanding of surrounding obstacles and trajectory planning.

Table 11 reports the absolute failure counts across all settings. Due to varying total failure counts across settings, we report absolute numbers rather than percentages. The aggregated results (bottom three rows) reveal clear patterns:

Cases 1, 2, and 4 show systematic improvement with 3D alignment. For **Case 1** (boundary perception), ROCKET reduces failures from 23 to 11 (-52% vs. baseline), indicating that multi-layer 3D alignment helps distinguish individual parcels among densely stacked, visually identical items. For **Case 2** (distance estimation), the improvement is the most dramatic: ROCKET reduces failures from 24 to only 4 ($-\mathbf{83\%}$ vs. baseline), confirming that multi-layer spatial features provide substantially better depth cues than either single-layer alignment or no alignment. For **Case 4** (obstacle awareness),

Table 11: Failure analysis across all real-world settings. Numbers denote absolute failure counts. Cases 1, 2, and 4 show clear improvement with 3D alignment; Case 3 remains comparable across methods, indicating a gap in fine-grained intra-object reasoning.

Environment	Method	Total	Case 1 Boundary	Case 2 Distance	Case 3 Position	Case 4 Obstacle
Env1 (Night + Artif.)	2D VLA	32	4	6	8	14
	SF	28	4	3	9	12
	ROCKET	**22**	**2**	**1**	9	**10**
Env2 (Night + Artif.)	2D VLA	27	9	11	4	3
	SF	16	3	2	6	5
	ROCKET	**13**	4	**1**	**5**	**3**
Env2 OOD (Day + Natural)	2D VLA	28	10	7	6	5
	SF	20	5	3	7	5
	ROCKET	**16**	5	**2**	**6**	**3**
Sum (All)	2D VLA	87	23	24	**18**	22
	SF	64	12	8	22	22
	ROCKET	**51**	**11**	**4**	20	**16**

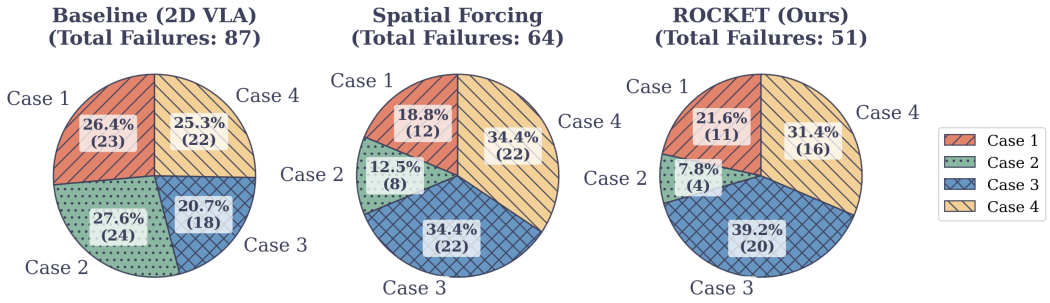


Figure 11: Aggregate failure distribution across all settings. ROCKET reduces total failures from 87 (baseline) to 51 (-41%), with the largest reductions in distance estimation (Case 2) and boundary perception (Case 1).

ROCKET achieves 16 failures versus 22 for both SF and the baseline (-27%), demonstrating better spatial understanding for collision-free trajectory execution.

Case 3 shows no clear difference across methods (18 for baseline, 22 for SF, 20 for ROCKET). This suggests that current VLA models generally lack fine-grained intra-object reasoning; for example, they do not infer that grasping parcel corners increases success probability when only $\sim$30% of the surface is a valid grasp region. Notably, this limitation is *independent of 3D spatial alignment*: neither implicit nor explicit 3D features address the need for object-level affordance reasoning. We consider this an important direction for future work.

Fig. 11 visualizes the aggregate failure distribution, and Fig. 12 provides a per-environment breakdown.

B.4 Robustness to Environmental Perturbations

To evaluate generalization beyond the training distribution, we test under three additional perturbation conditions with 20 grasps each:

- **Light Flashing**: Flickering lights during operation, simulating unstable industrial lighting.
- **OOD Object**: Unseen package types not present in the training data, testing object-level generalization.
- **W/O Container**: Parcels placed directly without the wire-mesh container, removing the obstacle constraint but also eliminating familiar visual context.

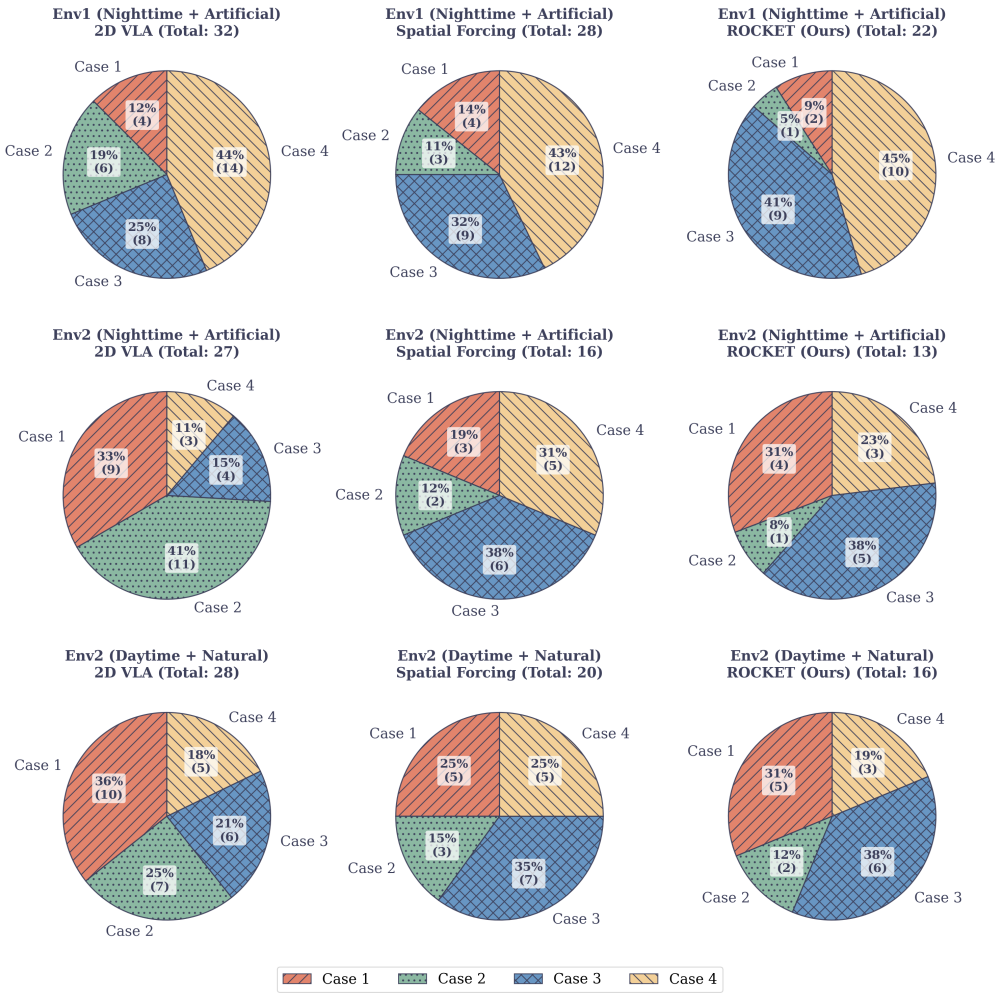


Figure 12: Per-environment failure distribution for all three methods. Each row corresponds to one environment setting (Env1 nighttime, Env2 nighttime, Env2 OOD daytime); columns correspond to methods. Pie size reflects the total number of failures. **Env1** is dominated by Case 4 (collision) due to the wire-mesh container; **Env2** failures shift to Case 1 (boundary) and Case 2 (distance); under **OOD lighting**, Case 1 increases further, but ROCKET remains the most robust.

As shown in Table 12, ROCKET achieves the best average robustness (**68.3%**), outperforming SF (63.3%) and the baseline (40.0%). Notably, ROCKET shows the strongest **generalization to unseen objects** (70%, +10 pp vs. SF, +25 pp vs. baseline), suggesting that multi-layer 3D alignment captures geometric features that transfer well across object types rather than overfitting to training-set appearance. ROCKET also achieves the **highest performance when the container is removed** (75%, +10 pp vs. SF), indicating that its spatial understanding does not depend on a specific obstacle configuration. Under **light flashing**, SF (65%) slightly outperforms ROCKET (60%), though both substantially outperform the baseline (35%).

C Extended Discussion

This appendix contains the full original discussion with complete per-suite breakdowns and detailed analysis.

Table 12: Robustness to environmental perturbations (SR%, 20 grasps per condition). ROCKET achieves the best average robustness, with particularly strong generalization to unseen objects and simplified environments.

