DK-VLM: GEOMETRY AS POSITION FOR SPATIAL REASONING IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models still struggle to understand 3D spatial relations across views and update spatial judgments as the observer moves. A common approach fuses 3D reconstruction features into visual tokens, but aligning these features with pretrained semantic representations remains challenging under limited spatial supervision. We propose DK-VLM with *Dual-Kernel Attention*, which computes attention from both a visual token's input position and its geometric position in the scene. A semantic kernel retains the pretrained model's native attention and positional encoding, while a separate geometric kernel uses estimated 3D points and camera poses to directly modulate attention through point displacements and relative view transformations. Then, a learned gate fuses their outputs. Geometry thus enters attention as position rather than as fused reconstruction features. We further introduce Reference-Frame Composition (RC) supervision to train the model to update its reference frame across successive judgments. It provides stepwise turn and heading targets derived from verified single-step annotations. Among the compared methods, DK-VLM-8B achieves the highest average score across six spatial benchmarks and the highest overall scores on VSI-Bench (74.33) and ReVSI-32 (58.28). RC supervision further improves multi-step spatial reasoning, while the full model preserves general video understanding.

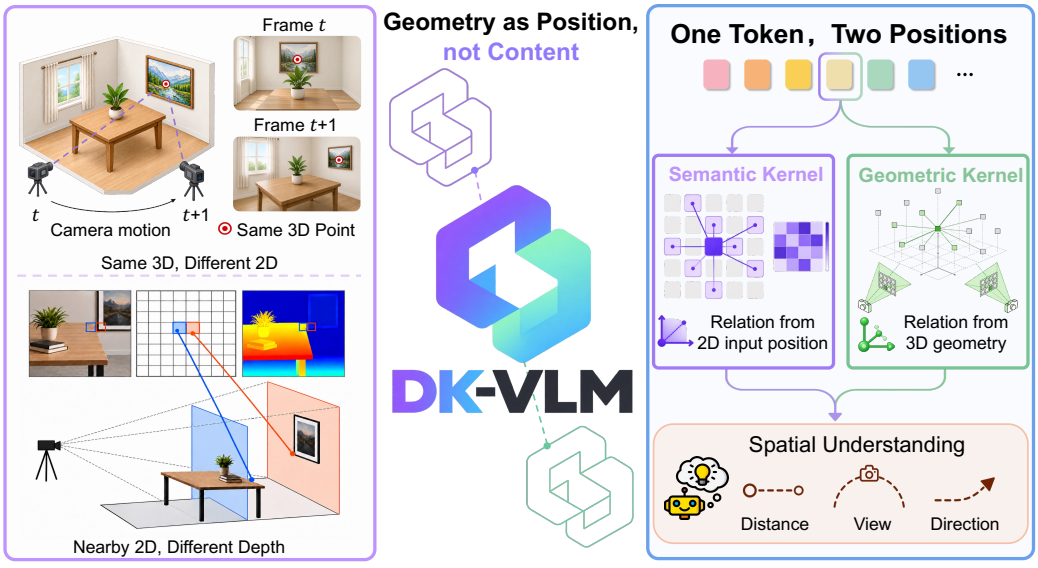

Figure 1: **Geometry as position in DK-VLM.** Left: 2D input positions do not fully capture 3D relations. Right: DK-VLM retains the native positional encoding and introduces scene geometry into attention as position.

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.

1 INTRODUCTION

Spatial reasoning connects visual understanding to agentic action in the physical world (Chen et al., 2024). Tasks like multi-view question answering, long-video understanding, robot manipulation, and embodied planning require more than recognizing visual content. Specifically, a vision-language model (VLM) must also perceive 3D relations across views and, as the observer moves, keep updating the position and heading on which its subsequent spatial judgments depend (Chen et al., 2024; Yang et al., 2025a). Despite strong progress in semantic recognition and language reasoning, VLMs remain unreliable at estimating distance, direction, camera motion, and cross-view correspondence, and weaker still on tasks such as route planning that chain several such judgments (Yang et al., 2025a; Zhang et al., 2026b).

Existing work attempts to enhance spatial reasoning by introducing 3D priors. One common approach fuses features from pretrained 3D reconstruction models with visual representations, incorporating geometry into the content of visual tokens (Fan et al., 2026; Li et al., 2026b). However, these geometric features are primarily learned for 3D reconstruction, an objective different from semantic understanding in VLMs. Integrating them therefore requires the model to align two different representation spaces while learning spatial relations. With limited spatial supervision, this additional adaptation may limit the effective use of geometry and interfere with the model's existing semantic representations.

Beyond token content, VLMs already use positional encoding to guide attention. In the Qwen-VL family (Wang et al., 2024; Bai et al., 2025b;a), multi-modal rotary position embedding (mRoPE) encodes sequence order, time, and 2D image grid coordinates, so that attention between two tokens depends on their relative positions in the input. Position thus offers a second interface for geometry: instead of being fused into token content, geometry can define relations between tokens that directly shape attention (Miyato et al., 2024; Li et al., 2025c; Wu et al., 2026; Xiang et al., 2026). However, input positions describe where a token appears in the input rather than where its content lies in the 3D world. As the camera moves, the same 3D surface appears at different image locations across frames, while neighboring 2D patches can lie at very different depths (Figure 1).

