GATED DEPTH ATTENTION: PRESERVING VISION-LANGUAGE UNDERSTANDING FOR GENERALIST ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) underpin modern vision-language-action (VLA) policies, yet remain fragile on spatially grounded queries and manipulation scenarios where 3D geometry matters. Prior 3D enhancements typically either align visual representations to geometry-centric teachers or explicitly inject depth/geometry tokens, both of which can perturb the pretrained RGB–language interface. We propose Gated Depth Attention (GDA), a lightweight module that distills a latent geometry branch from a frozen 3D foundation teacher and fuses it as a gated residual. GDA retrieves geometry through cross-attention while a learned per-channel gate controls how strongly the retrieved correction modifies each visual token; the external 3D teacher is removed at inference. Across VLM benchmarks spanning relational grounding, VQA/OCR, and robustness, GDA consistently outperforms alignment and concatenation baselines and matches or improves strong RGB-only models, yielding a better trade-off between geometry sensitivity and generic vision-language utility. When transferred to VLA policies, GDA also improves manipulation success in LIBERO, RoboTwin 2.0, and targeted real-world spatial/contact tasks. Teacher-swap experiments further preserve the benefit across two distinct 3D foundation teachers. These results suggest that stronger 3D supervision does not automatically preserve vision-language compatibility, and that geometry is better integrated as a selective refinement than as an always-on replacement.

1 INTRODUCTION

Vision-language models (VLMs) have become the perception backbone for modern vision-language-action (VLA) policies, enabling robots to interpret language instructions and act from visual observations (Brohan et al., 2022; Zitkovich et al., 2023; Black et al., 2024; Kim et al., 2024; Chen et al., 2025a). However, most VLM backbones are pretrained primarily on 2D image-text data (Liu et al., 2023c; Karamcheti et al., 2024; Dosovitskiy et al., 2020), and they remain brittle on spatially grounded queries and manipulation scenarios where depth and geometry are essential, such as "pick the middle bottle" or "place the object behind the cup." A natural solution is to leverage 3D foundation models that produce geometry-centric representations from images (Cheng et al., 2024; Wang et al., 2025; Lin et al., 2025a). The challenge is that these representations are typically optimized for reconstruction, correspondence, or depth prediction rather than recognition semantics, and can therefore interfere with the pretrained vision-language interface that makes VLMs useful.

Recent 3D-enhanced VLM/VLA methods commonly follow two routes. One route injects geometry explicitly, for example by concatenating depth maps, point clouds, or geometry tokens with RGB tokens and relying on a connector or fusion module to learn a joint image-geometry-language space (Lin et al., 2025b; Qu et al., 2025; Yuan et al., 2025). Another route transfers geometry implicitly by aligning intermediate visual representations to geometry-centric teacher features, as in Spatial Forcing (Li et al., 2025). Both approaches are intuitive and can improve spatial behavior, but they optimize for spatial enrichment rather than preservation of the pretrained RGB-to-text pathway. Explicit fusion forces the model to co-align RGB, geometry, and language, while representation alignment can reshape semantic visual features toward a geometry target. Thus, the key question is not only whether a model can absorb more 3D structure, but whether it can do so *without unnecessarily perturbing vision-language compatibility*.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This paper studies that trade-off. We find that common 3D integration strategies can improve spatial bias while degrading generic VLM behavior on tasks such as grounding, VQA (Goyal et al., 2017), OCR (Singh et al., 2019), hallucination-sensitive evaluation (Li et al., 2023b), and counting (Acharya et al., 2019). This suggests that geometry is useful but not universally beneficial: for many prompts, depth is a nuisance signal rather than necessary evidence. Motivated by this observation, we propose Gated Depth Attention (GDA), a lightweight fusion module that treats geometry as a *gated residual* on top of the projected RGB tokens. Instead of aligning RGB representations to 3D teacher embeddings or concatenating geometry tokens unconditionally, GDA first retrieves geometry through cross-attention and then uses a learned gate to modulate its residual effect on the visual tokens. During training, a frozen 3D teacher supervises a latent geometry branch via distillation; at inference, the teacher is removed and geometry tokens are predicted directly from RGB features. Importantly, our default gate is image-conditioned before multimodal grounding: geometry routing is treated as a visual filtering problem, while instruction-conditioned use of the resulting tokens is left to the LLM after image-text interaction. This design preserves the visual token count and LLM input format while allowing geometry-aware corrections only when supported by the visual context.

Across a diverse suite of VLM benchmarks, GDA improves spatial grounding relative to the tested 3D-integration methods while matching or improving RGB-only baselines on generic vision–language compatibility, and improves robustness-sensitive utility in our default DINO+SigLIP configuration. These VLM-side gains translate to stronger embodied control on LIBERO (Liu et al., 2023a), RoboTwin 2.0 (Chen et al., 2025b), and targeted real-world manipulation, without requiring depth sensors at test time. Mechanistic analyses further show that the benefit persists across different 3D teachers. Ablations also show that gating is necessary to retain generic vision–language understanding and that conditioning the gate on text *before* image–text grounding is counterproductive, supporting conservative visual routing followed by instruction-conditioned reasoning in the LLM.

Our contributions are:

- We identify and empirically characterize a failure mode in 3D-enhanced VLM/VLA design: stronger spatial supervision does not guarantee preservation of vision-language compatibility, and common geometry-integration strategies can degrade generic VLM utility.
- We introduce GDA, a gated residual fusion module that uses 3D foundation features as selective visual refinements rather than always-on replacements for RGB semantics.
- We evaluate GDA across VLM benchmarks, VLA simulation, and real-world manipulation, with mechanism-level validation through gate interventions, direct-teacher controls, cross-teacher transfer, and robustness analyses, together with two capacity-matched controls that respectively match and fall below the RGB-only baseline, showing that the gain requires coherent supervision from the 3D teacher rather than the added capacity or an auxiliary loss alone.

2 Related Work

Vision-language connectors. Modern VLMs connect a vision encoder to an LLM through projection modules or attention-based interfaces. LLaVA-style models use linear or MLP projectors followed by instruction tuning (Liu et al., 2023c), BLIP-2 learns a Q-Former bridge (Li et al., 2023a), and Flamingo inserts gated cross-attention layers into a frozen LLM (Alayrac et al., 2022). Our method follows the projection-based philosophy: it preserves the LLM interface and modifies only the visual tokens supplied to the LLM.

Gating for multimodal fusion. Gating and conditional modulation have a long history in multimodal learning, including FiLM (Perez et al., 2018), Gated Multimodal Units (Arevalo et al., 2017), and gated cross-attention in Flamingo (Alayrac et al., 2022). In contrast to methods that gate signals entering the language stream, GDA uses a visual gate before the LLM to control geometry residuals.

Geometry in VLA. Explicit 3D VLA methods inject depth, point clouds, or geometry tokens into the policy input (Lin et al., 2025b; Qu et al., 2025; Yuan et al., 2025), while implicit methods such as Spatial Forcing align intermediate visual features to geometry-centric teacher representations (Li et al., 2025). GDA differs by keeping the RGB representation as the default pathway and using geometry only as a gated residual refinement.

3 METHOD

Our goal is to add geometry awareness to VLM backbones for VLA without disrupting the pretrained vision-language interface. Rather than aligning RGB features to geometry-centric teachers or concatenating geometry tokens unconditionally, GDA treats geometry as a gated residual over the RGB pathway. It retrieves geometry through cross-attention and uses a learned gate to control how strongly the correction is admitted along each feature channel, keeping the pre-LLM interface close to the RGB-only representation. Figure 1 illustrates the resulting pipeline. A formal discussion of potential perturbations from alignment-based and unconditional fusion is provided in Appendix C.1. Diagnostic baselines are described in Appendices C.2.1 and C.2.2, with DINO-SigLIP backbone details in Appendix C.2.4 (Oquab et al., 2023; Zhai et al., 2023).

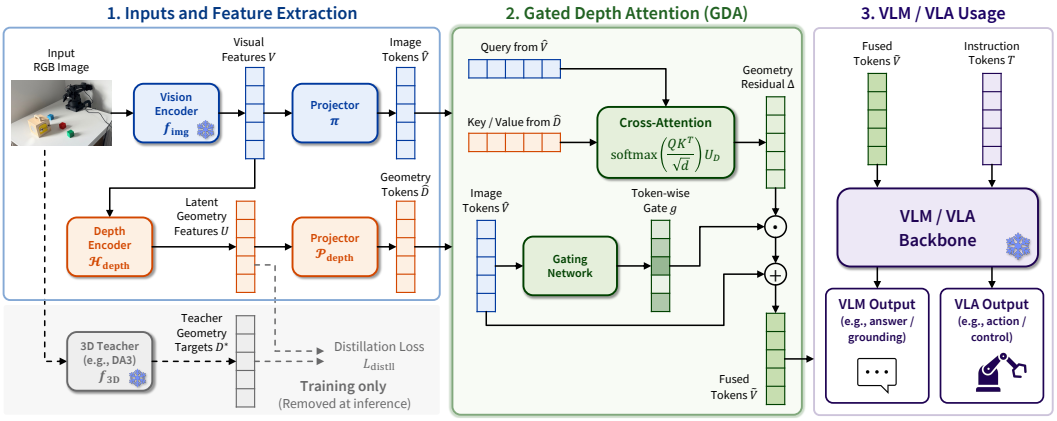

Figure 1: **Overview of our framework.** Given an input image, the model extracts semantic features through a standard vision encoder and geometry-aware features through a teacher-supervised latent depth pathway. The proposed GDA module first retrieves geometry information via cross-attention, producing a residual update Δ, and then modulates its injection using a learnable gate $g \in [0, 1]$ predicted from the visual context.

3.1 PRELIMINARIES: VISION-LANGUAGE MODELS

Given an image I and an instruction x, a vision encoder f_{img} produces visual tokens $V = f_{\text{img}}(I) \in \mathbb{R}^{N \times d_v}$, which a multimodal projector $\pi(\cdot)$ maps to the LLM embedding space as $\hat{V} = \pi(V) \in \mathbb{R}^{N \times d}$. The instruction is embedded as $T = \text{Embed}(x) \in \mathbb{R}^{M \times d}$, and the VLM generates $Y \sim p_\theta(\cdot \mid [\hat{V}; T])$. Our design modifies only the visual tokens entering the LLM, leaving the language pathway, the visual token count, and the LLM input format unchanged.

3.2 GEOMETRY TEACHER AND LATENT DEPTH BRANCH

During training, we use a frozen 3D foundation teacher, DA3, to provide geometry-oriented supervision (Lin et al., 2025a). Unless otherwise stated, all experiments in the main comparison use DA3 as the frozen 3D teacher. Our formulation only assumes patch-aligned geometry targets and is not tied to a specific 3D teacher; we verify this with a DA3↔VGGT teacher swap on both the VLM and manipulation axes (Appendix E.5). The teacher produces patch-aligned geometry targets $D^* = f_{\text{3D}}(I) \in \mathbb{R}^{N \times d_u}$.

On top of the same visual features V, we introduce a lightweight latent depth branch:

$$U = \mathcal{H}_{\text{depth}}(V) \in \mathbb{R}^{N \times d_u}, \qquad \hat{D} = \mathcal{P}_{\text{depth}}(U) \in \mathbb{R}^{N \times d}, \tag{1}$$

where $\hat{D}$ denotes geometry tokens projected into the same LLM embedding space as $\hat{V}$. We supervise the latent depth branch by distilling U toward the frozen teacher targets: $\mathcal{L}_{\text{distill}} = \ell(U, D^*)$, where ℓ is mean-squared error or cosine distance. For the default DINO+SigLIP configuration, $\mathcal{H}_{\text{depth}}$ is a single Linear$(1024 \to 1024)$ layer applied to the DINO branch; U is distilled toward the frozen

teacher's penultimate-block patch features using cosine distance, and $\mathcal{P}_{\text{depth}}$ is a two-layer MLP $(1024 \to 896 \to 896)$ with GELU between the layers. The dimensions are adjusted analogously for other backbones. At inference time, the external 3D teacher is removed, and $\hat{D}$ is predicted directly from RGB features.

3.3 Gated Depth Attention

Given RGB tokens $\hat{V}$ and predicted geometry tokens $\hat{D}$, we first compute a geometry residual through cross-attention from image tokens to geometry tokens: $A = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right), \Delta = AU_D$, where $Q = \hat{V}W_Q$, $K = \hat{D}W_K$, $U_D = \hat{D}W_V$, and $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are learnable parameters. Here, cross-attention determines *what* geometry information is retrieved for each visual token. However, attention alone does not control how strongly the retrieved geometry signal modifies the RGB representation. To preserve the pretrained RGB pathway when geometry is unnecessary or unreliable, we introduce an explicit gate on top of the attention-derived residual.