Perturbation	2D VLA	SF	ROCKET	Δ vs. SF
Light Flashing	35.0	**65.0**	60.0	−5.0
OOD Object	45.0	60.0	**70.0**	+10.0
W/O Container	40.0	65.0	**75.0**	+10.0
Average	40.0	63.3	**68.3**	+5.0

C.1 Optimal Number of Alignment Layers

A key design choice in multi-layer alignment is the number of aligned layer pairs. To ensure a fair comparison, we fix the projector capacity and study how many layers should be aligned under a fixed-size projector. Following the strategy in Sun et al. [2019], we uniformly sample N layers from the VLA LLM backbone and N layers from VGGT, and align them as N layer pairs. As shown in Table 7, aligning roughly 8–16 layer pairs yields the best trade-off, and performance saturates or degrades beyond this range. Therefore, ROCKET uses $N = 10$ aligned layer pairs by default in all experiments.

C.2 Criteria for Layer Selection

We investigate several layer selection strategies. We apply the same strategy to select the same layers for both VGGT and VLA. Specifically, we consider: (i) the Uniform sampling strategy in Sec. 7; (ii) selecting 10 layers separately from the Shallow, Middle, and Deep stages; (iii) E2M-Last1: uniformly sampling from early-to-middle layers and additionally including the last layer; and (iv) two importance-based heuristics: Motivated by the hypothesis that layers with higher input–output cosine similarity are less informative Gromov et al. [2024], He et al. [2024], we design two similarity-based criteria. Specifically, the first criterion computes the cosine similarity between the input and output representations aggregated over windows of 10 consecutive Transformer blocks; Sim-1-Top selects the 10 layers with the highest similarity, while Sim-1-Last selects those with the lowest similarity. The second one computes similarity per block; Sim-2-Top selects the 10 highest-similarity layers, whereas Sim-2-Last selects the 10 lowest-similarity layers.

Results in Table 6 show that even without expensive post-hoc search, most reasonable selection strategies provide consistent gains, suggesting that ROCKET is robust to layer choice. Unless otherwise stated, we adopt the E2M-Last1 strategy as the default setting.

C.3 Data and Computational Efficiency

We evaluate the efficiency of ROCKET from three perspectives: **(i) Limited-data training.** Using 1%, 5%, and 10% random subsets of LIBERO (each trained for 30k steps), Table 8 shows that ROCKET still achieves strong performance, indicating high data efficiency, which is particularly important in robotics where data collection is costly. **(ii) Convergence speed.** In Fig. 3, ROCKET outperforms the baseline by ∼10% at around 10k steps and continues to improve thereafter, suggesting that the proposed multi-level alignment mechanism more effectively exploits spatial cues across different layers. Please refer to Table 17 for the complete results. **(iii) Same training budget.** The results of prior methods in Table 2 are taken from their original papers, which use different training budgets. We therefore compare ROCKET with Spatial Forcing under the same training budget in Table 3, and report their full training trajectories in Table 17.

C.4 Ablation Study

We ablate the key components of ROCKET. Introducing multi-layer alignment with *independent* projectors degrades performance to 80.0%, indicating gradient interference across layers. Replacing independent projectors with the proposed *shared* projector improves performance to 98.2%. Finally, adding the Matryoshka-style sparse activation further increases performance to 98.5%.

Table 13: Ablation study of ROCKET on LIBERO (full per-suite results).

Method	Spatial	Object	Goal	Long	Avg.
Baseline	94.0	97.2	98.0	92.4	95.4
+ Multi-layer	93.6	99.2	42.2	85.0	80.0
+ Shared	**99.0**	**99.8**	97.0	96.8	98.2
+ Matryoshka	98.2	**99.8**	**98.8**	**97.0**	**98.5**

C.5 Additional Ablation Results

Multiple runs of multi-layer alignment with independent projectors. We repeat multi-layer alignment with independent projectors ("+ Multi-layer" in Table 9) with two additional random seeds. As shown in Table 14, although Runs 2 and 3 are better than Run 1, all three runs remain below the baseline, showing that the degradation is not caused only by the unusually low Goal score in one run, although its severity varies across seeds.

Table 14: Multi-layer alignment with independent projectors on LIBERO over three runs.

Method	Spatial	Object	Goal	Long	Avg.
Baseline	94.0	97.2	**98.0**	**92.4**	**95.4**
Run 1	93.6	99.2	42.2	85.0	80.0
Run 2	**96.0**	97.6	84.6	97.2	93.9
Run 3	94.8	**100.0**	79.6	88.2	90.7
Mean±Std	94.8±1.2	98.9±1.2	68.8±23.2	90.1±6.3	88.2±7.3

Matryoshka-style activation under different layer-selection strategies. We further evaluate Matryoshka-style activation under the three best layer-selection strategies in Table 6, with the same training settings. "Shared-only" removes Matryoshka activation while keeping the shared projector. As shown in Table 15, removing Matryoshka reduces the average performance under all three strategies, by 0.7, 1.8, and 1.6 points, respectively, showing that its benefit is not specific to one choice of aligned layers.

Table 15: Matryoshka-style activation under different layer-selection strategies on LIBERO.

Method	Spatial	Object	Goal	Long	Avg.
Uniform-8 (Shared-only)	98.8	99.6	96.2	92.2	96.7
Uniform-8 (ROCKET)	98.2	99.8	96.4	95.0	**97.4**
E2M-Last1 (Shared-only)	99.2	97.6	92.0	89.2	94.5
E2M-Last1 (ROCKET)	96.2	98.0	97.2	93.6	**96.3**
Sim-1-Last (Shared-only)	96.2	99.6	93.0	87.2	94.0
Sim-1-Last (ROCKET)	93.8	100.0	95.6	92.8	**95.6**

Loss reweighting vs. parameter allocation. We compare two ways of assigning greater importance to deeper layers: changing the alignment-loss weights and changing the number of active parameters in the shared projector. As shown in Table 16, increasing the importance from shallow to deep layers performs better than the reverse direction. In addition, controlling the importance through the number of active parameters (98.5%) performs better than changing the loss weights (95.1%), showing that the benefit does not come only from assigning larger loss weights to deeper layers.

D Performance Trajectories on LIBERO Across Training

Table 17 reports success rates on the four LIBERO suites (Spatial, Object, Goal, and Long) at different training stages, together with the overall average. Unlike single-point evaluation at the end of training, this table highlights performance trajectories over the full training course, enabling a more complete comparison of training dynamics across methods.

Table 16: Assigning importance across depths through loss weights or through active parameters of the shared projector on LIBERO.

Method (shallow → deep)	Spatial	Object	Goal	Long	Avg.
Loss weight, decreasing	94.2	96.6	88.4	84.2	90.9
Loss weight, increasing	**98.8**	96.8	94.2	90.6	95.1
Active parameters, decreasing	92.2	**100.0**	93.6	91.4	94.3
Active parameters, increasing (ROCKET)	98.2	99.8	**98.8**	**97.0**	**98.5**

Table 17: Success rates (%) on the LIBERO benchmarks. We compare the performance trajectories of multiple methods over the full training course; **bold** indicates the best final result, and underlining indicates the best result achieved during training.

Method	Steps	Spatial	Object	Goal	Long	Avg.
Baseline	10k	74.6	89.4	78.6	37.2	70.0
	20k	93.0	99.6	97.6	94.4	96.2
	30k	94.6	99.2	98.2	93.6	96.4
	50k	94.0	97.2	98.0	92.4	95.4
Spatial Forcing	10k	83.4	99.8	52.6	31.6	66.9
	20k	94.2	99.8	94.8	90.4	94.8
	30k	97.8	**100.0**	92.6	95.2	96.4
	50k	97.8	98.8	96.2	90.6	95.9
Multi-layer Alignment	10k	86.2	74.6	22.4	52.6	59.0
	20k	93.8	99.8	35.0	83.4	78.0
	30k	94.6	**100.0**	37.4	81.8	78.5
	50k	93.6	99.2	42.2	85.0	80.0
ROCKET (Shared-only)	10k	94.2	95.2	70.4	70.2	82.5
	20k	94.0	99.6	97.4	94.6	96.4
	30k	97.0	99.6	98.6	92.4	96.9
	50k	**99.0**	99.8	97.0	96.8	98.2
ROCKET	10k	90.8	91.4	74.0	55.6	78.0
	20k	93.4	**100.0**	98.0	94.4	96.5
	30k	96.0	99.2	98.0	93.8	96.8
	50k	98.2	99.8	**98.8**	**97.0**	**98.5**

E Dataset Details

E.1 LIBERO

LIBERO Liu et al. [2023] is a benchmark suite for language-conditioned robotic manipulation, designed to evaluate multi-task imitation learning and generalization. The original LIBERO release consists of **130** manipulation tasks grouped into **4** task suites: LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal with 10 tasks each, and LIBERO-100, which is further split into LIBERO-90 and LIBERO-10 (also referred to as LIBERO-Long). Tasks are defined with natural language instructions, enabling direct study of vision-language-action (VLA) policy learning.