Existing geometric attention mechanisms (Miyato et al., 2024; Li et al., 2025c; Wu et al., 2026; Xiang et al., 2026) introduce 3D relations directly into attention by applying geometric transformations to queries, keys, or values. However, replacing a pretrained VLM's native positional encoding with geometric encoding may disrupt the semantic attention patterns learned under the original encoding—patterns that remain useful for spatial understanding. We therefore bring scene geometry into attention through a separate kernel while retaining the native attention and positional encoding.

We introduce DK-VLM, which implements this design through *Dual-Kernel Attention*: attention is computed from both a visual token's input position and its geometric position in the scene. The semantic kernel keeps the backbone's native attention and mRoPE unchanged. The geometric kernel uses independent projections and computes relations between visual tokens conditioned on patch-level 3D points and frame-level camera poses, so that displacements between scene points and relative transformations between views directly modulate feature comparison and aggregation. A learned gate fuses the two outputs. Geometry thus enters attention as position rather than as reconstruction features fused into token content. The geometric inputs are estimated from images, requiring no ground-truth depth or camera poses at inference.

Dual-Kernel Attention gives the model direct access to point displacements and relative view transformations, but successive spatial judgments require more: earlier movements change the position and heading on which later judgments depend, even when the scene layout remains unchanged. Single-step question answering supervises each judgment within a given reference frame and does not explicitly supervise how that frame changes across steps. We therefore introduce Reference-Frame Composition (RC) supervision, which provides stepwise turn and heading targets along landmark routes and view sequences. These targets are derived by connecting verified single-step relations that share landmarks or views within the same scene (Yang et al., 2025c; Zhang et al., 2025).

We implement DK-VLM on Qwen2.5-VL-7B and Qwen3-VL-8B (Bai et al., 2025b;a). We evaluate it on seven spatial reasoning benchmarks (Yang et al., 2025a; Zhang et al., 2026b; Yang et al., 2025d; Wang et al., 2025b; Li et al., 2025a; Wang et al., 2025d; Tong et al., 2024) and two general video understanding benchmarks (Fu et al., 2025; Li et al., 2024). Among the compared methods, DK-VLM-

8B achieves the highest six-benchmark average score and the highest overall scores on VSI-Bench (74.33) and ReVSI-32 (58.28), with trainable parameters totaling only 3.86% of its backbone's parameter count. RC supervision further improves multi-step spatial reasoning, yielding consistent gains on route-planning and viewpoint-related tasks. Ablations show that DK-VLM benefits more from RC supervision than a Qwen baseline fine-tuned on the same instruction data, supporting the complementarity between geometric relation modeling and stepwise spatial supervision. The full model also preserves general video understanding on VideoMME and MVBench.

Our contributions are:

- **Dual-Kernel Attention**, which introduces scene geometry as position through a geometric kernel parallel to a pretrained VLM's native attention.
- **Reference-Frame Composition supervision**, which supervises reference-frame updates using stepwise targets derived from verified single-step relations.
- DK-VLM-8B achieves the highest overall scores among compared methods on VSI-Bench and ReVSI-32 and the highest average in the six-benchmark comparison.

2 Related Work

Learning Spatial Reasoning. Recent work improves spatial reasoning in VLMs through spatial data and reasoning-oriented training. VST and Cambrian-S scale spatial instruction tuning, SpatialLadder adopts a progressive curriculum from perception to reasoning, and SpaceR and SpatialEvo respectively introduce verifiable rewards and deterministic geometric environments (Yang et al., 2025b;c; Li et al., 2025b; Ouyang et al., 2025; Li et al., 2026a). However, compositional supervision for tracking reference frame changes across observations has received less attention.

Geometry-Augmented Vision-Language Models. One line of work introduces 3D information by encoding coordinates into visual tokens, including Video-3D LLM, LLaVA-3D, and HiSpatial (Zheng et al., 2024; Zhu et al., 2024; Liang et al., 2026). Video-3D LLM adds coordinate embeddings to visual features. Another line introduces features from 3D reconstruction models: VG-LLM adds projected VGGT features to visual tokens, VLM-3R queries reconstruction and camera tokens with cross-attention, and GeoThinker retrieves geometric features according to the current semantic state (Zheng et al., 2025; Fan et al., 2026; Li et al., 2026b). These fusion mechanisms let a VLM use geometric representations from reconstruction models, but require aligning features learned for reconstruction with pretrained semantic representations. In contrast, DK-VLM uses estimated 3D points and camera poses to directly modulate attention among visual tokens, rather than incorporating coordinate embeddings or reconstruction features into their content.