In the implementation, the gate is per-token and per-channel and observes both the RGB token and the correction it modulates. Let $C = [\hat{V}; \Delta] \in \mathbb{R}^{N\times 2d}$. We compute

$$g = \sigma(\mathrm{LN}(C)W_g + b_g) \in [0,1]^{N\times d}, \qquad \tilde{V} = \hat{V} + g \odot \Delta, \tag{2}$$

where $W_g \in \mathbb{R}^{2d\times d}$ and $b_g \in \mathbb{R}^d$; we write $\mathcal{G}$ for this gate map. Thus, $\mathcal{G}$ is a single linear map after LayerNorm, with no hidden layer or ReLU/GELU inside the gate; its nonlinearity comes from that LayerNorm and the outer sigmoid, which also supplies the bounded multiplicative modulation. In the default $d = 896$ configuration, this is Linear$(1792 \to 896)$. This explicit form is important because the gate modulates individual feature channels rather than a single scalar per token. Thus geometry enters as an attenuated refinement: $\tilde{V}$ stays close to the original RGB-only representation where the gate is near zero, and Δ refines the feature channels the gate admits.

Image-conditioned gate. Our default gate is image-conditioned in the forward pass: for a fixed image, it does not vary with the text prompt. We adopt this design to keep GDA as a visual adapter and avoid prompt-conditioned routing before image-text grounding. Nevertheless, the gate is trained end-to-end under the instruction-following objective $p_\theta(\cdot \mid [\tilde{V}; T])$, allowing it to learn where geometry-aware correction is useful under the training instruction distribution. Empirically, an explicitly instruction-conditioned gate underperforms the default image-conditioned gate (Section 4.4, Table 4), suggesting that in this pre-grounding position the gate is better conditioned on visual features, with instruction-conditioned use deferred to the LLM after multimodal grounding. Thus, the default gate acts as a conservative visual filter rather than a prompt-conditioned router, controlling which geometry-enriched visual tokens are exposed to the LLM. Although g is computed per token and per channel for every input, our analysis in Appendix E.3 shows that its learned selectivity is predominantly channel-wise rather than driven by large scene-level or local-texture changes. We therefore use "image-conditioned" to describe the forward computation, but do not interpret the gate as a strongly scene-adaptive scalar router.

Initialization for RGB-behavior preservation. To preserve pretrained RGB behavior at the start of training, we initialize the single gate linear layer with exactly zero weights and bias -2, so every gate channel starts uniformly at $\sigma(-2) \approx 0.119$. This matters methodologically: any channel structure measured after training (Appendix E.3) is necessarily learned rather than inherited from initialization. This attenuates the initial geometry residual to $\sigma(-2)\Delta$ rather than making it zero. At a fixed model checkpoint, the gated residual also provides a quantitative notion of interface preservation: under a finite local logit sensitivity L, the next-token distributions with and without fusion satisfy $\mathrm{KL}(p_0 \| p_1) \le L^2 \|g \odot \Delta\|_F^2 / 4$, with exact recovery of the bypassed RGB pathway when $g = 0$. Appendix D proves this result and further bounds the propagation of geometry-feature error through both attention and the residual-aware gate.

3.4 Training Objective and Usage Regimes

We optimize GDA using a task loss together with geometry distillation (Hinton et al., 2015):

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{distill}} \mathcal{L}_{\text{distill}}. \tag{3}$$

During VLM pretraining, $\mathcal{L}_{\text{task}}$ is the standard instruction-following negative log-likelihood:

$$\mathcal{L}_{\text{VLM}} = -\sum_{t} \log p_\theta(y_t \mid y_{<t}, [\tilde{V}; T]). \tag{4}$$

During downstream VLA fine-tuning, the same GDA-augmented visual backbone is retained and $\mathcal{L}_{\text{task}}$ becomes the action or policy loss used by the corresponding base VLA. This makes GDA agnostic to the training stage; the two regimes (*pretraining+FT* and *FT-only*) are specified in Section 4.1. In both cases, the inference-time architecture is identical: the model consumes RGB input only, predicts latent geometry tokens internally, and injects them through gated residual fusion.

4 Experiments

4.1 Experimental Setup

Training regimes. We evaluate GDA under two training regimes. For LLaVA/OpenVLA-style models, multimodal VLM pretraining data are available, so we train GDA during VLM pretraining and then reuse the same module during downstream VLA fine-tuning (*pretraining+FT*). For π-based models, we start from an existing pretrained multimodal backbone and introduce GDA only during downstream policy fine-tuning (*FT-only*). All controlled comparisons within a model family match training regime, data, and compute budget; differences therefore isolate the fusion mechanism. Training settings, matched budgets, frozen/trainable modules, and training/evaluation latency are summarized in Appendix C.4.

Benchmarks and metrics. For the multimodal backbone, we report accuracy (%) on referring expression comprehension (RefCOCO, RefCOCO+, RefCOCOg) (Kazemzadeh et al., 2014; Yu et al., 2016; Mao et al., 2016), compositional and open-domain VQA (GQA, VQA-v2) (Hudson & Manning, 2019; Goyal et al., 2017), hallucination (POPE) (Li et al., 2023b), counting (TallyQA) (Acharya et al., 2019), OCR-heavy VQA (TextVQA) (Singh et al., 2019), assistive VQA (VizWiz) (Gurari et al., 2018), and spatial relation reasoning (VSR) (Liu et al., 2023b). RefCOCO-family and VSR primarily test 2D/relational rather than metric 3D reasoning; we therefore use them to assess language-facing compatibility and relational behavior, while metric geometry is evaluated through the NYUv2 probe in Appendix C.2.3 and embodied manipulation in Section 4.3. For embodied policy evaluation, we report success rate (SR) on LIBERO (Liu et al., 2023a), RoboTwin 2.0 (Chen et al., 2025b), and a compact real-world manipulation benchmark. For VLA benchmarks, success rate is computed over repeated independent trials. Suite composition and the matched base-policy, data, and training-budget protocol used in every controlled comparison are detailed in Appendix F.1.

Compared methods. Unless otherwise stated, VLM experiments use Qwen2.5-0.5B with DINO-B, DINO-L, or DINO+SigLIP vision encoders. We compare GDA against representative geometry-integration baselines, including EVO-0 (Lin et al., 2025b) and Spatial Forcing (Li et al., 2025), whose fusion strategies are analyzed in Appendix C.1. We also include two diagnostic baselines to better isolate the source of degradation: DA3 replacement (Lin et al., 2025a) and Dualhead-SF (Appendix C.2.1). DA3 replacement tests whether a geometry-centric encoder can directly serve as the vision-language interface, while Dualhead-SF tests whether preserving a separate RGB-oriented branch is sufficient when geometry is still fused unconditionally, i.e., without the proposed gated residual module.

4.2 VLM Benchmark Results

All results in this subsection evaluate the multimodal backbone after VLM pretraining and before downstream policy fine-tuning, thereby isolating the effect of geometry fusion at the VLM stage. Unless otherwise stated, all methods use the same single-stage full-finetune recipe, in which the vision encoder is trained; the frozen components are the external 3D teacher only (Appendix C.4). Following the Prismatic/OpenVLA-style visual-connector recipe (Karamcheti et al., 2024; Kim et al., 2024; 2025), we use Qwen2.5 to enable controlled evaluation across VLM and VLA tasks and across 0.5B/1.5B/7B model scales; full implementation details are provided in Appendix C.4. For DINO+SigLIP, the default GDA model uses $\lambda_{\text{distill}} = 0.1$, selected by the sweep in Appendix E.1, while Spatial Forcing follows its original distillation setting.

Table 1: Comparison of depth fusion strategies on VLM benchmarks using Qwen2.5-0.5B. We compare an RGB-only baseline (Pure VLM), explicit geometry fusion (DA3 replacement and EVO-0 (Lin et al., 2025b)), implicit geometry alignment (Spatial Forcing (Li et al., 2025) and Dualhead-SF), and gated fusion (GDA). S-Avg averages RefCOCO, RefCOCO+, RefCOCOg, and VSR; VL-Avg averages GQA, TextVQA, VizWiz, and VQA-v2; R-Avg averages POPE and TallyQA.

Backbone	Method	Spatial Grounding					Generic VL Compatibility					Robustness			Avg.
		RefCOCO	RefCOCO+	RefCOCOg	VSR	S-Avg	GQA	TextVQA	VizWiz	VQA-v2	VL-Avg	POPE	TallyQA	R-Avg	
DINO-B	Pure VLM	13.5	9.6	13.3	48.5	21.2	45.3	**32.1**	31.2	**60.1**	42.2	**83.2**	**43.0**	**63.1**	38.0
	DA3	7.7	3.8	6.3	48.5	16.6	36.1	30.3	32.2	43.6	35.6	57.2	25.7	41.5	29.1
	EVO-0	9.8	6.1	9.5	47.4	18.2	40.6	30.4	26.4	50.3	36.9	71.6	33.7	52.7	32.6
	Spatial Forcing	7.2	4.1	5.2	48.5	16.3	36.9	28.8	22.3	43.6	32.9	58.8	19.6	39.2	27.5
	Dualhead-SF	7.9	4.1	6.3	48.5	16.7	36.5	30.8	28.4	43.3	34.8	58.6	30.2	44.4	29.5
	GDA	**14.8**	**10.9**	**14.6**	**49.3**	**22.4**	**46.4**	31.1	**38.4**	59.7	**43.9**	83.1	41.0	62.1	**38.9**
DINO-L	Pure VLM	14.6	10.7	15.5	48.5	22.3	**50.7**	30.0	31.3	**64.2**	44.1	**85.2**	**44.8**	**65.0**	39.6
	DA3	8.3	5.0	6.6	48.5	17.1	36.7	28.7	28.9	43.6	34.5	55.0	20.7	37.9	28.2
	EVO-0	11.1	7.3	10.8	48.5	19.4	43.1	30.1	31.4	53.4	39.5	77.8	31.8	54.8	34.5
	Spatial Forcing	5.8	3.5	4.4	48.5	15.6	37.0	27.9	26.7	44.4	34.0	58.3	16.6	37.5	27.3
	Dualhead-SF	5.9	3.3	4.4	48.7	15.6	36.3	29.5	31.2	43.4	35.1	57.6	30.9	44.3	29.1
	GDA	**18.4**	**14.9**	**18.1**	**51.6**	**25.8**	48.6	**30.2**	**40.0**	63.2	**45.5**	84.0	41.6	62.8	**41.1**
DINO +SigLIP	Pure VLM	17.0	12.4	**19.5**	48.5	24.4	53.1	**40.1**	39.6	**69.1**	50.5	85.7	47.5	66.6	43.2
	DA3	9.9	6.4	9.2	**49.0**	18.6	45.3	30.7	29.0	55.1	40.0	80.3	38.2	59.3	35.3
	EVO-0	14.5	9.8	14.1	48.5	21.7	51.6	39.8	**42.2**	68.3	50.5	84.8	41.1	62.9	41.5
	Spatial Forcing	12.9	9.0	14.1	48.7	21.2	49.5	36.0	33.4	64.5	45.8	83.2	49.3	66.2	40.0
	Dualhead-SF	6.3	4.4	5.5	**49.0**	16.3	37.3	28.9	34.2	45.2	36.4	57.5	31.8	44.7	30.0
	GDA	**17.9**	**13.2**	19.3	48.5	**24.7**	**53.4**	38.7	42.0	**69.1**	**50.8**	**85.8**	**52.9**	**69.3**	**44.1**

Main comparison on VLM benchmarks. Table 1 shows a consistent pattern: for DINO-B and DINO-L, GDA is the only 3D-enhanced method that improves spatial grounding while preserving broad VLM utility, yielding the best overall average. In contrast, explicit fusion (DA3 replacement, EVO-0) and implicit alignment (Spatial Forcing) often degrade VL-Avg and R-Avg, indicating that stronger geometry supervision can perturb the pretrained RGB-to-text interface. DA3 replacement performs poorly, suggesting that a geometry-centric encoder alone is not a suitable drop-in replacement for the RGB-oriented visual interface. Dualhead-SF also remains substantially below the RGB-only baseline, indicating that preserving a separate semantic branch is insufficient when geometry is still fused unconditionally. By contrast, GDA improves S-Avg and R-Avg while slightly improving VL-Avg, achieving the best overall average (43.2→44.1). Across three full GDA pretraining runs, the overall averages are 44.1/44.3/44.0 (mean 44.13, s.d. 0.15), showing that the GDA result itself is stable across the tested seeds; matched RGB-only repeats average 43.35 with s.d. 0.87, so the overall-average gap sits inside the baseline's seed spread while GDA's own run-to-run variation is roughly six times smaller. Appendix E reports further VLM-side results and analyses, and Appendix E.4 separates training-seed uncertainty from rollout-level uncertainty, and Table 14 reports matched three-seed repeats of both GDA and the RGB-only baseline.

Scaling across LLM sizes and vision backbones. We also evaluate whether the benefit of GDA persists beyond the Qwen2.5-0.5B setting used in Table 1, where GDA yields positive gains across all tested LLM sizes and both visual backbones. Overall, the gains remain consistent for DINO-L (+1.1 to +1.8) and become especially large for the strongest DINO+SigLIP + 7B setting (+4.7). These results indicate that GDA is not merely a small-model regularizer, but remains effective as model capacity increases. Detailed per-task results are provided in Appendix E.2.