The four task suites capture complementary generalization axes. LIBERO-SPATIAL focuses on spatial configuration variation, LIBERO-OBJECT emphasizes object-level diversity, LIBERO-GOAL introduces goal variation within shared task structures, and LIBERO-LONG contains long-horizon, multi-stage manipulation tasks requiring sequential reasoning Liu et al. [2023]. This structured design enables controlled evaluation of cross-task and cross-goal generalization.

LIBERO provides **50** human-teleoperated demonstrations for every task, collected in simulation using RoboSuite. Following common practice, we use the Spatial, Object, Goal, and Long suites, each with 10 tasks, and evaluate each suite over 500 episodes (50 per task). Each episode consists of temporally dense state-action sequences spanning hundreds of timesteps.

Each episode includes synchronized multimodal observations, including RGB images from fixed camera viewpoints and robot proprioceptive states such as joint positions and gripper status. Actions

are specified in continuous control space, typically as end-effector pose commands. Data are stored in a standardized trajectory format, facilitating efficient loading and integration with common imitation learning pipelines.

E.2 LIBERO-Plus

LIBERO-Plus Fei et al. [2025] is a robustness benchmark built on LIBERO for in-depth analysis of vision-language-action (VLA) models. It introduces controlled perturbations along **seven** dimensions: objects layout, camera viewpoints, robot initial states, language instructions, light conditions, background textures, and sensor noise, resulting in **10,030** tasks. LIBERO-Plus also provides new assets, including hundreds of new objects and textures, together with the corresponding training data, including a training set for each of the four task suites. Its usage is identical to LIBERO, so existing LIBERO training and evaluation pipelines can be reused directly.

Each episode includes synchronized multimodal observations, including RGB visual inputs, robot proprioceptive states, and continuous control actions specified in end-effector space. Data are stored in the same standardized trajectory format as LIBERO, enabling seamless reuse of training pipelines and direct comparison across LIBERO and LIBERO-Plus benchmarks.

E.3 RoboTwin 2.0

RoboTwin 2.0 Chen et al. [2025a] is a large-scale simulation benchmark and dataset for bimanual robotic manipulation, providing both an automated data generator and unified evaluation protocols. The official release instantiates **50 dual-arm tasks** across **5 robot embodiments**, targeting robust long-horizon manipulation with diverse object interactions. RoboTwin 2.0 is built on an annotated object library (RoboTwin-OD) containing **731 object instances** spanning **147 categories**, enabling systematic task variation in object geometry, placement, and scene composition.

In addition to the generator, RoboTwin 2.0 provides **over 100,000** pre-collected expert trajectories/episodes for downstream training and benchmarking. Each episode consists of temporally aligned state-action sequences with multimodal observations. Visual observations include rendered RGB images from calibrated cameras (often multi-view) to capture both global context and fine-grained contact-relevant details. Proprioceptive observations include robot joint states (e.g., positions/velocities) and gripper states. Actions are specified in continuous control space (e.g., end-effector or joint-space commands), which makes the dataset compatible with standard imitation learning and policy learning pipelines.

Overall, RoboTwin 2.0 is designed to support scalable training and controlled evaluation of general-purpose policies, particularly for studying cross-task and cross-embodiment generalization under instruction- or goal-conditioned settings.

F Model Details

F.1 OpenVLA

OpenVLA Kim et al. [2024] is a 7B-parameter VLA based on the **Prismatic-7B VLM** Karamcheti et al. [2024] framework. The architecture's vision tower employs a fused visual encoder (totaling 600M parameters) constructed by combining two pretrained vision models: **DINOv2** Oquab et al. [2023] and **SigLIP** Zhai et al. [2023]. Specifically, input image patches are passed separately through both encoders, and the resulting feature vectors are concatenated channel-wise. These fused visual features are then mapped into the language embedding space via a small 2-layer Multi-Layer Perceptron **(MLP) projector**. The central reasoning engine is a 7B-parameter **Llama 2** Touvron et al. [2023] language model backbone, which processes the interleaved visual tokens and language instructions to predict 7-dimensional robot actions. It features **32 Transformer layers** with **a hidden dimension of 4096** and **32 attention heads**. OpenVLA represents robot actions by discretizing them into **256 bins per dimension**, treating them as specialized tokens within the LLM's vocabulary.

F.2 OpenVLA-OFT

OpenVLA-OFT Kim et al. [2025] is a fine-tuning variant of OpenVLA that inherits the same base components. Instead of discretizing actions, OpenVLA-OFT employs a **continuous action prediction head**—a 2-block MLP ResNet that directly predicts 7-dimensional continuous actions from the LLM's hidden states via L1 regression. During fine-tuning, OpenVLA-OFT freezes the vision tower and projector and applies **LoRA** Hu et al. [2021] adapters to the LLM backbone, targeting all linear layers in self-attention and MLP blocks. OpenVLA-OFT+ extends it with **FiLM** Perez et al. [2018] for enhanced language grounding. Specifically, FiLM applies learnable affine transformations to the visual features based on the language instruction, enabling the model to modulate visual representations according to task-specific language cues.

F.3 PI0

PI0Black et al. [2024] is a vision-language-action (VLA) model that generates robot action sequences using flow matching. The model employs a dual-expert architecture: a vision-language expert and an action expert. The vision tower uses **SigLIP So400m/14** Zhai et al. [2023]. The language model is **PaliGemma** Beyer et al. [2024] based on Gemma 2B Mesnard et al. [2024], with **hidden dimension 2048, 18 transformer layers**. The action expert is a **Gemma 300M** model with hidden dimension 1024, 18 layers. During training, PI0 learns a velocity field $v_t(x)$ via **flow matching**.

F.4 PI0.5

PI0.5 Black et al. [2025] shares the same architectural foundation as PI0 but introduces two key modifications for improved open-world generalization through knowledge insulation. First, instead of providing the robot state as a continuous vector in the suffix, PI0.5 discretizes the state into **256 bins** over $[-1, 1]$ and incorporates it as discrete tokens in the language prefix. Second, PI0.5 replaces MLP-based timestep fusion with adaptive RMS normalization (**AdaRMSNorm**) for timestep conditioning. The timestep embedding is processed through a two-layer MLP with Swish activations to produce a 1024-dimensional condition vector. This condition is injected into the action expert through AdaRMSNorm layers in each of the 18 transformer layers, where AdaRMSNorm dynamically generates scale and shift parameters, which provides more flexible timestep conditioning compared to the MLP fusion approach in PI0.

G Projector Similarity

We flatten the MLP parameters of each independently initialized projector (i.e., $\phi = \{W_1, W_2\}$) and compute the pairwise cosine similarities. As shown in Fig. 13, the similarities between different projectors are nearly zero, indicating that naive multi-layer alignment with multiple independent projectors tends to learn distinct mapping trajectories when modeling the transformation from one residual stream to another. This discrepancy can lead to gradient conflicts, which in turn degrades the final performance.

H Why We Need a Nested Sparse-Activation Mechanism to Balance Multiple Alignment Losses

We compute the similarity between the student model OpenVLA-7B and the teacher model VGGT using centered kernel alignment (CKA) Cortes et al. [2012]. As shown in Fig.14, we observe that the shallow layers of VGGT are more similar to OpenVLA-7B, while the similarity gradually decreases as the depth increases. This finding not only suggests that different layers of VGGT contain semantic information with varying levels of richness, but also indicates that it is easier to align VGGT and OpenVLA-7B at shallow layers. In practice, a shared projector typically learns a mapping that works well for aligning shallow layers; however, it often fails to achieve equally good alignment for deeper layers.

To address this multi-layer alignment mismatch, we propose a nested sparse-activation mechanism. The key idea is to let shallow layers activate (and thus update/use) fewer parameters in the projector,

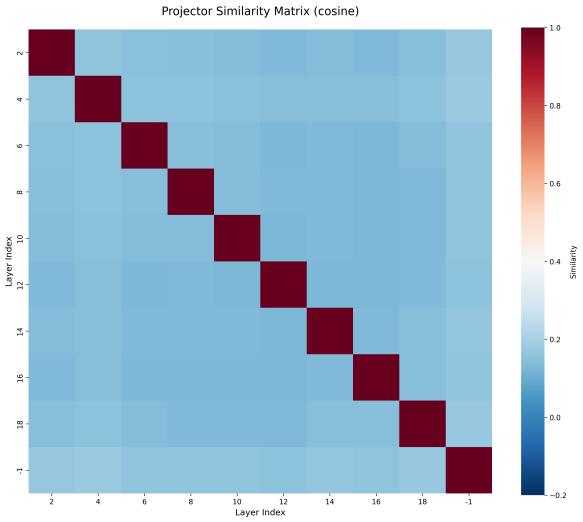


Figure 13: Pairwise cosine similarity of flattened MLP parameters for independently initialized projectors. The similarities are nearly zero, suggesting that naive multi-layer alignment with multiple independent projectors learns distinct mapping trajectories, which can induce gradient conflicts and harm final performance.

and gradually increase the number of activated parameters as the layer depth increases, until reaching the full projector capacity at the deepest layers.