Geometric Attention and Positional Encoding. Positional mechanisms provide another interface for introducing 3D information into Transformers. Geometric Transform Attention (GTA) incorporates group actions induced by camera transformations into attention, while PRoPE, RayRoPE, and ViewRoPE use 3D points, rays, or viewpoints to parameterize positional relations between tokens (Miyato et al., 2024; Li et al., 2025c; Wu et al., 2026; Xiang et al., 2026). These studies show that geometry can directly shape relation computation. How to introduce these mechanisms into a pretrained VLM while retaining its native positional encoding remains less explored.

3 Method

Given a sequence of RGB images and a spatial query, DK-VLM uses a geometry frontend to estimate aligned patch-level 3D points and frame-level camera poses from the images (Wang et al., 2025a; 2026a;b; Lin et al., 2025; Murai et al., 2025). DK-VLM uses these geometric states rather than latent features specific to a frontend, allowing compatible geometry frontends to be replaced. It models input position and scene geometry through parallel relation kernels. The semantic kernel retains the backbone's native positional computation, while the geometric kernel is parameterized by the estimated states. Dual-Kernel Attention computes and fuses the outputs of the two kernels (Section 3.1). To supervise the composition of geometric relations across observations, Reference-Frame Composition links verified single-step annotations into training chains with stepwise targets

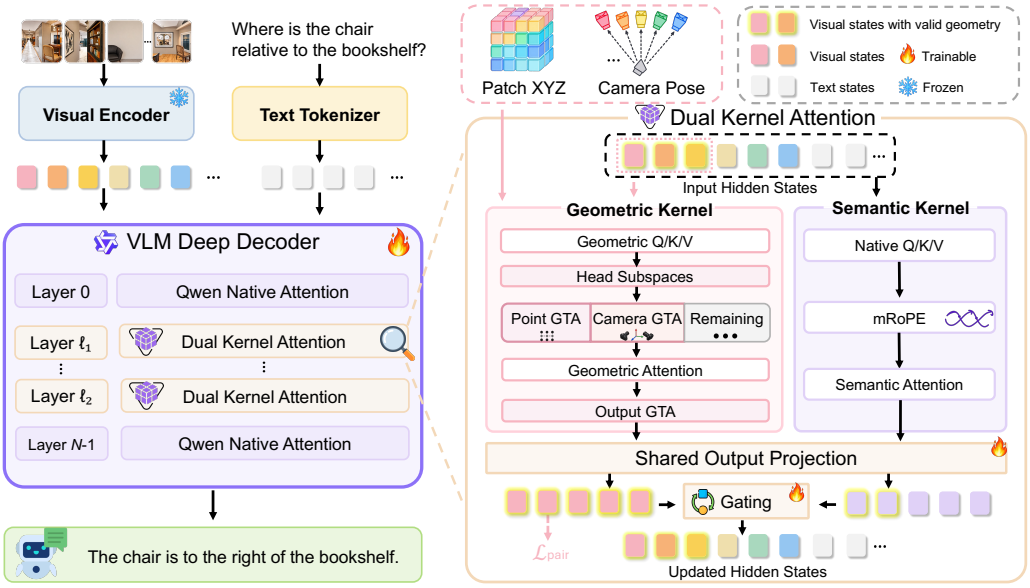

Figure 2: **Geometry as Position via Dual-Kernel Attention.** Selected decoder layers retain the backbone-native semantic kernel and introduce a geometric kernel over geometry-valid visual tokens. A learned gate fuses both outputs into the shared residual stream, while $\mathcal{L}_{\text{pair}}$ directly supervises the pre-fusion geometric representation during training.

(Section 3.2). Training jointly optimizes the instruction and pairwise geometric objectives in a single stage (Section 3.3). Figure 2 illustrates the dual-kernel architecture.

3.1 Dual-Kernel Attention

Dual-Kernel Formulation. To introduce 3D relations while retaining the backbone-native attention computation, we use two relation kernels in parallel. The semantic kernel preserves the semantic, temporal, and two-dimensional image relations learned during pretraining, while the geometric kernel conditions relations among geometry-valid visual tokens on 3D points and camera poses. At each selected Transformer layer, let $H^\ell \in \mathbb{R}^{L\times d}$ be the layer input, with sequence length L and hidden dimension d. Both kernels read H^ℓ and produce head-space outputs S^ℓ and G^ℓ. At other layers, attention follows the backbone's native computation.

Each visual token i with valid geometry carries a 3D point $\mathbf{x}_i \in \mathbb{R}^3$ in a common scene coordinate system. It is also associated with the camera-to-world pose $C_i^{\text{rel}} \in SE(3)$ of its source frame, expressed relative to the first frame. We denote its rotation and translation by R_i^{rel} and $\mathbf{t}_i^{\text{rel}}$; same-frame tokens share these quantities. Let $\mathcal{J}$ index tokens with both states valid.