4.3 VLA Benchmark Results

We next evaluate whether the VLM-side improvements of GDA translate into stronger embodied control; Appendix F gives the full simulation protocols, real-world task design, and additional results.

Simulation results on LIBERO. We report success rate (SR) on LIBERO's four suites (Table 2). For OpenVLA-style policies (pretraining+FT), we initialize from the VLMs in Section 4.2 and retain the same GDA-augmented backbone during downstream fine-tuning. In the controlled 0.5B setting, GDA improves average SR from 86.3 to 95.8 (+9.5) and exceeds Spatial Forcing by 12.2 points, with the largest gain on the Long suite (69.4→88.4). At 7B and in the π_0 regime the baselines are already high, so differences are smaller. At 7B, GDA reaches 98.1 average SR against 97.3 for the reproduced base policy and 98.0 for Spatial Forcing under the same recipe, too small a margin to separate the two geometry methods here. The published Spatial-Forcing entry (98.5) is not comparable: reference

Table 2: LIBERO benchmark comparison. Reference results from prior work are shown for context, while controlled comparisons use our reproduced base policies and matched training settings. Indented rows denote geometry-enhanced variants of the corresponding base policy. The reproduced numbers slightly differ from the published ones because the recipe, backbone, and training budget are matched to our base policy. Per-suite entries for the controlled comparisons are mean ± standard deviation over three independently trained policies; Avg. is the mean of the four per-suite means.

Method	Spatial	Object	Goal	Long	Avg.
Reference results from prior work					
Dita (Hou et al., 2025)	84.2	96.3	85.4	63.8	82.4
CoT-VLA (Zhao et al., 2025)	87.5	91.6	87.6	69.0	83.9
π_0-FAST (Pertsch et al., 2025)	96.4	96.8	88.6	60.2	85.5
UniVLA (Bu et al., 2025)	96.5	96.8	95.6	92.0	95.2
SpatialVLA (Qu et al., 2025)	88.2	89.9	78.6	55.5	78.1
DepthVLA (Yuan et al., 2025)	96.4	98.0	95.8	89.2	94.9
π_0 (Black et al., 2024)	96.8	98.8	95.8	85.2	94.2
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
Spatial Forcing (Li et al., 2025)	99.4	99.6	98.8	96.0	98.5
Controlled comparisons					
OpenVLA-OFT (Qwen2.5-0.5B, reproduced)	93.4 ± 0.1	87.8 ± 0.2	94.6 ± 0.1	69.4 ± 0.3	86.3
+ Spatial Forcing (Li et al., 2025)	87.6 ± 0.4	87.2 ± 0.2	92.8 ± 0.1	66.8 ± 0.5	83.6
+ GDA (Ours)	**97.4** ± 0.2	**99.6** ± 0.2	**97.8** ± 0.3	**88.4** ± 0.1	**95.8**
OpenVLA-OFT (Qwen2.5-7B, reproduced)	97.6 ± 0.1	98.8 ± 0.1	98.2 ± 0.1	94.4 ± 0.2	97.3
+ Spatial Forcing (Li et al., 2025)	**99.2** ± 0.1	99.6 ± 0.0	98.6 ± 0.1	94.4 ± 0.4	98.0
+ GDA (Ours)	**99.2** ± 0.0	**99.8** ± 0.1	**98.8** ± 0.1	**94.6** ± 0.2	**98.1**
π_0 (3B, replicated results)	96.8 ± 0.1	**97.8** ± 0.2	92.0 ± 0.2	**85.6** ± 0.1	93.1
+ GDA (Ours)	**97.6** ± 0.2	**97.8** ± 0.3	**96.2** ± 0.2	85.4 ± 0.1	**94.3**

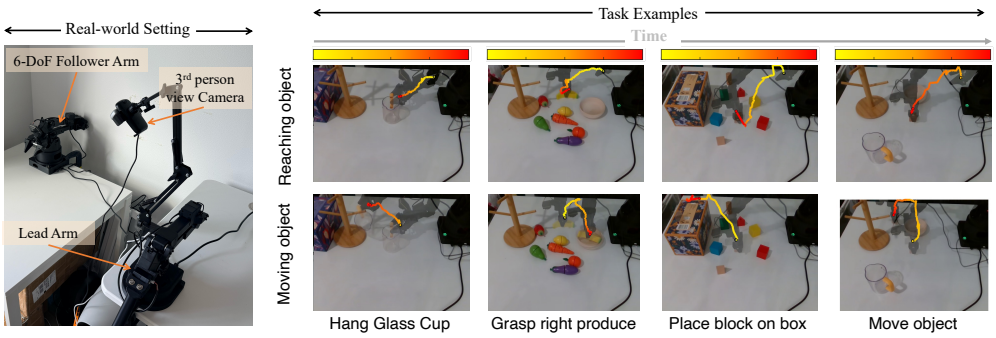

Figure 2: Real-world evaluation setup and task snapshots. We evaluate four manipulation tasks under controlled variation axes; policies observe third-person and wrist-view RGB images.

rows use each method's own backbone and budget, and only the controlled comparison rows are matched. In the π_0 FT-only regime GDA improves the base policy from 93.1 to 94.3, driven mainly by the Goal suite (92.0→96.2). GDA thus transfers across pretraining+FT and FT-only regimes, with the largest gains in the less-saturated 0.5B setting.

Simulation results on RoboTwin. We further evaluate GDA on RoboTwin 2.0, a more challenging bimanual real-to-sim benchmark. GDA improves the average SR from 51.3 to 66.7 on the easy split and from 23.0 to 42.7 on the hard split, outperforming both the base π_0 policy and Spatial Forcing. Full per-task results are provided in Appendix F.2.

Real-world manipulation results. We evaluate GDA on a compact real-world benchmark under the same π_0 FT-only protocol using a ViperX-style 6-DoF arm with a parallel gripper. Policies observe RGB from a third-person RealSense D435i camera and a wrist-mounted camera; *no depth channel is provided to any policy at test time*. Figure 2 shows the setup and task snapshots; task design and variations are in Appendix F.3. As shown in Table 3, GDA improves average SR from 49.4 (base) and 58.8 (SF) to 68.1. The largest gains occur on *Hang Glass Cups* and *Place Middle Block on Box*, which require vertical placement and contact-sensitive alignment. GDA also improves *Move Object*, suggesting benefits for spatial-language-conditioned placement, while *Grasp Right-side Produce* shows smaller gains, consistent with failures dominated by appearance and grasp selection rather than

Table 3: Real-world manipulation success rates under the same π_0 FT-only protocol. Each task is evaluated over 40 trials; task definitions and variation details are in Appendix F.3. Δ reports the absolute SR gain of GDA. The experiment rollouts can be found in the supplementary materials.

Task	Base	SF	GDA	Δ vs. Base	Δ vs. SF
Hang Glass Cups (light)	22.5	47.5	**62.5**	+40.0	+15.0
Grasp Right-side Produce (object)	37.5	37.5	**40.0**	+2.5	+2.5
Place Middle Block on Box (height)	62.5	82.5	**90.0**	+27.5	+7.5
Move Object (spatial language)	75.0	67.5	**80.0**	+5.0	+12.5
Average	49.4	58.8	**68.1**	+18.8	+9.4

geometry alone; Appendix F.4 discusses representative successes and failure cases. These four tasks were intentionally selected to probe spatial/contact-sensitive regimes, so we frame this experiment as targeted real-world evidence rather than as a general-purpose real-world benchmark. Analyzing the same 160 trials as matched conditions (each condition evaluated under all three methods with rotated execution order), the success rates are 68.1% (GDA), 58.8% (SF), and 49.4% (base). Exact pooled McNemar tests give $p \approx 0.004$ for GDA vs. SF (20 vs. 5 discordant conditions) and $p < 10^{-5}$ for GDA vs. base (36 vs. 6); per-task discordant counts are in Table 16.

4.4 Analysis and Ablations

VLM-VLA trade-off. Figure 3 jointly visualizes LIBERO success and the three grouped VLM metrics under the controlled OpenVLA-style 0.5B setting. Methods that inject geometry aggressively, such as Spatial Forcing, occupy a lower-VLM-utility region, whereas GDA moves toward the upper-right frontier by improving embodied performance while preserving generic VL and robustness-sensitive utility.

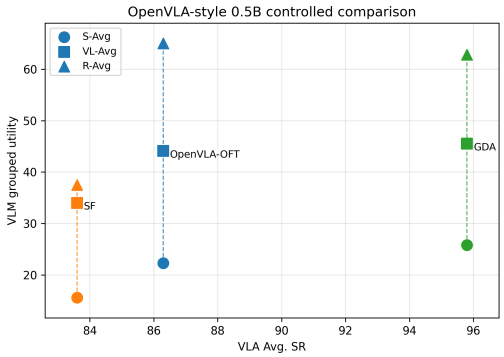

Figure 3: VLM–VLA trade-off under the controlled OpenVLA-style 0.5B setting. GDA improves LIBERO performance while preserving grouped VLM utility.

Gating ablations. Table 4 compares GDA against two gating variants formulated in Appendix C.3: Instruction-Conditioned Gating (ICG, Appendix C.3.2) and Ungated Depth Attention (UDA, Appendix C.3.1), which replaces the learned gate with a constant $\alpha = 0.09$ matching the mean gate value of GDA (Appendix E.3). First, ICG substantially degrades performance, with the largest drop on robustness-sensitive tasks (R-Avg 58.4). Thus *our pre-grounding pooled-text gate underperforms the image-conditioned gate*: pooled text embeddings are not yet grounded in the image and can trigger geometry updates based on weak lexical priors. Second, UDA tests whether a *non-selective* geometry update is sufficient by replacing the learned per-token, per-channel gate with a constant mixing coefficient. Despite matching the *mean gate value*, UDA collapses toward the unconditional-fusion regime: spatial grounding drops to S-Avg 21.2 (on par with Spatial Forcing) and the overall average falls to 41.0, well below both Pure VLM (43.2) and GDA (44.1). Thus the benefit of GDA is not "learning a good global scale" but learning a structured gate, and the interventions in Appendix E.3 localize that structure: it is the learned variation across *channels* of the residual, not a selection among tokens, that the constant coefficient destroys. A constant α discards both, whereas replacing g with its fixed per-channel profile, which retains the channel structure and discards only the input-conditioned part, costs a small fraction of the full-removal deficit. Overall, the key benefit of GDA is an *identity-preserving* gate that admits geometry along a learned subset of feature channels while leaving the pretrained representation largely intact, which also yields the highest R-Avg (69.3); that margin comes mainly from TallyQA counting (47.5→52.9) rather than from POPE (85.7→85.8).

Gate behavior and functional dependence. Despite its small mean value, the gate is functionally important. On a retrained VLM checkpoint, setting $g \to 0$ reduces VSR by 1.5 points while RefCOCO, POPE, and TallyQA change by at most 0.4 points; zeroing only the top-10% highest-gated tokens

Table 4: Gating ablation on DINO+SigLIP + Qwen2.5-0.5B. UDA removes the gate and always injects the attention-derived geometry residual. ICG conditions the gate on a pooled instruction embedding before multimodal grounding.

Method	Spatial Grounding					Generic VL Compatibility					Robustness			Avg.
	RefCOCO	RefCOCO+	RefCOCOg	VSR	S-Avg	GQA	TextVQA	VizWiz	VQA-v2	VL-Avg	POPE	TallyQA	R-Avg	
ICG	15.4	10.7	14.5	**51.0**	22.9	50.3	36.9	34.8	67.0	47.3	84.1	32.6	58.4	39.7
UDA	13.5	8.8	13.4	48.9	21.2	51.6	37.9	37.2	68.4	48.8	85.4	45.2	65.3	41.0
GDA	**17.9**	**13.2**	**19.3**	48.5	**24.7**	**53.4**	**38.7**	**42.0**	**69.1**	**50.8**	**85.8**	**52.9**	**69.3**	**44.1**

reproduces most of the VSR deficit (1.2 points), although zeroing an equal number of randomly chosen tokens costs a similar 1.1 points, so the correction is distributed across tokens rather than carried by a gate-selected subset. Disabling the gate on one trained 0.5B LIBERO policy yields 89.65 average success against the 95.8 three-policy mean, with the largest differences on Object (-9.8) and Long (-9.2). Selectivity is predominantly channel-wise: from a uniform initialization at $\sigma(-2)\approx 0.119$, the mean channel profile spans 0.009–0.253, while replacing the input-dependent gate with its dataset-mean channel profile costs 0.49 VSR points versus 1.47 points when removing the residual entirely. Together, these interventions show that the residual is important and are consistent with input-conditioned variation adding value beyond a static channel profile; full intervention curves, texture correlations, and representation-drift statistics are in Appendix E.3.