CKA principle. CKA measures the similarity between two representation spaces by comparing their induced pairwise similarity structure, and can be viewed as a normalized version of HSIC (Hilbert-Schmidt Independence Criterion). Concretely, suppose we collect representations from one layer of the student model as a matrix $X \in \mathbb{R}^{n \times d_x}$ and from one layer of the teacher model as $Y \in \mathbb{R}^{n \times d_y}$, where n is the number of matched observations (e.g., the same n tokens or image regions), while d_x and d_y can be different.

A common choice is *linear CKA*, computed via Gram matrices:

$$K = XX^\top, \quad L = YY^\top,$$

and centering them with the centering matrix $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$:

$$\tilde{K} = HKH, \quad \tilde{L} = HLH.$$

Then CKA is

$$\mathrm{CKA}(X, Y) = \frac{\langle \tilde{K}, \tilde{L} \rangle_F}{\|\tilde{K}\|_F \, \|\tilde{L}\|_F},$$

where $\langle A, B \rangle_F = \mathrm{tr}(A^\top B)$ is the Frobenius inner product. Equivalently, for centered X and Y, linear CKA can be written as:

$$\mathrm{CKA}(X, Y) = \frac{\|X^\top Y\|_F^2}{\|X^\top X\|_F \, \|Y^\top Y\|_F}.$$

Importantly, the above formulation does *not* require $d_x = d_y$; it only requires that X and Y share the same number of rows n (i.e., aligned observations). Intuitively, CKA compares how similarly the two models organize the same set of examples in representation space, and it is invariant to isotropic scaling (and, in the linear case, robust to orthogonal transformations), making it suitable for cross-model, cross-dimension similarity analysis.

Details of computing CKA similarity between OpenVLA-7B and VGGT. Since OpenVLA-7B and VGGT have different hidden dimensions, we adopt CKA because it can compare representations of different feature dimensions as long as they are evaluated on the same set of samples (tokens/images), i.e., the same number of observations.

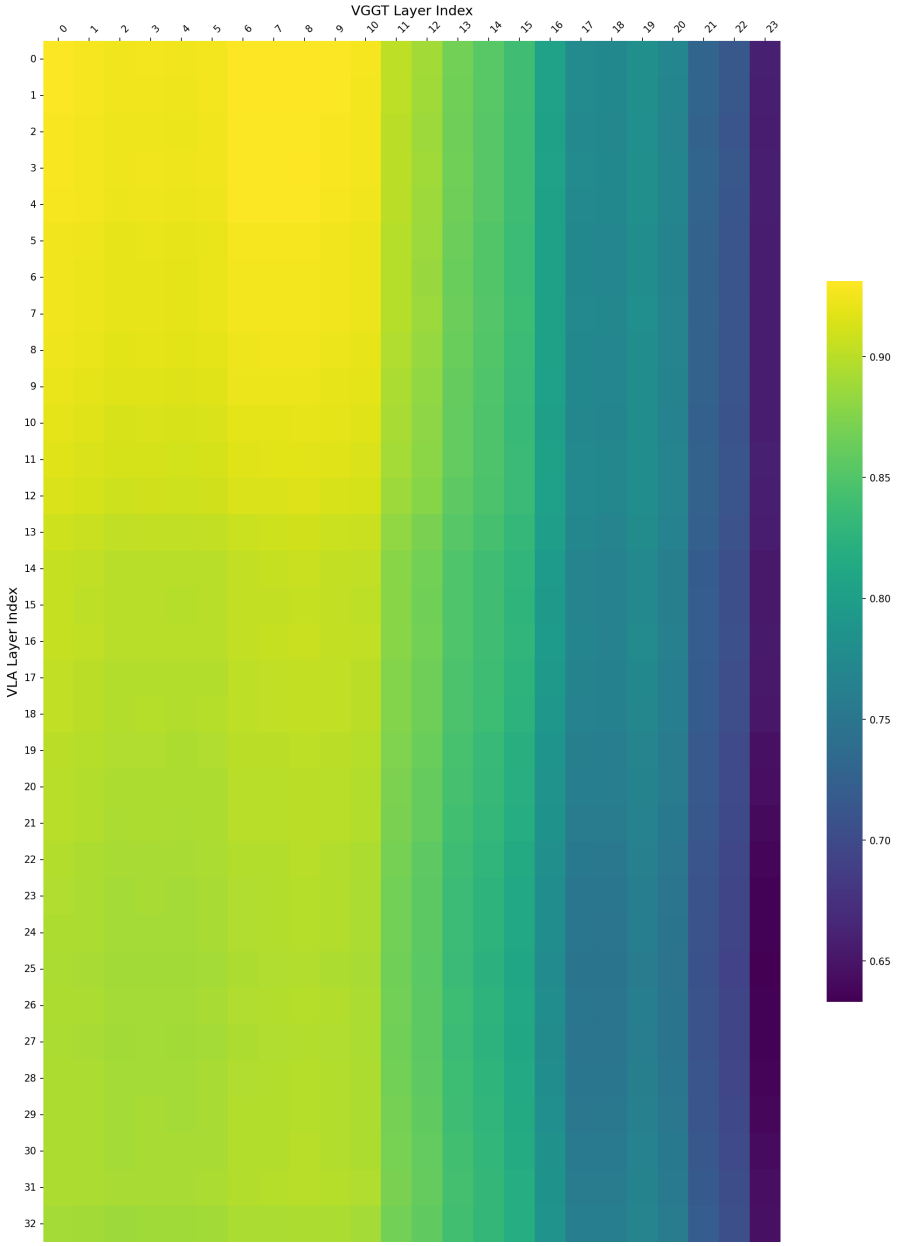


Figure 14: **Layer-wise CKA similarity between OpenVLA-7B and VGGT.** We compute linear CKA between OpenVLA-7B vision-token representations and VGGT visual features across layers. Higher values indicate more similar representations. Shallow VGGT layers exhibit higher similarity to OpenVLA-7B, and similarity generally decreases with depth.

In our implementation, we measure representation similarity between OpenVLA-7B (student) and VGGT (teacher) using *linear CKA* on a held-out set of samples. For each input batch, we extract the vision-token hidden states from selected layers of OpenVLA-7B and the corresponding visual features from selected layers of VGGT. Since the two models have different spatial resolutions and hidden dimensions, we first resample the VGGT features to match the spatial layout of OpenVLA's vision tokens, then flatten both representations into per-sample feature vectors and concatenate all samples to form matrices $X \in \mathbb{R}^{N \times D_x}$ and $Y \in \mathbb{R}^{N \times D_y}$. We then compute linear CKA for every layer pair by constructing and centering Gram matrices $K = XX^\top$ and $L = YY^\top$, and reporting the normalized HSIC score $\mathrm{CKA}(X, Y) = \langle \tilde{K}, \tilde{L} \rangle_F / (\|\tilde{K}\|_F \|\tilde{L}\|_F)$, which is comparable even when $D_x \neq D_y$.

I Per-task Results on LIBERO-Plus

Detailed per-task results corresponding to Fig. 5 are provided in Figs. 15–18.

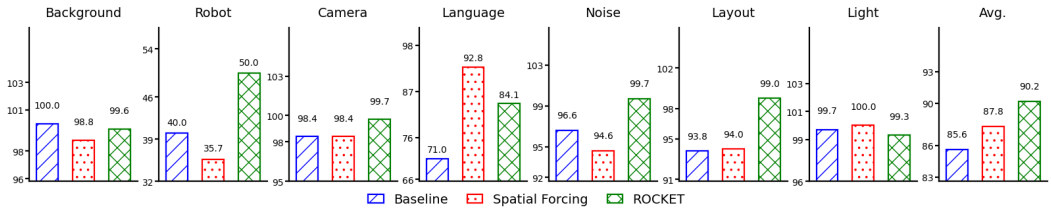


Figure 15: Per-task success rate (%) on LIBERO-Plus **Spatial** tasks under the seven perturbations.