The semantic kernel retains the backbone-native Q/K/V projections, mRoPE, and causal and padding masks (Wang et al., 2024; Bai et al., 2025b;a); S^ℓ is its output before the native output projection. Through independent projections, the geometric kernel derives $Q_g = H^\ell W^\ell_{Q,g}$, $K_g = H^\ell W^\ell_{K,g}$, and $V_g = H^\ell W^\ell_{V,g}$ from the same hidden states and conditions their interactions on aligned geometry. These Q/K/V projections are initialized from the corresponding native projections and have separate trainable parameters. Because $\mathcal{J}$ contains only observed visual tokens, a causal mask would make geometric relation computation depend on input order: earlier tokens could not use geometric evidence from observations appearing later in the sequence. We therefore apply full attention over $\mathcal{J}$, allowing each geometry-valid visual token to interact with all input frames; text and geometry-invalid tokens are excluded.

Geometric Kernel. We use two forms of GTA (Miyato et al., 2024) to introduce multi-view geometry into attention: Point GTA for relations between scene points and Camera GTA for camera motion across views. We split each projected geometric Q/K/V head into point, camera, and

remaining channels, with the two transforms acting on their respective subspaces. For queries, this gives $\mathbf{q}_i = [\mathbf{q}_i^{\mathrm{pt}} \| \mathbf{q}_i^{\mathrm{cam}} \| \mathbf{q}_i^{\mathrm{rem}}]$, with the same split for keys and values. The remaining channels keep their independently projected features and participate in the same geometric attention.

GTA incorporates relative geometric transforms into both attention weights and feature aggregation. Both forms use the same GTA operation: we divide the feature channels of the point and camera subspaces into blocks of four and apply the corresponding 4×4 transform to each block. Let $\mathbf{q}_{i,b}, \mathbf{k}_{i,b}, \mathbf{v}_{i,b}, \mathbf{o}_{i,b} \in \mathbb{R}^4$ denote the query, key, value, and attention output features in block b of token i. Given a transform $T_{i,b} \in SE(3)$ and its inverse $P_{i,b} = T_{i,b}^{-1}$, GTA transforms Q/K/V before attention and the output afterward:

$$\widehat{\mathbf{q}}_{i,b} = P_{i,b}^{\top} \mathbf{q}_{i,b}, \qquad \big[\widehat{\mathbf{k}}_{i,b}\ \widehat{\mathbf{v}}_{i,b}\big] = T_{i,b}\big[\mathbf{k}_{i,b}\ \mathbf{v}_{i,b}\big], \qquad \widehat{\mathbf{o}}_{i,b} = P_{i,b}\mathbf{o}_{i,b}. \tag{1}$$

The transformed query-key inner product is $\widehat{\mathbf{q}}_{i,b}^{\top}\widehat{\mathbf{k}}_{j,b} = \mathbf{q}_{i,b}^{\top} T_{i,b}^{-1} T_{j,b} \mathbf{k}_{j,b}$. Thus, the geometric transforms applied to the two tokens combine into a relative transform in the attention score.

Camera GTA uses the camera pose of the source frame, with $T_{i,b}^{\mathrm{cam}} = C_i^{\mathrm{rel}}$ for all camera feature blocks. Tokens from the same frame share this transform, whose relative action $(C_i^{\mathrm{rel}})^{-1} C_j^{\mathrm{rel}}$ describes camera motion between views.

However, spatial understanding also requires 3D relations between scene points. Camera poses alone do not specify the scene position of each patch. We therefore construct Point GTA using the patch's scene point as the translation anchor, while keeping the source frame orientation. The transform for feature block b is

$$T_{i,b}^{\mathrm{pt}} = \begin{bmatrix} R_i^{\mathrm{rel}} & \lambda_b \mathbf{x}_i \\ \mathbf{0}^{\top} & 1 \end{bmatrix}, \qquad P_{i,b}^{\mathrm{pt}} = \big(T_{i,b}^{\mathrm{pt}}\big)^{-1}. \tag{2}$$

Here, λ_b scales the displacement in feature block b. In the relative transform $\big(T_{i,b}^{\mathrm{pt}}\big)^{-1} T_{j,b}^{\mathrm{pt}}$, translation depends on the difference between point positions, while rotation captures the relative orientation of the source views:

$$R_{ij}^{\mathrm{view}} = {R_i^{\mathrm{rel}}}^{\top} R_j^{\mathrm{rel}}, \qquad \mathbf{t}_{ij,b}^{\mathrm{pt}} = \lambda_b {R_i^{\mathrm{rel}}}^{\top} (\mathbf{x}_j - \mathbf{x}_i). \tag{3}$$

The full matrix derivation is given in Appendix A.2.