Latent geometry and teacher robustness. Feeding frozen DA3 tokens through the same GDA mechanism at both training and inference reaches 43.1 overall, below distilled GDA (44.1) and essentially equal to the RGB-only baseline (43.2), while using DA3 directly as the language-facing interface falls further to 35.3. The train-and-test teacher variant also adds 303.2M parameters per inference call and increases the vision-side forward by about $1.4\times$, indicating that more direct access to geometry is neither a stronger nor a cheaper language-facing interface. An external NYUv2 probe confirms that usable metric-depth information remains decodable from both the RGB visual features and the latent branch, while patch-centered teacher tracking (0.113 against 0.002 for an untrained head) shows the branch acquires a teacher-specific component rather than inheriting it. Moreover, replacing DA3 with VGGT while keeping the gated-fusion recipe fixed yields VLM Avg. 43.6 and LIBERO Avg. 95.65, close to the DA3-GDA results (44.1 and 95.8, respectively). These results favor adapting geometric supervision into a language-compatible latent representation rather than relying on direct access to a particular 3D teacher; details are in Appendices C.2.3 and E.5.

Capacity and content controls. GDA adds both parameters and an auxiliary objective, and the two admit separate tests. Training the identical architecture with $\lambda_{\text{distill}}=0$ retains the full module (trained end-to-end by the language-modeling loss; the teacher is never run) and scores 43.1 overall, matching the RGB-only baseline (43.2) and falling below it on spatial grounding (S-Avg 23.2 vs. 24.4), while enabling the objective at the default weight adds $+1.0$ overall and $+1.5$ S-Avg. A second control varies the content of the supervision rather than its presence: keeping $\lambda_{\text{distill}}=0.10$ but permuting the teacher's patch tokens within each image at every step removes the patch-to-patch correspondence and scores 41.5 overall and 20.8 S-Avg, falling 1.7 points below the no-geometry baseline overall. Distilling scrambled targets is thus harmful, whereas omitting the objective is neutral; the two controls together point to coherent supervision from the 3D teacher rather than to the added parameters or the auxiliary loss itself (Appendix E.1).

5 Conclusion

We studied how to improve spatial sensitivity in VLM/VLA systems without degrading the pretrained vision-language interface. Gated Depth Attention treats geometry as a gated residual refinement, preserving the RGB pathway while selectively injecting geometry-aware corrections. Across VLM benchmarks, simulation, and targeted real-world manipulation, GDA achieves a stronger balance between relational/spatial performance and generic vision-language understanding than unconditional or alignment-based geometry integration. Mechanism analyses further show that the learned residual is functionally important, that direct teacher access is unnecessary at inference, and that the trade-off transfers across two 3D teachers. Overall, the results favor conservative geometry refinement before multimodal grounding, with task-conditioned use left to the downstream multimodal model.

AI USE STATEMENT

Generative AI tools were used in two ways. First, for language editing: grammar, spelling, wording, and formatting. Second, as a coding assistant during implementation, experiment scripting, and debugging; all code was read and tested by the authors, and every number reported in this paper comes from runs the authors launched and inspected. The research questions, method design, experimental design, analysis, and all scientific claims are the authors' own. The authors reviewed all AI-assisted text and code and take full responsibility for the content of this work.

ETHICS STATEMENT

This work does not involve human subjects or personal data. The real-robot experiments are conducted in controlled laboratory environments. As with other VLA systems, failures in perception or action prediction may cause unsafe robot behavior; the method should therefore not be deployed in unstructured or human-facing environments without appropriate safety mechanisms and task-specific validation.

REPRODUCIBILITY STATEMENT

Section 3 specifies the architecture and objective, Section 4.1 defines the evaluation protocols and controlled comparisons, and Appendix C.4 reports training settings and computational overhead. Appendix C.2.3, Appendix E.3, and Appendix E.4 provide the additional mechanism and uncertainty analyses. An anonymized supplementary code package contains configurations and evaluation scripts for the main experiments.

REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. TallyQA: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 8076–8084, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017.

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

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

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.

Jiahong Chen, Jing Wang, Long Chen, Chuwei Cai, and Jinghui Lu. NanoVLA: Routing decoupled vision-language understanding for nano-sized generalist robotic policies. *arXiv preprint arXiv:2510.25122*, 2025a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, 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*, 2025b.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. SpatialRGPT: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617. IEEE, 2018.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, et al. Dita: Scaling diffusion transformer for generalist vision-language-action policy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7686–7697, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

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.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

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

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

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*, 2025.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023b.

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.

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

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

Fangyu Liu, Guy Edward Toh Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023c.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11–20, 2016.

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.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

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.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

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

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

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*, pp. 5294–5306, 2025.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision (ECCV)*, pp. 69–85, 2016.

Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. DepthVLA: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

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*, pp. 11975–11986, 2023.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1702–1713, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

APPENDIX CONTENTS

A Limitations and Broader Impact **13**

B Code and Data Availability **14**

C Additional Method Details **14**

C.1 Why Alignment and Unconditional Fusion Can Perturb VLMs 14

C.2 Additional Baselines 15

C.3 Additional Gating Variants 17

C.4 Model Details, Training Settings, and Computational Overhead 18

D Theoretical Properties of Gated Geometry Fusion **19**

D.1 From Residual Magnitude to Output Preservation 19

D.2 Propagation of Geometry-Feature Error Through Attention and Gating 20

E Additional VLM Results and Analyses **22**

E.1 Distillation-Weight Sensitivity 22

E.2 Detailed Per-Task Scaling Results 23

E.3 Gate Structure, Activation, and Intervention Analysis 23

E.4 Statistical Robustness and Baseline Tuning 24

E.5 Cross-Teacher Robustness 26

F Additional VLA Results and Experiment Details **26**

F.1 Simulation Benchmark Details 26

F.2 Additional RoboTwin Results 26

F.3 Real-World Experiment Design 27

F.4 Qualitative Examples and Failure Cases 28

A LIMITATIONS AND BROADER IMPACT

Limitations. Although GDA improves the trade-off between spatial reasoning and vision-language compatibility, our evaluation still has several limitations. First, the real-world experiments are conducted on a compact, intentionally spatial/contact-focused set of tabletop tasks with a single robot embodiment and fixed camera configuration; they should therefore be interpreted as targeted evidence rather than broad real-world validation. Second, GDA relies on geometry supervision from a frozen 3D foundation teacher during training. The DA3→VGGT swap substantially reduces concern that the result is specific to one teacher, but broader teacher families and domains remain untested. Third, our controls bound but do not fully identify the role of geometric content in the distillation target. Setting $\lambda_{\text{distill}}=0$ removes the target and per-step patch permutation makes it incoherent, so together they separate coherent supervision from its absence or from noise. Isolating *geometric* content additionally requires a teacher that is coherent, learnable, and informative but not geometric, and for single RGB images the natural candidates are 2D representation teachers whose content a strong 2D backbone already encodes; a null result under such a teacher would therefore be confounded with redundancy. Fixed-permutation and feature-rotation variants do not resolve this either, since an invertible or learnable corruption is recoverable by the branch. Fourth, the default gate does not directly observe the current instruction. The pre-grounding ICG baseline performs worse, but this does not rule out more principled task conditioning after image-text grounding.

Broader impact. This work aims to improve spatial reasoning and manipulation robustness in RGB-only VLA policies. Such improvements may benefit assistive robotics, household manipulation, and industrial automation by reducing reliance on depth sensors or explicit 3D inputs at deployment time. However, VLA policies act in the physical world, and failures in perception, grounding, or action prediction may lead to unsafe robot behavior. Our experiments are conducted in controlled settings, and the proposed method should not be deployed in unstructured or human-facing environments without additional safety mechanisms, monitoring, and task-specific validation. Moreover, GDA inherits limitations and biases from the underlying VLM/VLA backbones and the data used to train them.

B Code and Data Availability

We provide anonymized code, configuration files, and evaluation scripts as supplementary material to support reproducibility. The released materials include the implementation of GDA, training configurations for the VLM and VLA experiments, and scripts for computing the reported VLM benchmark metrics and VLA success rates. For public benchmarks, including RefCOCO-family datasets, GQA, VQA-v2, POPE, TallyQA, TextVQA, VizWiz, VSR, LIBERO, and RoboTwin, we follow the standard dataset protocols and licenses. For the real-world manipulation benchmark, demonstrations and videos are collected by the authors in a controlled lab environment. Representative evaluation videos and task descriptions are included in the supplementary material.

C Additional Method Details

This section provides additional methodological details omitted from the main text, including the motivation for gated residual fusion, diagnostic baselines, backbone-specific token construction, gating variants, and computational overhead.

C.1 Why Alignment and Unconditional Fusion Can Perturb VLMs

This section provides a more explicit discussion of why common geometry-integration strategies can perturb the pretrained vision-language interface. Let $\hat{V} \in \mathbb{R}^{N\times d}$ denote RGB visual tokens projected into the LLM embedding space, and let $Z_{\text{3D}} \in \mathbb{R}^{N\times d}$ denote geometry tokens projected into the same space.

Representation alignment. Alignment-based methods impose an auxiliary loss that pulls an internal VLM representation H_ℓ (e.g., the ℓ-th transformer layer) toward geometry-centric teacher features:

$$\mathcal{L}_{\text{align}} = \sum_i \left\| q(H_{\ell,i}) - Z_{\text{3D},\sigma(i)} \right\|_2^2, \tag{5}$$

where q is a projector and $\sigma(\cdot)$ is a token-matching rule. This objective is appealing because it transfers 3D inductive bias without changing the input format. However, Z_{3D} is typically optimized for geometry-related objectives, such as depth, layout, correspondence, or reconstruction, rather than recognition semantics. As a result, forcing H_ℓ to match Z_{3D} can reshape the semantic structure of the visual representation space and weaken its compatibility with language grounding.

Unconditional concatenation. Concatenation-based methods feed a sequence such as $[\hat{V}; Z_{\text{3D}}; T]$ to the LLM, or to an intermediate fusion module before the LLM. This exposes the model to both RGB and geometry tokens, but also forces it to solve a harder three-way RGB-geometry-language co-alignment problem. When the geometry stream is not naturally aligned with recognition-oriented semantics, it can act as a nuisance signal for tasks that do not require explicit depth reasoning. Moreover, because geometry tokens are always present, the model has no explicit mechanism for preserving the original RGB-only pathway when geometry is irrelevant or unreliable. These two failure modes motivate the design of GDA. Rather than directly aligning the RGB pathway to a geometry teacher or always injecting geometry tokens, GDA treats geometry as a gated residual refinement on top of the original RGB representation. This preserves the pretrained visual-token interface while still allowing geometry-aware corrections when supported by the learned visual context.

C.2 Additional Baselines

For consistency with the main text, we omit the batch dimension throughout this section. All equations apply per example.

C.2.1 Dual-Head Channel Concatenation Baseline

Motivation. This baseline addresses a natural concern about alignment-based methods such as Spatial Forcing: perhaps directly pulling the original visual tokens toward a geometry teacher is too aggressive. We therefore consider a milder alternative in which the backbone predicts two streams from the same visual features: one stream is intended to preserve the original image-oriented representation, while the other is supervised to match the 3D teacher. This baseline lies between the two paradigms discussed in the main text. Compared with direct representation alignment, it preserves a dedicated RGB-oriented branch rather than explicitly moving the original visual tokens toward a geometry teacher. Compared with explicit token fusion, however, it still injects geometry unconditionally by concatenating the geometry-oriented branch before the multimodal projector. Thus, the projector must jointly map semantic and geometry-aware signals into a shared LLM embedding space for all inputs and prompts.

Formulation. Given backbone features $V \in \mathbb{R}^{N\times d_v}$, the dual-head baseline predicts

$$V_{\text{rgb}} = \mathcal{H}_{\text{lang}}(V) \in \mathbb{R}^{N\times d_{\text{rgb}}}, \qquad U = \mathcal{H}_{\text{depth}}(V) \in \mathbb{R}^{N\times d_u}, \tag{6}$$

where V_{rgb} is the language-facing branch and U is the geometry-oriented branch. To preserve the original visual pathway as much as possible, we implement $\mathcal{H}_{\text{lang}}$ as a linear layer initialized to the identity map when dimensions match, so that the baseline starts from $V_{\text{rgb}} \approx V$ at initialization. The geometry-oriented branch is supervised with the same frozen 3D teacher used by GDA:

$$\mathcal{L}_{\text{distill}} = \ell(U, D^*). \tag{7}$$

The two streams are concatenated along the channel dimension and projected into the LLM space:

$$\tilde{V}_{\text{dual}} = [V_{\text{rgb}}; U] \in \mathbb{R}^{N\times(d_{\text{rgb}}+d_u)}, \qquad \hat{V}_{\text{dual}} = \mathcal{P}(\tilde{V}_{\text{dual}}) \in \mathbb{R}^{N\times d}. \tag{8}$$

The resulting $\hat{V}_{\text{dual}}$ is then fed to the LLM in the same way as standard LLaVA-style visual tokens.

Interpretation. Unlike Spatial Forcing, this baseline does not explicitly pull the original RGB tokens toward a 3D teacher space. However, unlike GDA, it still mixes semantic and geometry-aware signals unconditionally before the LLM. As a result, the multimodal projector must absorb both streams into a single visual representation regardless of whether geometry is useful for the current input. This baseline therefore tests whether preserving a separate semantic branch is sufficient when geometry is still fused without an explicit gate.