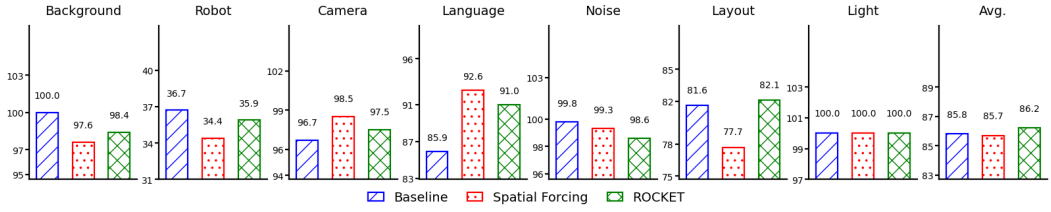


Figure 16: Per-task success rate (%) on LIBERO-Plus **Object** tasks under the seven perturbations.

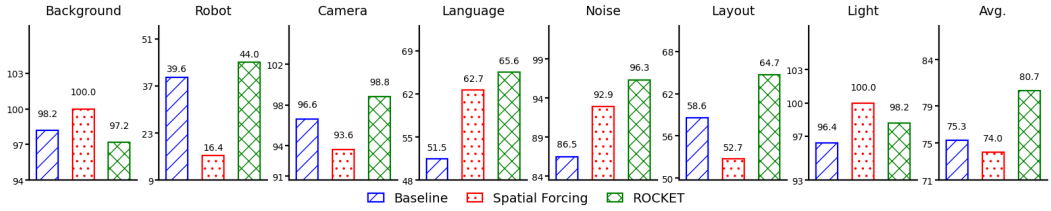


Figure 17: Per-task success rate (%) on LIBERO-Plus **Goal** tasks under the seven perturbations.

J Analysis: Why Shared Projectors Increase Cross-Layer Gradient Coherence

J.1 Takeaway: shared projector makes cross-layer gradients add constructively

This section explains why a *shared* nonlinear projector tends to yield more *cross-layer coherent* distillation gradients than *layer-specific* independent projectors in a Pre-LN Transformer. The argument has two steps: (i) under Pre-LN *residual-smallness*, back-propagation through the residual stream is close to identity, so the gradient at an earlier layer is approximately a superposition of

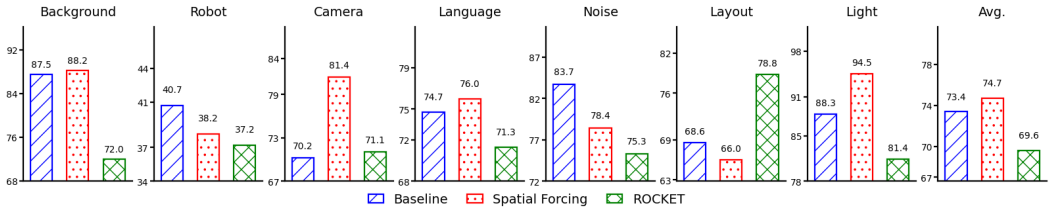


Figure 18: Per-task success rate (%) on LIBERO-Plus **Long** tasks under the seven perturbations.

future local distillation gradients; (ii) the stability of this superposition is governed by *cross-layer interference* terms, and a shared projector imposes a common operator structure on these terms (up to controlled deviation), whereas Multiple independent projectors do not. This analysis is a post-hoc explanation of the observed optimization behavior: it provides a sufficient condition under which sharing preserves cross-layer gradient coherence, rather than a claim that independent projectors must always fail.

J.2 Setup: Pre-LN residual stream, nonlinear projector, and distillation loss

Let the student network be a Pre-LN Transformer written on the residual stream:

$$\boxed{h_{l+1} \;=\; h_l \;+\; F_l(\mathrm{LN}(h_l);\theta_l)\,, \qquad l = 0,\dots,L-1} \tag{8}$$

where $h_l \in \mathbb{R}^d$ denotes the (flattened) residual representation at layer l, $\mathrm{LN}(\cdot)$ is LayerNorm, and θ_l are the block parameters. Define $u_l := \mathrm{LN}(h_l)$ and $\Delta_l := F_l(u_l;\theta_l)$ so that $h_{l+1} = h_l + \Delta_l$.

Nonlinear projector. For each distillation layer $l \in \mathcal{S}$, we attach a two-layer MLP projector (with input normalization LN_v):

$$\tilde{h}_l := \mathrm{LN}_v(h_l) \tag{9}$$

$$p_\phi(h_l) := W_2\,\sigma\Big(W_1\tilde{h}_l\Big)\,, \qquad \sigma = \mathrm{GELU}, \tag{10}$$

where $\phi = \{W_1, W_2\}$ (and potentially LN_v parameters).

Layerwise distillation loss. Let t_l denote the teacher representation used as the distillation target for the student's layer l, selected via a fixed layer mapping. We define the multi-layer distillation objective:

$$\mathcal{L}_{\mathrm{align}}(\theta,\{\phi_l\}) \;:=\; \sum_{l\in\mathcal{S}} \alpha_l\,\ell(z_l,t_l)\,, \qquad z_l := p_{\phi_l}(h_l(\theta)), \tag{11}$$

where $\ell(\cdot,\cdot)$ is any differentiable similarity loss (e.g., $1-\cos(\cdot,\cdot)$), and $\alpha_l \geq 0$ is the coefficient of the alignment loss; by default, we use uniform (mean) weighting.

J.3 Jacobians needed for gradient transport and projector-induced coupling

Pre-LN residual Jacobian. Differentiating (8) yields

$$\frac{\partial h_{l+1}}{\partial h_l} \;=\; I + A_l, \qquad A_l := \frac{\partial F_l(u_l;\theta_l)}{\partial u_l}\,J_{\mathrm{LN}}(h_l), \tag{12}$$

where $J_{\mathrm{LN}}(h_l)$ denotes the Jacobian of LN at h_l. Thus, for $i < l$,

$$\frac{\partial h_l}{\partial h_i} \;=\; \prod_{k=1}^{l-i}(I + A_{l-k}). \tag{13}$$

Projector Jacobian. Let $x := W_1\tilde{h}$ and $D_\sigma(x) := \mathrm{diag}(\sigma'(x))$. Then the Jacobian of (10) is

$$J_{p_\phi}(h) = W_2\,D_\sigma(W_1\tilde{h})\,W_1\,J_{\mathrm{LN}_v}(h), \tag{14}$$

where $J_{\mathrm{LN}_v}(h)$ is the Jacobian of LN_v.

J.4 Exact gradient at an early layer and its superposition approximation

This subsection writes the exact distillation gradient at h_i and then states the approximation that removes deep transport effects under Pre-LN residual-smallness.

For each $l \in \mathcal{S}$, define the local gradient from the distillation loss:

$$g_l \;:=\; \nabla_{z_l} \ell(z_l, t_l). \tag{15}$$

By the chain rule, the gradient at the student layer l is:

$$\nabla_{h_l} \ell(z_l, t_l) \;=\; J_{p_{\phi_l}}(h_l)^\top \, g_l. \tag{16}$$

For an earlier layer $i < l$, the gradient is back-propagated through the residual blocks. Note that for a product of matrices $M_{l-1} \dots M_i$, the transpose reverses the order to $M_i^\top \dots M_{l-1}^\top$. Thus, we have:

$$\nabla_{h_i} \ell(z_l, t_l) \;=\; \left(\prod_{k=i}^{l-1} (I + A_k)^\top \right) J_{p_{\phi_l}}(h_l)^\top \, g_l, \tag{17}$$

where the product is ordered from $k = i$ (leftmost) to $l - 1$ (rightmost). Summing over all distillation layers $l \geq i$ gives the total gradient at layer i:

$$\nabla_{h_i} \mathcal{L}_{\text{align}} \;=\; \sum_{l \in \mathcal{S}: l \geq i} \alpha_l \left(\prod_{k=i}^{l-1} (I + A_k)^\top \right) J_{p_{\phi_l}}(h_l)^\top \, g_l. \tag{18}$$

J.5 Residual-smallness implies near-identity transport

Assumption (Residual-smallness for Pre-LN). There exists $\varepsilon > 0$ such that for all k of interest,

$$\|A_k\| = \left\| \frac{\partial F_k(u_k; \theta_k)}{\partial u_k} \, J_{\text{LN}}(h_k) \right\| \leq \varepsilon. \tag{19}$$

Lemma 1 (Near-identity transport). Under (19), for any $i < l$, the back-propagation matrix remains close to the identity:

$$\left\| \prod_{k=i}^{l-1} (I + A_k)^\top - I \right\| \leq \exp\big((l-i)\varepsilon\big) - 1. \tag{20}$$

For small depth or residuals (i.e., $(l-i)\varepsilon \ll 1$), this bound is approximately $(l-i)\varepsilon$.

Corollary 1 (Gradient is a sum of local distillation gradients, up to transport error). Substituting the bound from Lemma 1 into (18), we obtain that the gradient at layer i is the superposition of future local gradients plus an error term dependent on depth:

$$\boxed{\nabla_{h_i} \mathcal{L}_{\text{align}} \;=\; \sum_{l \in \mathcal{S}: l \geq i} \alpha_l \, J_{p_{\phi_l}}(h_l)^\top \, g_l \;+\; \mathcal{E}_i} \tag{21}$$

where the error $\mathcal{E}_i$ is bounded by order $O\Big(\varepsilon \sum_{l \geq i} (l-i)\, \alpha_l \| J_{p_{\phi_l}}^\top g_l \|\Big)$.