The feature blocks transformed by Point GTA and Camera GTA, together with the remaining channels, form $\widetilde{Q}_g$, $\widetilde{K}_g$, and $\widetilde{V}_g$. We gather the visual tokens with valid geometry using the index set $\mathcal{J}$ and compute geometric attention:

$$O_{g,\mathcal{J}}^{\ell} = \mathrm{Softmax}\left(\frac{\widetilde{Q}_{g,\mathcal{J}} \widetilde{K}_{g,\mathcal{J}}^{\top}}{\sqrt{d_h}}\right) \widetilde{V}_{g,\mathcal{J}}, \tag{4}$$

where d_h is the head dimension. Applying the inverse GTA transform to $O_{g,\mathcal{J}}^{\ell}$ introduces relative token transforms into value aggregation as well, yielding $G_{\mathcal{J}}^{\ell}$. We then write these outputs back to their corresponding sequence positions to fuse the two kernels.

Kernel Fusion. Both kernels share an output projection ϕ_g^{ℓ}, initialized from the native output projection and trained independently of it. A separate scalar gate at each selected layer learns the mixing ratio between the two kernels. For positions $i \in \mathcal{J}$, we fuse their outputs as:

$$Y_i^{\ell} = (1 - \alpha_{\ell})\, \phi_g^{\ell}(S_i^{\ell}) + \alpha_{\ell}\, \phi_g^{\ell}(G_i^{\ell}), \qquad \alpha_{\ell} = \sigma(g_{\ell}). \tag{5}$$

We initialize $g_{\ell} = 0$ ($\alpha_{\ell} = 0.5$); training then learns an independent mixture for each selected layer. At positions outside $\mathcal{J}$, the semantic output passes through its native output projection. The fused outputs are written back to their original positions and passed to the residual stream.

3.2 Reference-Frame Composition

Dual-Kernel Attention conditions token relations on 3D position and relative pose, but multi-step reference-frame reasoning additionally requires composing relative transformations across observations. Standard single-step supervision guides spatial judgments within a given reference frame, but does not explicitly supervise how that frame changes across successive steps. Reference-Frame Composition (RC) instead derives and supervises the reference-frame update at every step. Figure 3 illustrates one such chain and its stepwise targets.

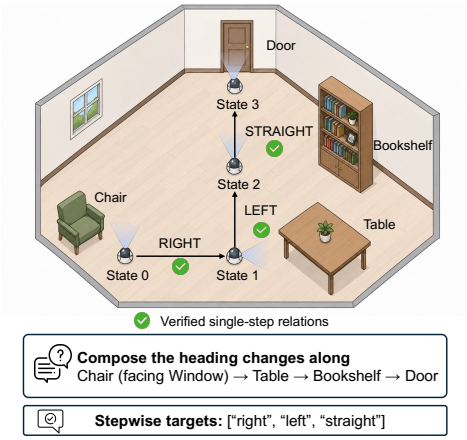

Figure 3: **Reference-frame composition.** Verified single-step relations sharing endpoints form a training chain with stepwise turn targets.

Chain Construction. RC first converts each verified relation according to the direction convention of its source. Compatible relations are then connected along a sequence of landmarks or views to form a path $\mathcal{C} = (v_0, \ldots, v_K)$. The heading obtained at one step provides the reference for the next, turning the path into a sequence of reference-frame updates. The target at each step is recomputed from the source annotations; Appendix A.4 details the construction and verification.

Stepwise Supervision. Each retained relation is converted under its source coordinate convention into a canonical clockwise planar turn increment $\delta_k \in [0, 360^\circ)$. RC describes the resulting reference-frame composition as

$$h_0 = 0, \qquad h_k = (h_{k-1} + \delta_k) \bmod 360^\circ. \tag{6}$$

Here h_k is the reference-frame heading after transition k, for $1 \le k \le K$. RC uses the turn increments and, when available, their cumulative headings as stepwise targets.

We apply RC to VSI-590K and the view-change subset of SPAR-7M to construct four types of supervision (Yang et al., 2025c; Zhang et al., 2025): exact turn angles and discrete turning actions from VSI-590K, and yaw changes between consecutive views and cumulative headings from SPAR-7M. We format each target type as a structured assistant response and train on these samples together with standard instruction data using the same autoregressive SFT objective.

3.3 Joint Training Objective

Pairwise Geometric Prediction. Answer-level SFT constrains the final response but does not provide an explicit target for the relations encoded by the geometric branch. Because the geometric kernel operates on token pairs, we use the relative 3D displacement of ordered token pairs to directly supervise the geometric representation before fusion. We decompose the displacement into distance and unit direction to constrain its magnitude and orientation. At the layer $\ell^\star$ selected for pairwise geometric supervision, we read $\mathbf{h}_i = \phi_g^{\ell^\star}(G_i^{\ell^\star}) \in \mathbb{R}^d$ after the shared output projection and before kernel fusion. We sample ordered pairs of visual tokens with valid geometry within and across frames, with $d_{ij} = \|\mathbf{x}_j - \mathbf{x}_i\|_2$. For each sampled pair, we construct the pair representation and its distance and direction targets as follows:

$$\begin{aligned} \mathbf{z}_{ij} &= [\mathbf{h}_i; \mathbf{h}_j; \mathbf{h}_j - \mathbf{h}_i; \mathbf{h}_i \odot \mathbf{h}_j] \in \mathbb{R}^{4d}, \\ y_{ij}^d &= \log(1 + d_{ij}), \qquad \mathbf{u}_{ij} = \frac{\mathbf{x}_j - \mathbf{x}_i}{d_{ij}} \quad (d_{ij} > \varepsilon). \end{aligned} \tag{7}$$