C.2.2 DA3 Replacement Baseline

Motivation. This baseline tests the opposite extreme of Spatial Forcing. If aligning RGB features toward a geometry-centric teacher perturbs the pretrained RGB-language interface, one might instead ask whether it is better to start directly from a geometry-oriented visual encoder and align it with language from scratch using VLM pretraining data. Concretely, we replace the DINO vision encoder in a LLaVA-style VLM with DA3 and train the multimodal connector and language model on the same VLM pretraining data. This baseline asks whether a geometry-centric backbone can serve as the primary visual interface for language without relying on RGB-oriented semantic features.

Formulation. Let

$$V_{\text{DA3}} = f_{\text{DA3}}(I) \in \mathbb{R}^{N\times d_{\text{DA3}}} \tag{9}$$

denote DA3 features extracted from the input image. We project them into the LLM space using a learned connector:

$$\hat{V}_{\text{DA3}} = \pi_{\text{DA3}}(V_{\text{DA3}}) \in \mathbb{R}^{N\times d}. \tag{10}$$

The resulting VLM is then trained in the standard LLaVA-style manner:

$$Y \sim p_\theta(\cdot \mid [\hat{V}_{\text{DA3}}; T]). \tag{11}$$

Interpretation. This baseline removes the tension between RGB and geometry by construction: there is no RGB semantic stream to preserve. If it performs poorly, the conclusion is not that DA3 is a weak model, but rather that geometry-centric representations alone are not an ideal drop-in replacement for the semantic visual backbone of a generic VLM. In other words, replacing an RGB-oriented encoder with a depth-oriented encoder does not by itself solve the problem of preserving broad image-text compatibility.

C.2.3 Latent Branch Validation and Direct-Teacher Control

We validate two aspects of the latent branch directly: whether the RGB visual features retain enough geometry to support distillation, and whether frozen teacher features should instead be used directly inside GDA.

Direct teacher features inside GDA. We replace the predicted geometry tokens with frozen DA3 tokens while keeping the same gated-fusion mechanism; DA3 therefore runs at both training and inference. The variant reaches 43.1 overall (S-Avg 24.1, VL-Avg 50.6, R-Avg 66.0), compared with 44.1/24.7/50.8/69.3 for the distilled latent branch. In contrast, using DA3 directly as the language-facing visual interface gives 35.3 overall, while the no-geometry RGB baseline is 43.2. Thus, greater directness of teacher access does not monotonically improve the VLM (Table 5): distilled GDA (44.1) $>$ frozen-DA3-in-GDA (43.1) $\approx$ RGB-only (43.2) $>$ DA3 replacement (35.3). These variants necessarily differ in more than directness: feeding teacher features into the gated module changes what the module consumes, and replacing the interface changes the architecture, so the comparison is between three designs with increasing teacher exposure. Against the RGB-only baseline (43.2; three-seed mean 43.35 ± 0.87), supplying the teacher's own features at train and test does not improve on having no geometry at all, and a construction that does not improve on the no-geometry baseline cannot serve as an upper bound for the distilled branch. This ordering is consistent with the limiting factor being compatibility with the pretrained token space rather than access to geometry.

Table 5: Direct-teacher control and inference cost. The frozen-DA3 variant runs the 3D teacher at test time; GDA predicts geometry internally from RGB.

Variant	Avg.	GDA add-on	Teacher run at test	Vision forward (bs=1 / 32)
RGB-only	43.2	–	–	–
GDA (latent)	44.1	8.5M	No	23.7 / 2.74 ms/img
Frozen DA3 $\rightarrow$ GDA	43.1	8.5M	Yes (+303.2M)	34.6 / 3.85 ms/img
DA3 replacement	35.3	–	Yes	–

External geometry probe and teacher tracking. We train the same two-layer MLP probe on frozen features to predict per-patch NYUv2 ground-truth depth (1,449 RGB-D images, image-level 80/20 split, 256 patches/image). The original visual features V achieve AbsRel 0.277 and δ_1 0.524; the latent features $U = \mathcal{H}_{\text{depth}}(V)$ achieve 0.282 and 0.520 (Table 6). Therefore, usable metric-depth signal is already present in V and is preserved through the latent branch. The probe is an availability check and is not expected to separate U from V, since U is a linear map of V; the teacher-tracking controls below carry the added-value evidence. Separately, on approximately 49k held-out images, U tracks the frozen teacher feature space with cosine similarity approximately 0.86. We use this cosine only as a teacher-tracking diagnostic, not as a metric-depth accuracy measure. Two controls sharpen both readings. For the probe, an untrained depth head of the same architecture reaches $\delta_1 = 0.514$; because the branch is a linear map of V and the probe head is trained on top of it, the probe measures availability of metric-depth signal rather than the branch's added value, which the interventions in Appendix E.3 address instead (a second seed reproduces this: 0.511/0.523/0.528 for trained/untrained/V). For the tracking cosine, the teacher's token space is anisotropic (mean pairwise cosine 0.962), so we decompose the raw value: after removing each image's mean token from both sides, the patch-specific agreement is 0.113, with linear CKA 0.072–0.118 (a second seed gives 0.962/0.877/0.116 over 13,222 images). That patch-specific component is genuine transfer rather than the shared direction reappearing: an untrained head attains 0.002 and the raw visual tokens -0.001, against the trained branch's 0.113, roughly $50\times$ above chance and stable across the four evaluation sets (0.115–0.119; $n = 25{,}222$ images per source).

Table 6: Geometry retained by the latent branch. The NYUv2 probe is trained identically on frozen V and U features; teacher tracking is measured on approximately 49k held-out images.

Feature source	**AbsRel** $\downarrow$	$\boldsymbol{\delta_1}$ $\uparrow$	**Cosine to teacher**
Visual features V	0.277	0.524	–
Latent geometry U	0.282	0.520	≈ 0.86

C.2.4 DINO-SigLIP Visual Token Construction

For Prismatic-style dual-encoder VLMs, the image may be encoded by both DINO and SigLIP before projection. Let

$$V_{\text{dino}} \in \mathbb{R}^{N \times d_{\text{dino}}}, \qquad V_{\text{siglip}} \in \mathbb{R}^{N \times d_{\text{siglip}}} \tag{12}$$

denote the corresponding patch features. The baseline fused representation is

$$V_{\text{fused}} = [V_{\text{dino}}; V_{\text{siglip}}] \in \mathbb{R}^{N \times (d_{\text{dino}} + d_{\text{siglip}})}. \tag{13}$$

In our depth-augmented design, the RGB pathway remains unchanged, while geometry supervision is applied only to a latent branch derived from the DINO features. The fused RGB representation is projected to the LLM embedding space as

$$\hat{V} = \pi(V_{\text{fused}}). \tag{14}$$

In parallel, the latent geometry branch is constructed only from the DINO features:

$$U = \mathcal{H}_{\text{depth}}(V_{\text{dino}}), \qquad \hat{D} = \mathcal{P}_{\text{depth}}(U). \tag{15}$$

The projected geometry tokens $\hat{D}$ are then fused with the original projected RGB tokens $\hat{V}$ through the gated residual mechanism in Section 3.3. Thus, the SigLIP pathway remains part of the original RGB representation and is not directly supervised by the geometry-distillation objective.

We make this design choice for three reasons. First, our 3D teacher features (DA3) are more naturally aligned with DINO-style patch representations than with SigLIP, so restricting teacher supervision to the DINO branch reduces teacher-student mismatch. Second, DINO patch features are better suited to dense local geometric supervision, whereas SigLIP is more strongly optimized for global image-text semantics. Third, supervising only the DINO branch localizes geometric adaptation and preserves the semantically strong SigLIP pathway, reducing the risk of introducing an additional alignment problem into the dual-encoder representation. Overall, this design minimizes alignment difficulty while preserving the complementary semantics already present in the original Prismatic visual backbone.

C.3 Additional Gating Variants

This section provides details for the gating variants discussed in Section 4.4, whose results are reported in Table 4. We consider two variants of GDA: Ungated Depth Attention (UDA) and Instruction-Conditioned Gating (ICG). Both variants use the same latent geometry branch and the same depth-attention residual as the default model, but differ in how the residual is injected into the visual representation.

C.3.1 Ungated Depth Attention

UDA removes the learned gate from GDA. Given the attention-derived geometry residual Δ, UDA directly updates the visual tokens as

$$\tilde{V}_{\text{UDA}} = \hat{V} + \alpha \cdot \Delta. \tag{16}$$

This baseline tests whether cross-attention alone is sufficient for selective geometry integration, and whether a *non-selective* geometry update is sufficient by replacing the learned per-token, per-channel gate with a constant mixing coefficient. While cross-attention can determine *what* geometry information to retrieve, UDA has no mechanism for deciding *whether* the retrieved geometry should modify the RGB representation. We set $\alpha = 0.09$ to match the mean gate value of GDA (Appendix E.3). Thus, any retrieved geometry residual is always injected into the visual tokens, even when geometry is irrelevant or potentially harmful.

C.3.2 INSTRUCTION-CONDITIONED GATING

ICG augments the default image-conditioned gate with a pooled instruction embedding. Let

$$s = \mathrm{Pool}(T) \in \mathbb{R}^d \tag{17}$$

denote a pooled text representation. For each visual token, ICG predicts the gate as

$$g_i = \sigma\Big(\mathcal{G}_{\mathrm{ICG}}\big([\mathrm{LN}(\hat{V}_i); \mathrm{LN}(\Delta_i); s]\big)\Big) \in [0,1]^d, \tag{18}$$

where $[\cdot;\cdot;\cdot]$ denotes concatenation. The fused visual tokens are then computed as

$$\tilde{V}_{\mathrm{ICG}} = \hat{V} + g \odot \Delta. \tag{19}$$

This variant tests whether explicitly conditioning the gate on the instruction improves geometry routing. In practice, the pooled instruction embedding is introduced before multimodal grounding, and thus may provide only a weak linguistic prior over whether geometry is useful for the image.

C.4 MODEL DETAILS, TRAINING SETTINGS, AND COMPUTATIONAL OVERHEAD

Backbone configurations. For LLaVA/OpenVLA-style models, unless otherwise stated, we build on the Prismatic VLM backbone with a Qwen2.5 language model (0.5B / 7B) and a ViT-based vision encoder (trained in the single-stage full-finetune recipe): DINOv2-B / DINOv2-L / SigLIP, or the dual-encoder DINOv2-L+SigLIP setting (Oquab et al., 2023; Zhai et al., 2023; Kim et al., 2025). For downstream VLA fine-tuning, we attach the standard OpenVLA-style action head (action projector + decoder; $\sim$0.3B parameters), following prior practice (Kim et al., 2025). For π_0-based models, we follow the official openpi JAX implementation to conduct full finetuning.

Training protocol (pretraining+FT). We follow the two-stage regime used throughout the paper: (i) **VLM pretraining** (multimodal instruction modeling) with GDA-enabled visual tokens, and (ii) **VLA fine-tuning** on robot data (e.g., LIBERO) using the same GDA-augmented visual backbone. For the Prismatic-style VLM pretraining stage, we use a matched recipe across all fusion methods: learning rate 2×10^{-5}, weight decay 0.1, cosine LR schedule, warmup ratio 0.03, and 1 epoch over the same training mixture (e.g., LLaVA-style instruction data and OXE/Open X-Embodiment data (Liu et al., 2023c; O'Neill et al., 2024)). For VLA fine-tuning, we follow OpenVLA-OFT-style LoRA adaptation (Kim et al., 2025; Hu et al., 2022). We report compute/latency for the DINO+SigLIP + Qwen2.5-0.5B configuration as a representative setting. Across all trained methods in the main comparisons: (i) the vision encoder(s) (DINO/SigLIP) are trained in the single-stage full-finetune recipe, (ii) the external 3D teacher is frozen and used *only* to generate training-time supervision targets, and (iii) the teacher is removed at inference time. The direct-teacher control of Appendix C.2.3 is the one deliberate exception, since it runs DA3 at inference by construction. All methods train the same language-side components (LLM + multimodal connector/projector) during VLM pretraining, and the same action head during VLA fine-tuning. Geometry-enhanced variants additionally train a lightweight latent geometry branch; GDA further trains the gated depth-attention module (cross-attention + gating network). All controlled VLM pretraining comparisons in this setting use the same update budget: 1 epoch (approximately 5.2k optimizer steps at $\sim$665k samples with global batch 128), effective global batch size 128 ($8\times$A100-40GB, per-device batch size 4, gradient accumulation 4), and the same epoch schedule. Thus, differences reflect the fusion mechanism rather than scale, batch size, or total updates.

Training protocol (FT-only for π_0). We follow the official openpi JAX implementation to finetune the VLA tasks with GDA components for 30,000 steps.