J.6 Shared projector makes cross-layer interference constructive

By Corollary 1, the stability of the early-layer update is controlled by how the terms $J_{p_{\phi_l}}(h_l)^\top g_l$ interact across distillation layers. We now show that sharing the projector creates a *common operator* governing these interactions (up to controlled deviation), yielding a signal-aligned lower bound on cross-layer interference.

We compare two parameterization strategies for the auxiliary tasks:

- **Multiple independent projectors:** each $l \in \mathcal{S}$ has distinct parameters ϕ_l.
- **Shared projector:** a single set of parameters $\phi_l \equiv \phi$ is used for all $l \in \mathcal{S}$.

Gradient forms and the interference term. Recalling (21), the gradient at representation h_i is a superposition of terms. Let

$$v_l \;:=\; J_{p_{\phi_l}}(h_l)^\top g_l, \qquad G \;\approx\; \sum_{l\in\mathcal{S}:l\geq i} \alpha_l\, v_l. \tag{22}$$

The squared norm expands as

$$\|G\|_2^2 \;=\; \sum_l \alpha_l^2 \|v_l\|_2^2 \;+\; \sum_{a\neq b} \alpha_a\alpha_b\, \langle v_a, v_b\rangle, \tag{23}$$

where the cross terms $\langle v_a, v_b\rangle$ determine whether different distillation layers reinforce or cancel each other.

Proposition 1 (Cross-layer interference as a bilinear form on teacher-side error signals). For any $a, b \in \mathcal{S}$,

$$\boxed{\langle v_a, v_b\rangle \;=\; g_a^\top \underbrace{\left(J_{p_{\phi_a}}(h_a)\, J_{p_{\phi_b}}(h_b)^\top\right)}_{=:\mathcal{M}_{ab}} g_b} \tag{24}$$

Error-signal subspace. Let the *error-signal subspace* be

$$\mathcal{G} \;:=\; \mathrm{span}\{g_l :\; l\in\mathcal{S}\} \subseteq \mathbb{R}^{\dim(z)}. \tag{25}$$

All teacher-side signals that affect the student update through distillation lie in $\mathcal{G}$ by construction. Accordingly, we will only require geometric regularity of the shared projector on $\mathcal{G}$.

J.6.1 Case 1 (Shared projector): a common PSD operator governs cross-layer interference.

The advantage of shared projector is that cross-layer interactions are governed by (approximately) a *single* operator on the teacher-side error signals. The following assumptions control how the shared projector Jacobian varies along the residual stream, enabling us to relate $\mathcal{M}_{ab}$ to a fixed PSD reference operator.

Assumption (Bounded residual increments). There exists $\delta > 0$ such that for all layers l,

$$\|h_{l+1} - h_l\|_2 = \|\Delta_l\|_2 \leq \delta. \tag{26}$$

Assumption (Projector Jacobian Lipschitz). There exists $L_J > 0$ such that for all x, y in the bounded feature domain,

$$\|J_{p_\phi}(x) - J_{p_\phi}(y)\|_2 \leq L_J \|x - y\|_2. \tag{27}$$

Lemma 2 (Cross-layer Jacobian closeness under residual dynamics). Under (26) and (27), for any $a < b$,

$$\|J_{p_\phi}(h_b) - J_{p_\phi}(h_a)\|_2 \leq L_J\,(b-a)\,\delta. \tag{28}$$

To convert Jacobian closeness into a statement about alignment, we introduce an *on-subspace* near-isometry requirement on the teacher-side signals that actually occur.

Assumption (Near-isometry on the error-signal subspace). For a reference layer $\bar{l} \in \mathcal{S}$ and $\bar{h} := h_{\bar{l}}$, define

$$J \;:=\; J_{p_\phi}(\bar{h}), \qquad M \;:=\; JJ^\top \succeq 0. \tag{29}$$

Assume there exist constants $c > 0$ and $\eta \in [0, 1)$ such that for all $x \in \mathcal{G}$,

$$(1-\eta)c\,\|x\|_2^2 \;\leq\; x^\top M x \;\leq\; (1+\eta)c\,\|x\|_2^2. \tag{30}$$

Equivalently, M has condition number at most $\frac{1+\eta}{1-\eta}$ when restricted to $\mathcal{G}$.

Lemma 3 (Approximate inner-product preservation on $\mathcal{G}$). Under (30), for all $x, y \in \mathcal{G}$,

$$\left|x^\top M y - c\, x^\top y\right| \;\le\; \eta c \cdot \frac{\|x\|_2^2 + \|y\|_2^2}{2}. \tag{31}$$

Proof sketch. Use the polarization identity $x^\top M y = \frac{1}{4}\big((x+y)^\top M(x+y) - (x-y)^\top M(x-y)\big)$ and apply the quadratic bounds (30) to $x \pm y$.

Proposition 2 (Shared projector cross terms decompose into a common PSD part plus deviation). Let $\bar{h} = h_{\bar{l}}$ and decompose

$$J_{p_\phi}(h_l) = J + E_l, \qquad \|E_l\|_2 \le L_J |l - \bar{l}|\delta, \tag{32}$$

where the bound follows from Lemma 2. Then for any $a, b \in \mathcal{S}$ with $a, b \ge i$,

$$\langle v_a, v_b\rangle_{\text{share}} = g_a^\top \big(J_{p_\phi}(h_a) J_{p_\phi}(h_b)^\top\big) g_b = g_a^\top M g_b \;+\; \Delta_{ab}, \tag{33}$$

where the deviation satisfies

$$|\Delta_{ab}| \;\le\; \|g_a\|_2 \|g_b\|_2 \Big(\|E_a\|_2 \|J\|_2 + \|E_b\|_2 \|J\|_2 + \|E_a\|_2 \|E_b\|_2\Big). \tag{34}$$

Moreover, if $g_a, g_b \in \mathcal{G}$, then by Lemma 3,

$$\boxed{\langle v_a, v_b\rangle_{\text{share}} \;\ge\; c\, g_a^\top g_b \;-\; \eta c \cdot \frac{\|g_a\|_2^2 + \|g_b\|_2^2}{2} \;-\; |\Delta_{ab}|} \tag{35}$$

Interpretation. When (i) the teacher error signals are consistent across layers ($g_a^\top g_b$ positive and not too small), (ii) the shared projector is near-isometric on $\mathcal{G}$ (small η), and (iii) adjacent-layer Jacobians are close (small $\|E_l\|$, i.e., small $(|l - \bar{l}|\delta)$), the cross-layer interference term is biased toward being non-destructive and often constructive.

Theorem 1 (Shared projector induces coherent cross-layer interference under Pre-LN transport)**.** *Consider the Pre-LN student* (8) *with multi-layer distillation* (11)*. Under residual-smallness* (19)*, the gradient at an earlier representation h_i admits the superposition form* (21)*. For the shared projector parameterization ($\phi_l \equiv \phi$), assume bounded residual increments* (26)*, projector Jacobian Lipschitzness* (27)*, and near-isometry on the error-signal subspace* (30)*. Then for any $a, b \in \mathcal{S}$ with $a, b \ge i$ and $g_a, g_b \in \mathcal{G}$, the cross-layer interference term $\langle v_a, v_b\rangle$ in* (23) *satisfies the shared projector decomposition* (33) *with deviation bounded by* (34)*, and furthermore admits the signal-aligned lower bound* (35)*. Consequently, when teacher-side error signals are positively correlated across layers (large $g_a^\top g_b$) and the controlled-deviation terms are small, shared projector gradients exhibit reduced destructive interference and higher cross-layer coherence in the superposition* (22)*.*

J.6.2 Case 2 (Multiple independent projectors): lack of a uniform constructive lower bound.

For Multiple independent projectors, the cross term becomes

$$\langle v_a, v_b\rangle_{\text{multi}} = g_a^\top \big(J_{p_{\phi_a}}(h_a) J_{p_{\phi_b}}(h_b)^\top\big) g_b, \tag{36}$$

which does not reduce to a single PSD form $g_a^\top (J J^\top) g_b$ and therefore admits no analogue of (31) or (35) without additional coupling assumptions. A generic upper bound is

$$\boxed{|\langle v_a, v_b\rangle_{\text{multi}}| \;\le\; \|g_a\|_2 \|g_b\|_2 \, \|J_{p_{\phi_a}}(h_a)\|_2 \, \|J_{p_{\phi_b}}(h_b)\|_2} \tag{37}$$

but this bound does not preclude sign flips (destructive interference) since the bilinear form $J_{p_{\phi_a}}(h_a) J_{p_{\phi_b}}(h_b)^\top$ is not constrained to be PSD or even symmetric.