Here, $\odot$ denotes elementwise multiplication. A prediction head maps $\mathbf{z}_{ij}$ to a predicted log-distance $\widehat{y}_{ij}^d$ and an L2-normalized direction $\widehat{\mathbf{u}}_{ij}$. For a sample n, we use a Huber loss on log distance, $\ell_{\text{dist}}^{(n)}$,

Table 1: **Spatial reasoning on VSI-Bench and ReVSI-32.** Both panels report overall scores and task results. Higher is better. Darker and lighter shading mark the best and second best results in each column. * denotes results obtained using public checkpoints and official evaluation code.

(a) VSI-Bench

Model	Overall	Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order
Qwen2.5-VL-7B	29.30	25.20	10.50	36.40	29.60	38.40	38.00	29.80	26.80
Qwen3-VL-8B	57.90	67.60	47.00	76.30	61.90	58.00	51.00	35.00	66.30
VG-LLM-8B	62.20	71.40	56.80	69.00	69.10	67.90	83.20	47.40	32.50
VLM-3R-7B	60.90	70.20	49.40	69.20	67.10	65.40	80.50	45.40	40.10
Cambrian-S-7B	67.50	73.20	50.50	74.90	72.20	71.10	76.20	41.80	80.10
SpatialStack-5B	67.50	71.00	55.60	69.10	68.20	67.30	84.10	41.20	**83.50**
Q-GeoMem-7B	71.20	74.80	59.90	76.50	75.30	71.10	87.20	50.00	74.60
GeoThinker-7B	68.50	71.96*	56.89*	74.34*	73.37*	70.42*	85.89*	46.39*	68.77*
GeoThinker-8B	72.60	73.93*	**62.49***	76.23*	**75.31***	72.96*	90.29*	49.48*	76.21*
DK-VLM-7B	69.42	72.02	54.17	74.01	64.97	68.73	87.41	56.19	77.83
DK-VLM-8B	**74.33**	**75.91**	60.48	**77.82**	72.78	**74.79**	**90.48**	**60.31**	82.04

(b) ReVSI-32

Model	Overall	Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan
Qwen3-VL-8B-Instruct	45.60	36.20	47.10	67.20	45.30	50.20	37.50	36.00
InternVL3.5-38B	53.10	43.40	56.50	70.00	56.90	55.80	47.80	41.50
VLM-3R-7B	50.10	41.60	61.60	64.80	52.50	46.50	**49.50**	34.10
GeoThinker-7B	48.96*	37.66*	65.22*	62.16*	50.28*	38.74*	47.15*	41.47*
GeoThinker-8B	54.16*	36.99*	**71.23***	65.67*	53.32*	61.35*	47.54*	43.02*
DK-VLM-7B	52.02	**44.30**	61.69	66.65	60.20	37.58	46.84	46.90
DK-VLM-8B	**58.28**	42.23	70.55	**72.70**	**60.99**	**65.51**	48.35	**47.67**

and a cosine loss on direction, $\ell_{\text{dir}}^{(n)}$, combined as $\ell_{\text{pair}}^{(n)} = \ell_{\text{dist}}^{(n)} + 0.5\,\ell_{\text{dir}}^{(n)}$. Averaging over samples with valid token pairs gives $\mathcal{L}_{\text{pair}}$; Appendix A.3 specifies the exact reductions and empty-set cases.

Unified Objective. The two objectives are jointly optimized in a single training stage:

$$\mathcal{L} = \mathcal{L}_{\text{SFT}} + \lambda(t)\mathcal{L}_{\text{pair}}, \tag{8}$$

where $\lambda(t)$ controls the pairwise-loss weight. The prediction head is used only during training and removed for inference.

4 Experiments

4.1 Experimental Setup

Implementation Details. We build DK-VLM on Qwen2.5-VL-7B and Qwen3-VL-8B (Bai et al., 2025b;a). A frozen VGGT-Ω estimates 3D points and camera poses from the input images (Wang et al., 2026a). We also evaluate alternative geometry frontends without retraining and report the results in Appendix A.10. We uniformly sample up to 32 frames from each training video. For both models, we uniformly select five decoder layers to introduce Dual-Kernel Attention and apply rank-64 LoRA (Hu et al., 2022) to the language model's Q/K/V/O projections. The geometric projections, gates, and pairwise prediction head are also trained, while the visual encoder remains frozen. For the 7B and 8B variants, the trainable parameter counts are only 2.88% and 3.86% of their respective backbone parameter counts. Both models use AdamW (Loshchilov & Hutter, 2019) with a main learning rate of 10^{-4} and a global batch size of 32. Further implementation details are provided in Appendix A.6.