Gate initialization. To preserve RGB-only behavior at the start of training, we initialize the final layer of the gating network $\mathcal{G}$ with zero weights and a negative bias so that g starts near zero (e.g., $\sigma(-2) \approx 0.12$). This encourages geometry-aware residuals to be adopted only when supported by the training signal.

Training-time overhead. Table 7 reports end-to-end VLM pretraining wall-clock cost under the matched budget. Geometry-enhanced variants add $< 1\%$ parameters relative to the base VLM. The wall-clock overhead is dominated by training-time teacher inference and auxiliary geometry supervision, not by inference-time compute (the teacher is removed at test time).

Table 7: Training-time overhead for the VLM setting (DINO+SigLIP + Qwen2.5-0.5B) under a matched budget (global batch 128; 8×A100-40GB; 1 epoch). We report end-to-end seconds per optimizer step (sec/step).

Method	# Params (Millions)	Training time (sec/step)	Δ time vs. base
Pure VLM	1,227.7	2.47	–
SF	1,238.6	2.84	+15.0%
GDA	1,236.2	2.91	+17.8%

Evaluation-time latency on LIBERO. Table 8 reports end-to-end evaluation latency as **seconds per episode** (sec/ep), computed from 500 rollouts per suite (Goal/Object/Spatial/Long). These measurements include environment simulation and policy inference, so they should be interpreted as coarse system-level latency rather than model-only FLOPs. Because the external 3D teacher is removed at inference time, the overhead of geometry-enhanced variants is small in end-to-end evaluation.

Table 8: End-to-end LIBERO evaluation latency (sec/ep) for DINO+SigLIP + Qwen2.5-0.5B on a single A100 GPU. Values are computed from 500 rollouts per suite. **Overall** is the mean over all suites (2000 rollouts total).

Method	Goal	Object	Spatial	Long	Overall
Base	8.40	9.43	9.28	17.47	11.15
SF	8.48	8.81	8.97	17.79	11.01
GDA	9.10	8.91	8.87	18.65	11.38

D Theoretical Properties of Gated Geometry Fusion

We analyze the fusion equations in Section 3.3 at a fixed checkpoint, with dropout and other stochastic training operations disabled. These are deterministic perturbation results for the stated architecture, not assumptions about successful optimization or the relative performance of independently trained models. The RGB tokens, instruction, and any decoding prefix are held fixed unless explicitly varied. All logarithms are natural.

D.1 From Residual Magnitude to Output Preservation

Proposition 1 (Residual-controlled output preservation)**.** *Fix model parameters, an instruction, and a decoding prefix. Let z_0, z_1 be the next-token logits obtained with visual inputs $\hat{V}$ and $\hat{V} + E$, respectively, and let $p_j = \mathrm{softmax}(z_j)$ for $j \in \{0, 1\}$. Let $E = g \odot \Delta$ denote the injected residual and $r = \|E\|_F$ its Frobenius norm. Suppose the fixed-prefix logit map has a finite sensitivity bound L along this perturbation, so that $\|z_1 - z_0\|_2 \le L\|E\|_F$. With $\alpha = \max_{i,c} g_{ic}$,*

$$r \le \alpha\|\Delta\|_F, \qquad \mathrm{KL}(p_0\|p_1) \le \frac{L^2 r^2}{4} \le \frac{L^2\alpha^2\|\Delta\|_F^2}{4}. \tag{20}$$

Moreover, if the largest baseline logit exceeds the second largest by $m > 2Lr$, the greedy next-token prediction is unchanged.

Proof. Because each gate entry lies in $[0, \alpha]$,

$$\|E\|_F^2 = \sum_{i,c} g_{ic}^2\Delta_{ic}^2 \le \alpha^2\|\Delta\|_F^2.$$

For a logit vector z, write $F(z) = \log\sum_k \exp z_k$ and $p = \mathrm{softmax}(z)$. Its Hessian is $\nabla^2 F(z) = \mathrm{diag}(p) - pp^\top$. For any direction h,

$$h^\top\nabla^2 F(z)h = \mathrm{Var}_{K\sim p}(h_K) \le \frac{(\max_k h_k - \min_k h_k)^2}{4} \le \frac{\|h\|_2^2}{2}.$$

The variance inequality follows from $(X-\ell)(u-X)\geq 0$ for a variable supported on $[\ell,u]$, which gives $\mathrm{Var}(X)\leq(u-\mathbb{E}X)(\mathbb{E}X-\ell)\leq(u-\ell)^2/4$. For distinct maximum and minimum coordinates, Cauchy–Schwarz gives $(\max h-\min h)^2\leq 2\|h\|_2^2$; the constant-vector case is immediate.

Set $h=z_1-z_0$. The softmax KL is the corresponding log-sum-exp remainder:

$$\begin{aligned}\mathrm{KL}(p_0\|p_1)&=F(z_1)-F(z_0)-\nabla F(z_0)^\top h\\&=\int_0^1(1-t)h^\top\nabla^2F(z_0+th)h\,\mathrm{d}t\\&\leq\frac{1}{4}\|h\|_2^2\leq\frac{1}{4}L^2r^2.\end{aligned}$$

Combining this with $r\leq\alpha\|\Delta\|_F$ proves Equation 20.

For the prediction statement, let $k^\star$ be the unique maximizer of z_0. Since each logit changes by at most $\|h\|_\infty\leq Lr$, for every $k\neq k^\star$,

$$z_{1,k^\star}-z_{1,k}\geq z_{0,k^\star}-z_{0,k}-2Lr\geq m-2Lr>0.$$

Thus the greedy next-token choice is unchanged. □

Finite-horizon interpretation. For H autoregressive steps, let P_0,P_1 be sequence distributions under bypassed and fused inputs, respectively. If a bound L_t holds uniformly over common prefixes at step t, the KL chain rule yields

$$\mathrm{KL}(P_0\|P_1)=\sum_{t=1}^{H}\mathbb{E}_{Y_{<t}\sim P_0}\,\mathrm{KL}(p_{0,t}(\cdot\mid Y_{<t})\|p_{1,t}(\cdot\mid Y_{<t}))\leq\frac{r^2}{4}\sum_{t=1}^{H}L_t^2. \tag{21}$$

This follows by factorizing each sequence probability into its conditionals and taking the expectation of the log probability ratio. A single-prefix bound alone does not control an entire generated sequence. Likewise, greedy sequence equality requires the margin condition at every prefix on the baseline greedy path; induction then makes the two paths coincide. These statements concern fixed-image language generation, not a robot rollout in which actions change future observations.

What is and is not preserved. The relevant perturbation is $r=\|g\odot\Delta\|_F$, not the arithmetic mean of g. A few large gate entries aligned with large residual entries can dominate r even when the gate mean is small. For the stated initialization $W_g=0$, $b_g=-2$, one has $r=\sigma(-2)\|\Delta\|_F$ exactly. This attenuates the residual but does not set it to zero or bound the unscaled residual norm. No bound on L is measured in the current experiments, so the proposition is a conditional certificate, not an empirical certification of any benchmark prediction. Changes to the trained RGB projector or LLM parameters are outside this same-checkpoint comparison.

D.2 Propagation of Geometry-Feature Error Through Attention and Gating

The implemented gate depends on the correction it modulates. Consequently, changing geometry tokens perturbs both the attention residual and the gate. We quantify both effects for the single-head attention equations stated in Section 3.3.

For a matrix X with token rows, define $\|X\|_{\infty,2}=\max_i\|X_i\|_2$. In this subsection, $D,D'\in\mathbb{R}^{N\times d}$ denote two possible projected geometry-token matrices $\hat{D}$ for the same RGB tokens. Let

$$\Phi(D)=\hat{V}+g(D)\odot\Delta(D),\quad e=\|D'-D\|_{\infty,2},\quad\max\{\|D\|_{\infty,2},\|D'\|_{\infty,2}\}\leq B.$$

All spectral norms below are operator 2-norms. Define

$$q_{\max}=\max_i\|Q_i\|_2,\quad w_K=\|W_K\|_2,\quad w_V=\|W_V\|_2,\quad C_{\mathrm{att}}=w_V\left(1+\frac{Bq_{\max}w_K}{\sqrt{d}}\right). \tag{22}$$

Let the row-wise LayerNorm in the gate be L_{LN}-Lipschitz in Euclidean norm and put $L_g=\|W_g\|_2L_{\mathrm{LN}}/4$.

Proposition 2 (Geometry-error propagation with a residual-aware gate)**.** *Under the definitions above, with $\alpha = \max_{i,c}\{g_{ic}(D), g_{ic}(D')\}$, the actual gate of Equation 2 satisfies*

$$\|\Delta(D') - \Delta(D)\|_{\infty,2} \le C_{\mathrm{att}} e, \tag{23}$$

$$\|\Phi(D') - \Phi(D)\|_{\infty,2} \le (\alpha + B w_V L_g) C_{\mathrm{att}} e. \tag{24}$$

Thus the gate attenuates the direct residual-error path by α, while its own response contributes the additional term $B w_V L_g$.

Proof. First, softmax obeys $\|\operatorname{softmax}(a') - \operatorname{softmax}(a)\|_1 \le \|a' - a\|_\infty$. To see this, interpolate $a_t = a + th$ for $h = a' - a$ and write $p_t = \operatorname{softmax}(a_t)$. Then

$$\|\dot{p}_t\|_1 = \mathbb{E}_{K\sim p_t}|h_K - \mathbb{E}_{J\sim p_t} h_J| \le \frac{\max h - \min h}{2} \le \|h\|_\infty.$$

The middle inequality follows from Cauchy–Schwarz and the bounded-support variance inequality proved above. Integration gives the claim.

For query row i, changing D to D' changes each attention logit by at most $q_{\max} w_K e/\sqrt{d}$. Hence its probability row satisfies

$$\|A_i(D') - A_i(D)\|_1 \le q_{\max} w_K e/\sqrt{d}.$$

Since attention rows are probability vectors and $\|D_j W_V\|_2 \le B w_V$,

$$\begin{aligned}\|\Delta_i(D') - \Delta_i(D)\|_2 &\le \left\| \sum_j A_{ij}(D')(D'_j - D_j) W_V \right\|_2 \\ &\quad + \left\| \sum_j (A_{ij}(D') - A_{ij}(D)) D_j W_V \right\|_2 \\ &\le w_V e + B w_V \|A_i(D') - A_i(D)\|_1 \le C_{\mathrm{att}} e.\end{aligned}$$

Also $\|\Delta_i(D)\|_\infty \le \|\Delta_i(D)\|_2 \le B w_V$.

Because the RGB token is fixed, the concatenated gate input changes only through Δ_i. The sigmoid derivative is at most $1/4$, giving

$$\|g_i(D') - g_i(D)\|_2 \le L_g \|\Delta_i(D') - \Delta_i(D)\|_2.$$

Finally, expand the complete gated residual difference as

$$g_i(D') \odot (\Delta_i(D') - \Delta_i(D)) + (g_i(D') - g_i(D)) \odot \Delta_i(D).$$

Its Euclidean norm is at most $(\alpha + B w_V L_g)\|\Delta_i(D') - \Delta_i(D)\|_2$. Taking the maximum over rows proves Equation 24. □

For standard row-wise LayerNorm with variance stabilizer $\epsilon_{\mathrm{LN}} > 0$ and scale vector γ, one valid global bound is $L_{\mathrm{LN}} \le \|\gamma\|_\infty/\sqrt{\epsilon_{\mathrm{LN}}}$. Indeed, centering is an orthogonal projection; the derivative of $x \mapsto x/\sqrt{\|x\|_2^2/(2d) + \epsilon_{\mathrm{LN}}}$ has operator norm at most $1/\sqrt{\epsilon_{\mathrm{LN}}}$, and coordinate scaling contributes $\|\gamma\|_\infty$. This bound can be loose; the result does not assert that the trained map is a contraction.

Connection to latent distillation. To compare the student and teacher targets through the same module, set $D = \mathcal{P}_{\mathrm{depth}}(U)$ and $D' = \mathcal{P}_{\mathrm{depth}}(D^*)$. If this row-wise projector is L_P-Lipschitz on the relevant features, then

$$e \le L_P \|U - D^*\|_{\infty,2}. \tag{25}$$

Substitution into Equation 24 relates feature approximation error to fusion-output error. This is a comparison within the same fixed module, not a prediction of the performance gap to a separately trained direct-teacher baseline.

For the implemented cosine loss, feature direction alone is insufficient to bound the right-hand side. For nonzero student and teacher vectors u, t,

$$\|u - t\|_2^2 = (\|u\|_2 - \|t\|_2)^2 + 2\|u\|_2\|t\|_2\big(1 - \cos(u,t)\big). \tag{26}$$

Consequently, if every row has norm at most B_U, norm mismatch at most ρ, and cosine loss at most $\varepsilon_{\cos}$, then $e \le L_P\sqrt{\rho^2 + 2B_U^2\varepsilon_{\cos}}$. These are additional conditions, not properties guaranteed by the training loss. For example, $u = ct$ with $c > 0$ has zero cosine loss but error $|c-1|\|t\|_2$, which is unbounded as $c \to \infty$. Average teacher-tracking cosine also does not certify a worst-case per-token bound. The distillation link therefore identifies the role of feature norms and projection sensitivity without assuming unreported normalization or modifying the implemented method.