Empirical specialization and orthogonalization. In our experiments, we observe a consistent "decoupling" behavior: as training proceeds and the per-layer alignment losses saturate, different projectors tend to rotate their effective Jacobian row spaces into near-orthogonal subspaces. Concretely, if we measure the pairwise similarity between projector weights (or their local linearizations), e.g., $\cos(\mathrm{vec}(W_a), \mathrm{vec}(W_b))$ for linear heads or an analogous metric based on $J_{p_{\phi_l}}(h_l)$, the values decrease toward (and remain close to) zero in the early stage after training has stabilized, and this behavior persists consistently until the end of training (The visualization results are provided in Appendix G (Fig.13).). This indicates that, rather than enforcing a shared representation geometry

across layers, independent heads can absorb layer-specific discrepancies by specializing into mutually orthogonal directions, which suppresses cross-layer coupling in (36). In Fig. 1 (top), we show the cosine similarity between the gradients induced by different alignment losses at an early layer (the projector between the vision tower and the LLM backbone) during the early stage of training and the mid-to-late stage. For the 10 layers, we compute the pairwise cosine similarities among their gradients. The visualization shows that the gradients in the early stage are nearly perfectly orthogonal, indicating substantial training fluctuations; in the mid-to-late stage, this low cosine similarity persists. In Fig. 1 (bottom), we visualize the corresponding results for ROCKET. From the very early stage of training to the mid-to-late stage, the shared-projector mechanism in ROCKET keeps the gradients induced by different alignment losses consistently aligned. This directional consistency holds throughout training, and the resulting mutual reinforcement among gradients contributes to the superior performance of ROCKET. Specifically, Fig. 19 shows the gradient similarities in the early stage of training, while Fig. 20 shows the gradient similarities in the mid-to-late stage.

As a consequence, even when the teacher error signals are correlated across layers (i.e., $g_a^\top g_b > 0$), the induced student-side gradients v_a and v_b can become weakly correlated because the interaction matrix $J_{p_{\phi_a}}(h_a)J_{p_{\phi_b}}(h_b)^\top$ itself shrinks in the relevant directions. In practice, this often yields a small average cross term $\mathbb{E}[\langle v_a, v_b\rangle_{\mathrm{multi}}]$ and can even lead to destructive interference when the effective interaction changes sign across mini-batches.

Consequence for gradient stability. Plugging (35) (shared projector) into (23) shows that the second summation in (23) admits a *signal-aligned* lower bound (up to controlled errors), while Multiple independent projectors admit only the trivial magnitude bound (37). Thus, under the same transport approximation of Corollary 1, shared projector gradients tend to exhibit higher cross-layer coherence and reduced destructive interference, consistent with empirical observations.

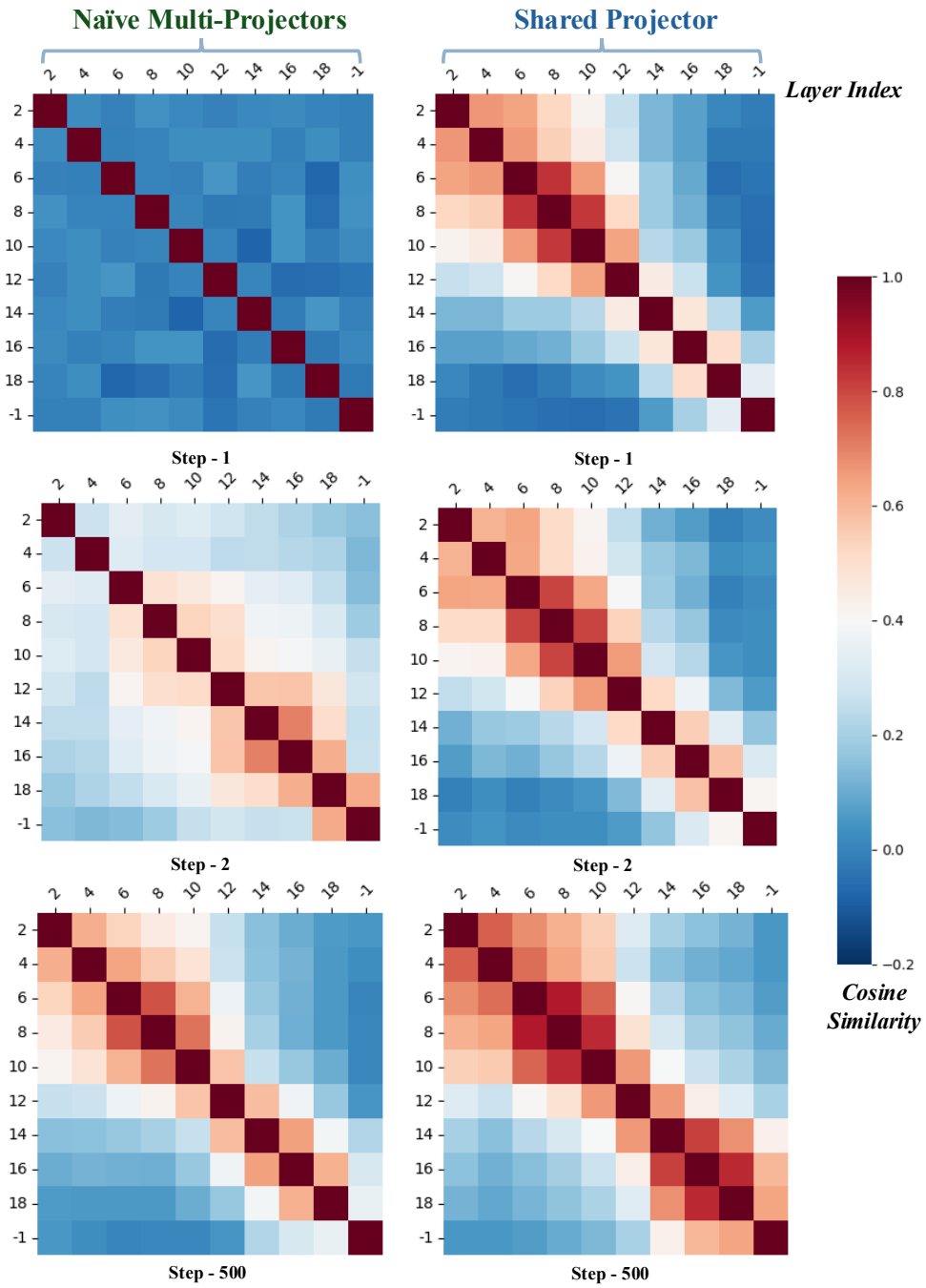


Figure 19: Gradient cosine similarity in the early stage of training.