Training Data. We use the training data from GeoThinker's scaled regime (Li et al., 2026b) and retain all original instruction samples. We construct approximately 1.6M RC samples with stepwise supervision from existing VSI-590K and SPAR annotations. After filtering and sampling, we use

Table 2: **Results across six spatial benchmarks.** Avg. is the mean of the six overall scores. Higher is better.

Model	Avg.	VSI-Bench	MMSI-Bench	MindCube	ViewSpatial	SITE	CV-Bench
Qwen2.5-VL-7B	40.31	29.30	26.80	36.00	36.80	37.60	75.40
Qwen3-VL-8B	47.73	57.90	28.80	29.80	39.00	45.80	**85.10**
Cambrian-S-7B	47.28	67.50	25.80	39.60	40.90	33.00	76.90
VLM-3R-7B	45.40	60.90	27.90	40.00	40.50	31.30	71.80
VG-LLM-8B	51.05	62.20	30.00	36.10	45.80	50.50	81.70
GeoThinker-7B	60.43	68.50	**31.70**	83.60	41.90	54.80	82.10
GeoThinker-8B	62.23	72.60	30.90	83.00	45.90	55.90	**85.10**
DK-VLM-7B	60.84	69.42	29.95	83.81	44.98	57.47	79.38
DK-VLM-8B	**64.09**	**74.33**	30.79	**83.83**	**51.25**	**59.81**	84.52

Table 3: **Effect of RC supervision on public tasks.** Both models use Qwen3-VL-8B. Person denotes the average score across three person-perspective tasks in ViewSpatial: relative direction, object view orientation, and scene simulation relative direction.

Model	VSI Route	ReVSI Route	MMSI Multi-Step	ViewSpatial Person
DK-VLM (Full)	**60.31**	**47.67**	**34.34**	**61.72**
DK-VLM w/o RC	51.21	43.48	29.19	56.32

150K RC training samples containing only two- and three-step chains. We train on the combined dataset for one epoch, jointly optimizing the SFT and pairwise geometric objectives.

Evaluation. We evaluate on seven spatial benchmarks: VSI-Bench, ReVSI-32, MMSI-Bench, MindCube, ViewSpatial, SITE, and CV-Bench (Yang et al., 2025a; Zhang et al., 2026b; Yang et al., 2025d; Wang et al., 2025b; Li et al., 2025a; Wang et al., 2025d; Tong et al., 2024). We follow GeoThinker's evaluation settings on the six shared benchmarks (Li et al., 2026b) and use the official 32-frame setting for ReVSI. Our comparisons cover base VLMs (Wang et al., 2025c), spatial post-training methods, and geometry-enhanced models (Li et al., 2026b; Zhang et al., 2026a; Gao et al., 2026). We also evaluate RC on public tasks involving multiple reasoning steps or changes in reference frame, and assess general video understanding on VideoMME and MVBench (Fu et al., 2025; Li et al., 2024).

4.2 Spatial Reasoning

VSI-Bench and ReVSI-32. DK-VLM-8B achieves the highest overall scores on both benchmarks, reaching 74.33 and 58.28, respectively (Table 1). On VSI-Bench, both the 7B and 8B variants outperform the corresponding GeoThinker models. The 8B variant also outperforms GeoThinker-8B on route planning in both benchmarks, showing that the full method improves performance on tasks that require combining several spatial relations.

Results across Spatial Benchmarks. To assess overall spatial understanding, we further compare models across six spatial benchmarks (Table 2). DK-VLM-8B achieves the highest average score of 64.09, outperforming the spatial post-training and geometry-enhanced methods in the table. It also ranks first on four of the six benchmarks, showing that the gains extend beyond a single benchmark.

4.3 Effect of RC Supervision

To assess whether RC supervision improves performance on public tasks involving multiple reasoning steps or changes in reference frame, we compare the full DK-VLM-8B with a version trained without RC supervision (Table 3). Both retain Dual-Kernel Attention and pairwise geometric supervision. RC improves route planning on VSI-Bench and ReVSI-32, as well as multi-step reasoning on MMSI-Bench. These gains show that geometric relation modeling benefits from additional stepwise supervision, helping the model use spatial relations across reasoning steps. Improvements on the

person-perspective tasks in ViewSpatial further suggest that the benefits extend beyond sequential reasoning to spatial judgments under a different reference frame.

Performance across Chain Lengths. To assess whether the benefits of RC extend to chain lengths unseen during training, we evaluate stepwise spatial prediction on a held-out set constructed using our RC procedure, with four-step chains excluded from RC training. Progress measures the fraction of steps correct before the first error; Chain Acc. measures full-sequence accuracy. Table 4 shows that RC improves both metrics at every evaluated length. The gains on unseen four-step chains suggest that RC helps the model apply spatial relations more consistently across reasoning steps.

Table 4: **Performance across reasoning chain lengths.** Each entry reports Progress / Chain Acc. (%). See Appendix A.8 for metric definitions.