Scope. Proposition 2 accounts for gate dependence on the retrieved correction; omitting that dependence would incorrectly drop the $Bw_V L_g$ term. A gate smaller than one does not by itself imply that the full module is more robust than an ungated module, because the gate can be sensitive to its inputs. Neither proposition proves better optimization, calibrated geometry, universal vision-language alignment, superiority over another architecture, or higher closed-loop manipulation success. Their role is to make precise the residual-control mechanism and the conditions under which geometry approximation preserves the language-facing interface.

E Additional VLM Results and Analyses

This section provides additional VLM-side results and analyses that complement the main experiments, including sensitivity to the distillation weight, detailed scaling results, gating-variant ablations, and gate/representation diagnostics.

E.1 Distillation-Weight Sensitivity

Table 9 reports the per-dataset sweep over the distillation weight λ_{distill} for DINO+SigLIP-based GDA. A moderate value, $\lambda_{\text{distill}} = 0.1$, provides the best overall trade-off and is therefore used as the default DINO+SigLIP setting in the main experiments. Smaller weights preserve generic VL behavior and robustness slightly better but underuse geometry on spatial benchmarks, while larger weights degrade spatial grounding and overall performance. This non-monotonic trend supports our design principle: geometry should be injected conservatively rather than forced into the representation indiscriminately.

Table 9: Distillation-weight ablation for DINO+SigLIP-based GDA on Qwen2.5-0.5B. The table also contains two matched controls that are not weight variants: $\lambda_{\text{distill}} = 0$ retains the full module but never supervises it (the teacher is never run), and the shuffled-teacher row keeps $\lambda_{\text{distill}} = 0.10$ but permutes the teacher's patch tokens within each image at every optimization step, preserving the objective and the teacher's feature statistics while destroying patch correspondence. S-Avg averages RefCOCO, RefCOCO+, RefCOCOg, and VSR; VL-Avg averages GQA, TextVQA, VizWiz, and VQA-v2; R-Avg averages POPE and TallyQA.

Model	Spatial Grounding					Generic VL Compatibility					Robustness			Avg.
	RefCOCO	RefCOCO+	RefCOCOg	VSR	S-Avg	GQA	TextVQA	VizWiz	VQA-v2	VL-Avg	POPE	TallyQA	R-Avg	
Pure VLM	17.0	12.4	**19.5**	48.5	24.4	53.1	**40.1**	39.6	**69.1**	50.5	85.7	47.5	66.6	43.2
GDA, $\lambda_{\text{distill}} = 0$	15.6	11.2	17.3	48.5	23.2	53.0	39.8	41.0	**69.1**	50.7	85.5	49.8	67.7	43.1
GDA, $\lambda_{\text{distill}} = 0.10$, shuffled teacher	13.6	9.2	11.8	48.5	20.8	51.1	38.5	41.6	68.5	49.9	85.1	47.5	66.3	41.5
GDA, $\lambda_{\text{distill}} = 0.02$	16.1	11.6	17.4	48.5	23.4	52.9	39.1	45.0	**69.1**	**51.5**	85.5	**53.8**	**69.7**	43.9
GDA, $\lambda_{\text{distill}} = 0.10$	**17.9**	**13.2**	19.3	48.5	**24.7**	**53.4**	38.7	42.0	**69.1**	50.8	85.8	52.9	69.3	**44.1**
GDA, $\lambda_{\text{distill}} = 0.50$	14.1	8.8	12.9	**48.6**	21.1	51.1	38.7	38.6	68.4	49.2	85.3	51.1	68.2	41.7
GDA, $\lambda_{\text{distill}} = 1.00$	12.9	8.1	12.3	48.5	20.5	51.1	38.3	**46.3**	68.5	51.1	**85.9**	51.5	68.7	42.3

The $\lambda_{\text{distill}} = 0$ row separates capacity from geometry. GDA adds both parameters and an auxiliary objective, leaving open whether the gain reflects the added fusion capacity rather than geometric supervision. The $\lambda_{\text{distill}} = 0$ row is the matched control for that question: the full module is present and trained end-to-end by the language-modeling loss, the parameter count and architecture are identical to the default setting, and only the distillation objective is removed. At $\lambda_{\text{distill}} = 0$ the teacher is never run, so no geometric signal enters training. It reaches overall average 43.1, essentially the RGB-only baseline (43.2; three-seed mean 43.35 ± 0.87), and below that baseline on spatial grounding (S-Avg 23.2 versus 24.4). Enabling the objective at the default weight recovers $+1.0$ overall and $+1.5$ S-Avg. The added capacity therefore buys nothing on its own: the improvement is consistent with dependence on the distillation target rather than on module size.

The shuffled-teacher row varies the coherence of the target. Corrupting the objective tests a different question from removing it: whether the distilled signal must be geometrically meaningful.

Holding $\lambda_{\text{distill}} = 0.10$ and permuting the teacher's patch tokens within each image at every optimization step preserves the loss form, its weight, and the teacher's marginal feature statistics, and removes the patch-to-patch correspondence. Performance falls to 41.5 overall and 20.8 S-Avg, i.e. 2.6 and 3.9 points below the default setting and 1.7 and 3.6 points below the RGB-only baseline. Corrupted geometric supervision is therefore worse than no supervision at all (43.1). An auxiliary objective that merely regularized the representation would not become harmful when its targets were scrambled. Taken together, the three rows point to coherent supervision from the 3D teacher rather than to module size or to the presence of an auxiliary loss.

E.2 Detailed Per-Task Scaling Results

Table 10 reports the full per-task results for the scaling experiments. The gains of GDA remain positive across LLM sizes and visual backbones, indicating that the method is not tied to a specific model scale or encoder choice.

Table 10: Detailed scaling results across LLM sizes and vision backbones. S-Avg summarizes spatial grounding; VL-Avg summarizes generic vision-language compatibility; R-Avg summarizes robustness-sensitive evaluation.

Backbone	LLM	Spatial Grounding					Generic VL Compatibility					Robustness			Avg.	Δ
		RefCOCO	RefCOCO+	RefCOCOg	VSR	S-Avg	GQA	TextVQA	VizWiz	VQA-v2	VL-Avg	POPE	TallyQA	R-Avg		
DINO-L	0.5B	14.6	10.7	15.5	48.5	22.3	50.7	30.0	31.3	64.2	44.1	85.2	44.8	65.0	39.6	–
	0.5B+GDA	18.4	14.9	18.1	51.6	25.8	48.6	30.2	40.0	63.2	45.5	84.0	41.6	62.8	41.1	+1.5
	1.5B	34.7	28.4	31.2	52.6	36.7	56.3	38.9	52.8	70.0	54.5	87.0	61.7	74.4	51.4	–
	1.5B+GDA	36.2	29.9	32.8	53.8	38.2	56.8	39.9	54.3	70.0	55.3	87.6	63.4	75.5	52.5	+1.1
	7B	58.0	53.4	52.1	55.8	54.8	60.0	44.8	54.2	73.8	58.2	87.9	67.6	77.8	60.8	–
	7B+GDA	60.0	55.2	55.0	63.4	58.4	59.9	45.5	55.5	73.5	58.6	88.3	69.2	78.8	62.6	+1.8
DINO +SigLIP	0.5B	17.0	12.4	19.5	48.5	24.4	53.1	40.1	39.6	69.1	50.5	85.7	47.5	66.6	43.2	–
	0.5B+GDA	17.9	13.2	19.3	48.5	24.7	53.4	38.7	42.0	69.1	50.8	85.8	52.9	69.3	44.1	+0.9
	1.5B	35.3	29.4	32.4	56.6	38.4	57.1	49.4	52.1	73.0	57.9	86.5	65.2	75.9	53.7	–
	1.5B+GDA	37.9	32.3	34.3	57.9	40.6	57.2	48.5	55.2	73.1	58.5	86.3	63.6	75.0	54.6	+0.9
	7B	56.5	51.9	50.3	55.0	53.4	55.6	49.2	50.3	71.3	56.6	85.6	62.5	74.1	58.8	–
	7B+GDA	58.2	53.5	54.4	62.9	57.3	60.0	54.3	57.0	76.1	61.9	88.8	69.6	79.2	63.5	+4.7

E.3 Gate Structure, Activation, and Intervention Analysis

A global scalar gate mean can hide the structure of the implemented per-token, per-channel gate (Eq. 2); we therefore analyze its channel profile, input dependence, and interventional effect in addition to representation drift.

Channel structure. The gate is initialized exactly uniformly at $\sigma(-2) \approx 0.119$ in every channel. After training, the mean per-channel gate profile spans 0.009–0.253 (a ≈27× range; 25–27× across individual evaluation sets) with split-half reliability $r = 1.00$. Across RefCOCO, VSR, POPE, TallyQA, and 2,000 RoboTwin manipulation frames, pairwise correlations of the 896-dimensional gate profile are 0.998–1.000; the corresponding injected-correction profiles $|g \odot \Delta|$ correlate at 0.999–1.000. The manipulation-frame comparison uses the VLM checkpoint applied to manipulation-domain frames, so it establishes cross-domain consistency of the pretrained injection profile, not a policy-level test after VLA fine-tuning; the policy intervention below supplies that policy-level evidence.

Scene and texture dependence. Across approximately 306k evaluation images, the per-image mean gate is nearly constant (p1–p99 span approximately 0.002). Per-token gate values are also essentially uncorrelated with Laplacian energy or local contrast ($|r| < 0.07$, permutation-tested, with unstable signs across retrained checkpoints). Thus, the gate does not primarily route by local texture or by large scene-level amplitude changes. Its selectivity is predominantly channel-wise.

The intervention is selective (Table 11): removing the residual costs 1.47 points on VSR while changing the other tested axes by at most 0.4 points. Zeroing the top-10% most-gated tokens costs 1.23 points, but zeroing an equal number of randomly chosen tokens costs 1.06 points; the random mask recovers 86% of the same deficit, so the correction is distributed across tokens rather than carried by a gate-selected subset. Both shrinking and amplifying the learned gate also hurt, suggesting that the learned amplitude is calibrated rather than arbitrary. All rows are single evaluation passes of

the same checkpoint via a forward hook that rewrites the pre-sigmoid gate logits, so only within-table deltas are meaningful; the random mask is a single draw.

Table 11: VLM-side gate interventions on a retrained 0.5B checkpoint. Entries are changes in accuracy (percentage points) relative to the same no-intervention checkpoint.

Intervention	RefCOCO	VSR	POPE	TallyQA
$g \to 0$ (residual removed)	-0.35	-1.47	-0.40	$+0.04$
$g \to 0$ on top-10% gated tokens	-0.05	-1.23	$+0.01$	$+0.01$
$g \to 0$ on random 10% of tokens	-0.16	-1.06	-0.15	$+0.07$
$g \to 0.5g$	$+0.02$	-1.06	-0.11	$+0.10$
$g \to 2g$	-0.48	-0.33	-0.08	-0.14
$g \to$ fixed dataset-mean profile	-0.03	-0.49	-0.15	$+0.02$

Static-profile control. Replacing g by its fixed dataset-mean 896-dimensional channel vector reduces VSR by 0.49 points, whereas removing the residual entirely reduces it by 1.47 points, so the static profile preserves 0.98 of those 1.47 points, about two-thirds of the full-removal effect. The static learned channel profile therefore accounts for a substantial portion of the effect, while input-conditioned variation provides additional benefit. This is why we describe GDA as predominantly channel-selective rather than claiming strong image-adaptive routing.

Table 12: Policy-level gate intervention on a trained 0.5B LIBERO policy. Each suite uses the same 500-rollout protocol as the main table. The GDA column reproduces the three-policy per-suite means of Table 2 rather than rollout counts from one policy; the $g \to 0$ column is the intervened checkpoint itself ($470/449/478/396$ successes per suite, $1793/2000$ pooled, 89.65%; 50 trials on each of 10 tasks per suite). Δ therefore measures that checkpoint against the multi-policy mean and includes any offset between the two.

Suite	GDA	$g \to 0$	Δ
Spatial	97.4	94.0	-3.4
Object	99.6	89.8	-9.8
Goal	97.8	95.6	-2.2
Long	88.4	79.2	-9.2
Average	95.8	89.65	-6.15

Switching off injection after training yields 89.65 against the 95.8 three-policy mean (Table 12), with the largest differences on Object and Long. Relative to the separately trained no-geometry policy (86.3), the sequence 86.3 $\to$ 89.65 $\to$ 95.8 is consistent with benefits from both geometry-aware training and inference-time injection. We do not treat the difference as an exact causal partition because the 86.3 base is a different trained architecture and evaluation-time zeroing is out-of-distribution for the action head.

Perturbation magnitude of the injection. The family-level scalar summaries from the default model remain useful as a measure of perturbation magnitude (Table 13): mean gate activation is approximately 0.09 and cosine preservation with the pre-fusion RGB token stays above 0.9994. These drift statistics are computed with the projected geometry tokens in place of the attention output (i.e., $\hat{V} + g \odot \hat{D}$), an approximation of the deployed fusion; the interventions above act on the deployed computation directly.