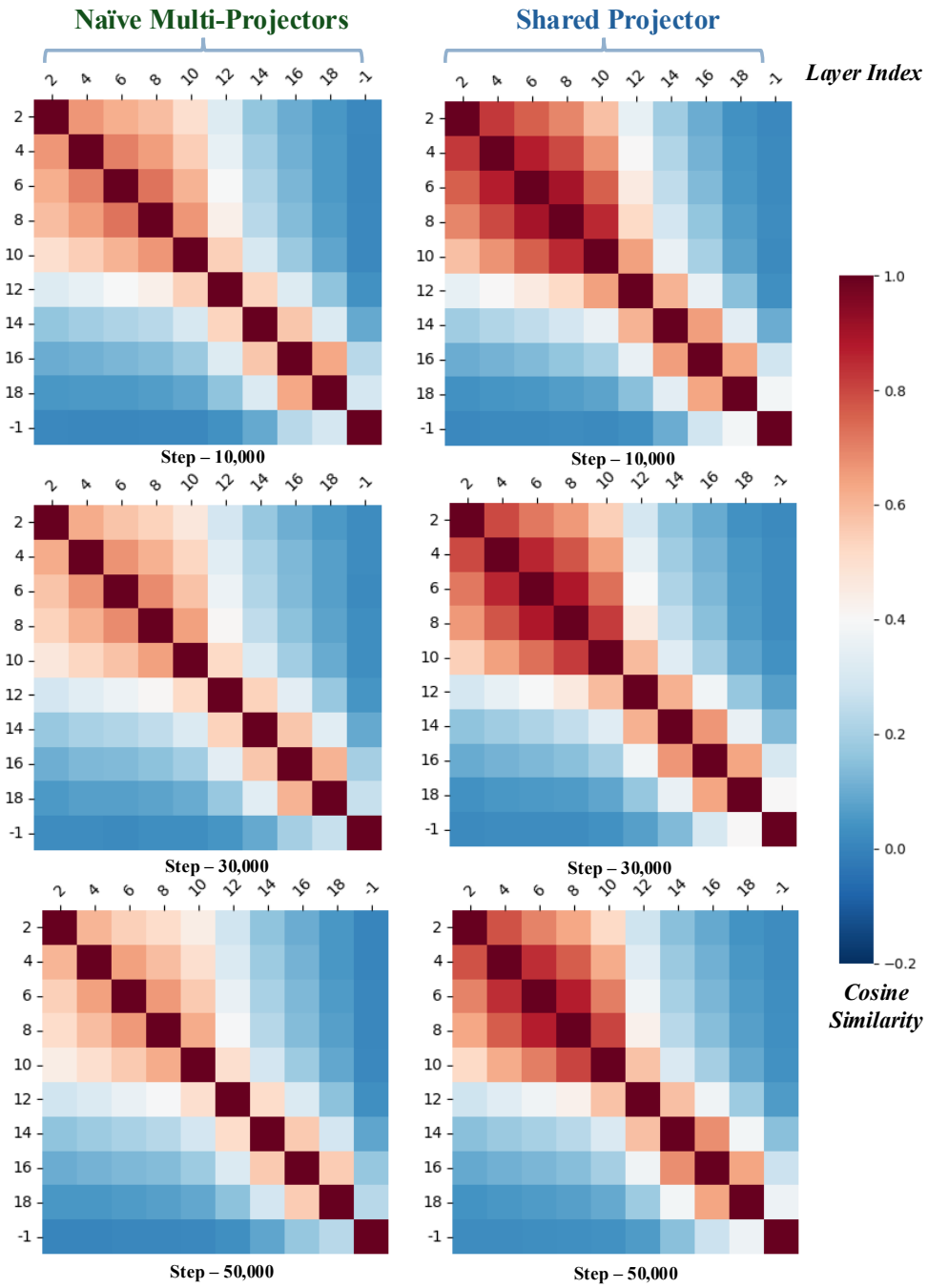


Figure 20: Gradient cosine similarity in the mid-to-late stage of training.

K Details of Layer Selection Strategies

K.1 Similarity-based Layer Selection

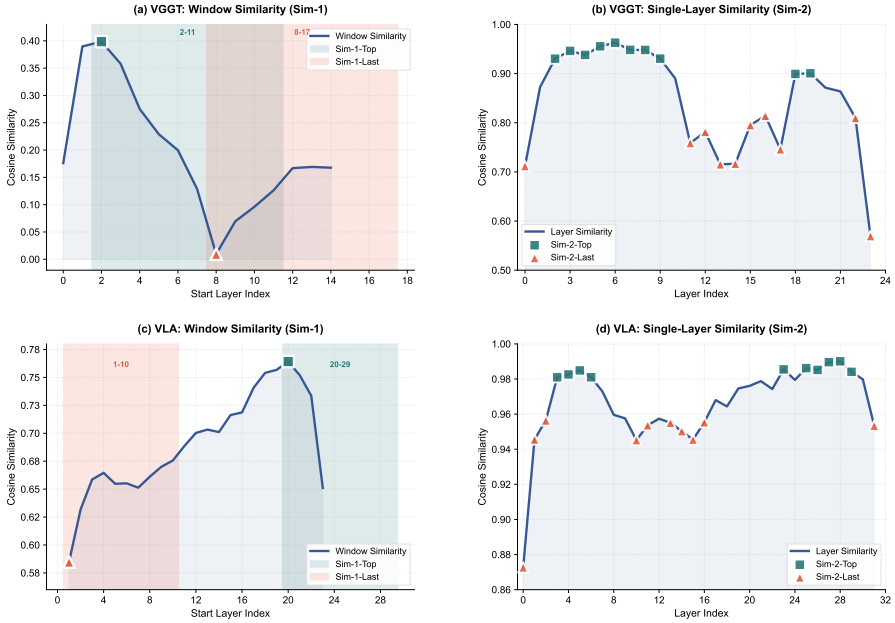


Figure 21: **Similarity-based layer selection strategies.** We evaluate two criteria for selecting alignment layers: **Sim-1** (left) measures 10-layer window similarity by computing cosine similarity between layer i's input and layer $i+9$'s output; **Sim-2** (right) measures single-layer similarity between each layer's input and output. For each criterion, we select 10 layers with either highest similarity (**Top**, green squares) or lowest similarity (**Last**, orange triangles).

Figure 21 shows that VGGT (24 layers, top row) and VLA (32 layers, bottom row) reveal distinct similarity distributions across the network depth. Lower similarity indicates greater information transformation, potentially identifying more informative layers for alignment.

K.2 Detailed Chosen indices for All Layer Selection Strategies

Strategy	# Layers	VLA Layers	VGGT Layers
Uniform Sampling			
Uniform-4	4	8, 16, 24, 32	5, 11, 17, 23
Uniform-8	8	4, 8, 12, 16, 20, 24, 28, 32	2, 5, 8, 11, 14, 17, 20, 23
Uniform-16	16	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23
Uniform-24	24	2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15, 16, 18, 19, 20, 22, 23, 24, 26, 27, 28, 30, 31, 32	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
Depth-based Sampling			
Shallow	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Middle	10	12, 13, 14, 15, 16, 17, 18, 19, 20, 21	7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Deep	10	23, 24, 25, 26, 27, 28, 29, 30, 31, 32	14, 15, 16, 17, 18, 19, 20, 21, 22, 23
Even-to-Max (E2M) Sampling			
$E2M\text{-}Last1^{*}$	10	2, 4, 6, 8, 10, 12, 14, 16, 18, 32	2, 4, 6, 8, 10, 12, 14, 16, 18, 23
$E2M\text{-}Last1^{\dagger}$	10	2, 3, 4, 5, 6, 7, 8, 9, 10, 18	2, 4, 6, 8, 10, 12, 14, 16, 18, 23
Similarity-based Sampling			
Sim-1-Top	10	20, 21, 22, 23, 24, 25, 26, 27, 28, 29	2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Sim-1-Last	10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	8, 9, 10, 11, 12, 13, 14, 15, 16, 17
Sim-2-Top	10	4, 5, 6, 7, 24, 26, 27, 28, 29, 30	2, 3, 4, 5, 6, 7, 8, 9, 18, 19
Sim-2-Last	10	1, 2, 3, 11, 12, 14, 15, 16, 17, 32	0, 11, 12, 13, 14, 15, 16, 17, 22, 23

Table 18: Detailed layer selection for multi-layer alignment. We show the aligned layer indices for both the student VLA visual backbone and the teacher VGGT model. For OpenVLA, we count the projector between vision tower and LLM as a layer so there are total 33 indices in our default setting. The selected layers are sorted in ascending order for both VLA and VGGT. * For OpenVLA (32 layers, or 33 including the projector), we select layers uniformly using indices 2–18 plus the last layer. † For PI0 and PI0.5 (18 layers), since layer 18 coincides with the last layer, we instead select indices 2–10 plus the last layer.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions (shared projector, Matryoshka activation, theoretical analysis) and the experimental scope (LIBERO, LIBERO-Plus, RoboTwin 2.0, real-world deployment), all of which are supported by the results in Sections 5–7.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: The paper does not include a dedicated limitations section. However, the real-world failure analysis (Section 6 and Appendix) identifies that fine-grained intra-object reasoning (Case 3) remains an open challenge for all methods, which partially addresses limitations of the current approach.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results are stated with explicit assumptions and proved in full in Appendix J. Section 3 provides proof sketches and intuition in the main text.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Training hyperparameters, layer indices, projector architecture, and dataset details are fully specified in the Appendix. All simulation benchmarks used are publicly available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code and model weights are publicly available at `https://github.com/CASE-Lab-UMD/ROCKET-VLA`, including the training scripts for the main experiments. All simulation benchmarks are publicly available; the real-world dataset was collected internally.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 summarizes the experimental setup; full details including learning rate schedules, batch sizes, training steps, and layer indices are provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following common practice in VLA benchmarking, we report success rates without error bars for simulation experiments. For real-world experiments, we report per-run results across 3 independent runs but do not compute standard deviations due to the limited number of runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A reports the GPU type (NVIDIA H200) and the full training configurations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. All experiments use publicly available benchmarks or internally collected data with proper safety protocols for real-robot operation.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This work is foundational research on improving spatial reasoning in robotic manipulation models. It does not have a direct path to negative societal applications beyond general-purpose robotics.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released models are task-specific robotic manipulation policies with no risk for misuse beyond their intended application.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (LIBERO, LIBERO-Plus, RoboTwin 2.0) and models (OpenVLA, PI0, PI0.5, VGGT) are properly cited. We use these assets in accordance with their respective licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the code and the trained model weights of `ROCKET` at `https://github.com/CASE-Lab-UMD/ROCKET-VLA`; the repository documents environment setup, data preparation, training, and evaluation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the core method. The VLA models fine-tuned in this work use pre-existing LLM backbones in their standard configuration.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.