E.4 Statistical Robustness and Baseline Tuning

VLM training-seed stability. Across three full GDA pretraining runs, the overall averages are 44.1/44.3/44.0 (mean 44.13, s.d. 0.15); grouped S-Avg is 24.7/25.0/24.5, VL-Avg 50.8/51.1/50.8, and R-Avg 69.3/69.2/69.4. These runs establish stability of GDA across the tested seeds. Repeating RGB-only pretraining with the same three seeds and identical recipe yields overall averages 42.37/43.65/44.03 (mean 43.35, s.d. 0.87), roughly six times GDA's seed spread (Table 14).

Table 13: Family-level aggregate gate amplitude and representation drift. Gate Mean averages over tokens and channels.

Task Family	Gate Mean	L2 Drift	Cosine Preservation
Spatial	0.0931	1.0167	0.999454
VL	0.0928	0.9847	0.999461
Robustness	0.0929	0.9867	0.999464

Table 14: Training-seed stability for the default DINO+SigLIP + Qwen2.5-0.5B VLM. Both methods are repeated with seeds 7/8/9 under the identical recipe; the RGB-only Table-1 value (43.2) is the original single run, and its seed-7 repeat (42.4) lies within the observed run-to-run spread.

Method	Seed 7	Seed 8	Seed 9	Mean	s.d.
RGB-only	42.4	43.6	44.0	43.35	0.87
GDA	44.1	44.3	44.0	44.13	0.15

Matched-seed analysis. The $+0.8$ mean difference between GDA (44.13 ± 0.15) and the repeated RGB-only baseline (43.35 ± 0.87) is not detectable at this sample size (paired $t = 1.53$, $\mathrm{df} = 2$; exact paired sign-flip test, $p = 0.50$; unpaired permutation over all 20 label splits, $p = 0.20$), and on one seed the baseline slightly exceeds GDA (44.03 vs. 44.00). The baseline's instability concentrates in two benchmarks, VSR (s.d. 4.6) and TallyQA (s.d. 4.0), while the RefCOCO family, GQA, VQA-v2, and POPE vary by at most a point; GDA is stable on the spatial axis where the RGB-only model is not (S-Avg s.d. 0.25 vs. 1.32). VL-Avg is the group with the most consistent margin: GDA leads on every seed ($+1.33/+2.37/+1.18$; paired $t = 4.35$, $\mathrm{df} = 2$, $p \approx 0.05$; sign-flip $p = 0.25$), which we report as an observation rather than a significance claim. The comparison therefore supports two statements: no overall-average difference is detectable between GDA and the RGB-only baseline, and GDA's run-to-run spread is markedly smaller (0.15 vs. 0.87). The single-seed Table-1 comparisons should be read with this variability in mind.

RoboTwin evaluation repeats. Holding the trained checkpoints fixed and rerunning evaluation three times gives Easy SR (mean±s.d.) 51.27±0.25 / 55.57±0.35 / 66.83±0.23 for base/SF/GDA, and Hard SR 22.80±0.26 / 31.63±0.85 / 42.7±0.85. These quantify evaluation-run variability, not policy-training seed variability.

Spatial-Forcing tuning parity. Under the matched VLM recipe, sweeping the Spatial-Forcing distillation coefficient over $\{0.02, 0.1, 0.5, 1.0\}$ gives overall averages 40.4/40.1/39.2/36.5 (Table 15). The best point in this 50× range remains below GDA (44.1), reducing the likelihood that the reported gap is explained by the swept weight.

Table 15: Spatial-Forcing distillation-weight sweep under the matched VLM recipe.

λ_{SF}	0.02	0.1	0.5	1.0
Overall Avg.	40.4	40.1	39.2	36.5

Paired real-world trials. The 40 conditions per task (160 total) of Table 3 were evaluated under all three methods under the same settings, so the same trials support a matched-pair analysis. The paired success rates are 68.1% (GDA), 58.8% (SF), and 49.4% (base). The pooled discordant counts are 20:5 for GDA vs. SF ($p \approx 0.004$, exact McNemar) and 36:6 for GDA vs. base ($p < 10^{-5}$); Table 16 reports the per-task breakdown.

Independent VLA training seeds. Each controlled LIBERO method in Table 2 is trained three times independently; the table reports per-suite mean ± s.d. over the three policies (500 rollouts per suite per run). This complements the repeated-evaluation analyses, which characterize evaluation noise conditional on a fixed checkpoint rather than policy-training variance.

Table 16: Paired real-world discordant conditions. Each entry is GDA-win:GDA-loss among 40 matched conditions per task; the pooled row is used for the exact McNemar test.

Task	GDA vs. SF	GDA vs. Base
Hang Glass Cups	7:1	18:2
Grasp Produce	1:0	2:1
Place Block	5:2	11:0
Move Object	7:2	5:3
Pooled	20:5 ($p \approx 0.004$)	36:6 ($p < 10^{-5}$)

E.5 Cross-Teacher Robustness

We test whether the gains depend on a fortunate match between DA3's training distribution and tabletop manipulation. Keeping the gated-fusion mechanism and training recipe fixed, we replace DA3 with VGGT; Table 17 summarizes both teachers against the RGB-only baseline. On the VLM evaluation, VGGT-GDA reaches overall Avg. 43.6 (S-Avg 24.7; TallyQA 50.9), within the run-to-run range of the RGB-only baseline (43.35 ± 0.87 over three seeds) and close to the DA3-GDA result, so on the VLM side the teacher swap preserves the trade-off without improving on it. On LIBERO, the VGGT-teacher policy obtains 97.6/98.0/97.6/89.4 on Spatial/Object/Goal/Long, for 95.65 average ($488/490/488/447$ successes; $1913/2000$ pooled under the same 500-rollout protocol) versus 95.8 for DA3-GDA. The benefit therefore survives the teacher swap essentially unchanged on the manipulation axis; the gains are therefore not a byproduct of one teacher's domain alignment, though two teachers establish transfer rather than teacher-independence in general.

Table 17: Cross-teacher robustness. VLM and LIBERO results are available for both 3D teachers; RoboTwin-hard is reported for the DA3 teacher.

Geometry teacher	VLM Avg.	LIBERO Avg.	RoboTwin-Hard SR
None (RGB-only)	43.2	86.3	23.0
DA3	44.1	95.8	42.7
VGGT	43.6	95.65	–

F Additional VLA Results and Experiment Details

This section provides additional details for the VLA experiments, including simulation benchmark protocols, real-world experiment design, and qualitative examples.

F.1 Simulation Benchmark Details

We evaluate simulated manipulation performance on LIBERO and RoboTwin. LIBERO contains four suites: Spatial, Object, Goal, and Long, which test spatial layout variation, object variation, goal variation, and long-horizon instruction following. RoboTwin contains easy and hard settings, where the hard setting introduces stronger nuisance factors such as clutter, background variation, lighting variation, and varied tabletop heights. All methods within each controlled comparison use the same base policy, data, optimization recipe, and training budget.

F.2 Additional RoboTwin Results

RoboTwin is a bimanual real-to-sim benchmark with an *easy* setting and a more challenging *hard* setting involving clutter, background variation, lighting variation, and varied tabletop heights. For π_0-based settings on RoboTwin, we start from the released pretrained multimodal backbone and add GDA only during downstream VLA fine-tuning. Table 18 summarizes success rates on RoboTwin 2.0. Compared with the base π_0 policy, GDA improves the average success rate from 51.3 to 66.7 on the easy split and from 23.0 to 42.7 on the hard split. Relative to Spatial Forcing, GDA also improves the

Table 18: Success rates on RoboTwin 2.0 under easy and hard settings. GDA achieves the best average performance in both regimes and shows especially large gains over the base π_0 policy on the hard split.

Task	Easy						Hard					
	DP	ACT	RDT	π_0	SF	GDA	DP	ACT	RDT	π_0	SF	GDA
Move Playingcard	47	36	43	53	60	**63**	0	0	11	22	32	**44**
Turn Switch	36	5	35	27	37	**57**	1	2	15	23	23	**36**
Click Bell	54	58	**80**	44	47	57	0	3	9	3	8	**12**
Open Microwave	5	**86**	37	80	85	79	0	0	20	50	77	**84**
Beat Block	42	56	77	43	47	**79**	0	3	37	21	34	**53**
Lift Pot	39	88	72	84	91	**97**	0	0	9	36	42	**58**
Place Shoes	23	5	**35**	28	22	**35**	0	0	7	6	3	**12**
Average	35.1	47.7	54.1	51.3	55.6	**66.7**	0.1	1.1	15.4	23.0	31.3	**42.7**

average from 55.6 to 66.7 on easy and from 31.3 to 42.7 on hard. The gains are especially pronounced on the hard split, where the benchmark introduces stronger visual nuisance factors. This pattern supports our hypothesis that geometry-aware fusion is most useful when the policy must reason about relative object location under clutter, lighting changes, and viewpoint variation. At the task level, the largest improvements over π_0 appear on *Move Playingcard*, *Turn Switch*, *Beat Block*, and *Lift Pot*, all of which require precise spatial localization or robust contact under perturbation. At the same time, GDA does not dominate every individual task; for example, ACT remains stronger on the easy split of *Click Bell* and *Open Microwave*. This nuance reinforces the same conclusion as the VLM experiments: geometry is valuable when spatial precision is the bottleneck, but it is not a universal replacement for other task-relevant inductive biases. Three repeated evaluation runs on the fixed checkpoints give Easy 51.27$\pm$0.25 / 55.57$\pm$0.35 / 66.83$\pm$0.23 and Hard 22.80$\pm$0.26 / 31.63$\pm$0.85 / 42.7$\pm$0.85 for base/SF/GDA, respectively; the GDA–SF gaps are therefore much larger than evaluation-run variability.

F.3 Real-World Experiment Design

We use a ViperX-style robotic arm with a 6-DoF manipulator and a parallel gripper. The observation setup includes a RealSense D435i camera for the third-person view and a wrist-mounted webcam for the first-person view. All policies receive RGB observations only at inference time; no explicit depth input or external 3D teacher is used during deployment.

We evaluate on four real-world manipulation tasks intentionally designed to probe complementary spatial/contact-sensitive dimensions under the same π_0-based FT-only protocol. Because the task set is targeted rather than representative of arbitrary real-world manipulation, we use it as mechanism-focused evidence and avoid claiming general real-world coverage. Each method is trained with 40 demonstrations per task variation and evaluated using the same trial protocol. Success is measured as binary task completion under the task-specific criteria below.

Hang Glass Cups. The robot must grasp a glass cup and hang it on a hook by its handle. This task stresses precise terminal alignment and contact-sensitive execution. The primary variation is **lighting condition** (bright vs. dark). Success requires the cup to remain stably hanging after release. Substeps: (i) approach cup, (ii) grasp handle, (iii) lift without dropping, (iv) align the handle with the hook, (v) insert onto hook, (vi) release stably.

Grasp Right-side Produce. The robot must grasp the produce item located on the right side. This task probes object variation and spatial discrimination. The primary variation is **target object identity** (banana vs. carrot). Success requires grasping and lifting the correct target object without picking the distractor. Substeps: (i) identify the correct right-side target, (ii) approach, (iii) establish grasp, (iv) lift without dropping.

Place Middle Block on Box. The robot must place the middle block onto a target box. This task stresses height reasoning and precise placement. The primary variation is **box height** (high vs. low). Success requires the block to be stably placed on the box top after release. Substeps: (i) identify the middle block, (ii) grasp, (iii) lift, (iv) align above the box, (v) lower to the box top, (vi) release stably.

Move Object. The robot receives a language instruction specifying where to move the target object. This task evaluates spatial-language-conditioned placement. Success requires moving the correct object to the instructed spatial configuration. Substeps: (i) select the correct object, (ii) grasp, (iii) move toward the instructed target region, (iv) release at the desired location, (v) avoid disturbing distractors.

F.4 Qualitative Examples and Failure Cases

We include qualitative examples to illustrate when geometry-aware fusion helps and when it provides limited benefit.

Representative successes. GDA is most helpful in tasks where success depends on relative depth, vertical placement, or language-conditioned spatial relations. Examples include placing an object on the correct-height support, moving a block behind a bowl, and moving a block between two cups. In these cases, the geometry-aware residual can refine the visual representation in a way that improves spatial localization and contact precision.

Representative failures. The gains are smaller on tasks where object identity, grasp affordance, or appearance-specific cues dominate the difficulty. For example, in object-variation grasping tasks, GDA may match stronger baselines without substantially improving over them. This is consistent with our main claim: geometry is useful when spatial precision is the bottleneck, but it is not a universal replacement for other task-relevant inductive biases.

Supplementary video. We provide supplementary real-world demonstrations showing side-by-side comparisons across the base policy, Spatial Forcing, and GDA. The videos highlight both successful spatial-language manipulation and representative failure cases under challenging real-world variations.