Chain Length	w/o RC	w/ RC
2	10.6 / 2.2	**52.2 / 45.6**
3	9.0 / 0.3	**50.0 / 37.5**
4 (unseen)	8.2 / 0.0	**35.4 / 20.6**

4.4 General Video Understanding

We evaluate whether the gains in spatial reasoning are accompanied by a decline in general video understanding. The baseline is Qwen3-VL-8B fine-tuned on the same data, including RC samples. DK-VLM-8B scores 1.63 points higher on VideoMME and 0.49 points lower on MVBench (Table 5), indicating comparable general video performance.

Table 5: **General video understanding.** Parentheses show changes from the corresponding Qwen SFT baseline in percentage points.

Model	VideoMME	MVBench
Qwen3-VL-8B + SFT	64.59	**68.42**
DK-VLM-8B	**66.22** (+1.63)	67.93 (−0.49)

4.5 Ablation Studies

We use Qwen3-VL-8B to examine the contributions of geometric transforms and pairwise supervision, as well as the combined effect of the architecture and RC supervision, on VSI-Bench.

Geometric Modeling Components. Table 6 shows that using Point GTA and Camera GTA together outperforms using either alone. Replacing both with identity transforms lowers the score by 4.31 percentage points, with the full geometric branch and its parameters retained, showing that the gains extend beyond added capacity. Pairwise geometric supervision improves the model's ability to make spatial judgments.

Table 6: **Component ablations.**

Configuration	Overall
Full	**74.33**
w/o Point GTA	71.81
w/o Camera GTA	72.46
w/o Point & Camera GTA	70.02
w/o Pairwise Supervision	73.20

Architecture and RC Supervision. To examine whether geometric relation modeling and RC supervision are complementary, we train DK-VLM and Qwen3-VL-8B on the same instruction data, each with and without RC. RC yields a larger gain on DK-VLM (Table 7), supporting the complementarity between geometric relation modeling and stepwise supervision.

Table 7: **Effect of RC supervision.**

Model	w/o RC	w/ RC	Δ
Backbone	68.80	69.20	+0.40
DK-VLM	73.28	**74.33**	+1.05

5 Conclusion

We present DK-VLM, which introduces 3D geometry as position into the attention computation of vision-language models. Dual-Kernel Attention preserves the native semantic kernel and uses Point GTA and Camera GTA to model relations between scene points and camera motion across views. Reference-Frame Composition supervision reorganizes existing single-step annotations into training chains with stepwise targets, further improving spatial reasoning over multiple steps. Among the compared methods, DK-VLM-8B achieves the highest average score across six spatial benchmarks while preserving general video understanding. Ablations further support the complementarity between geometric relation modeling and stepwise supervision.

AI USE STATEMENT

We used AI assistants to help draft, translate, and polish the manuscript, retrieve literature, and implement code. The authors reviewed the AI-assisted content and code and take full responsibility for the accuracy and integrity of this work.

REFERENCES

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities, 2024. URL https://arxiv.org/abs/2401.12168.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. VLM-3R: Vision-language models augmented with instruction-aligned 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31054–31065, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fan_VLM-3R_Vision-Language_Models_Augmented_with_Instruction-Aligned_3D_Reconstruction_CVPR_2026_paper.html.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

Xianqiang Gao, Qizhi Chen, Delin Qu, Haoming Song, Zhigang Wang, Bin Zhao, Dong Wang, and Xuelong Li. Q-GeoMem: Question-guided geometric memory for video spatial reasoning, 2026. URL https://arxiv.org/abs/2605.27318.

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. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. ViewSpatial-Bench: Evaluating multi-perspective spatial localization in vision-language models, 2025a. URL https://arxiv.org/abs/2505.21500.

Dingming Li, Yingxiu Zhao, Xinrui Cheng, Kangheng Lin, Hongbo Peng, Hongxing Li, Zixuan Wang, Yuhong Dai, Haodong Li, Jia Wang, Yukang Shi, Liang Zhao, Jianjian Sun, Zheng Ge, Xiangyu Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. SpatialEvo: Self-evolving spatial intelligence via deterministic geometric environments, 2026a. URL https://arxiv.org/abs/2604.14144.

Haoyuan Li, Qihang Cao, Tao Tang, Kun Xiang, Zihan Guo, Jianhua Han, JiaWang Bian, Hang Xu, and Xiaodan Liang. Thinking with geometry: Active geometry integration for spatial reasoning, 2026b. URL https://arxiv.org/abs/2602.06037.

Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. SpatialLadder: Progressive training for spatial reasoning in vision-language models, 2025b. URL https://arxiv.org/abs/2510.08531.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Li_MVBench_A_Comprehensive_Multi-modal_Video_Understanding_Benchmark_CVPR_2024_paper.html.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems*, 2025c. URL https://papers.neurips.cc/paper_files/paper/2025/hash/17a7075094632c88cccdd86270ad715b-Abstract-Conference.html.

Huizhi Liang, Yichao Shen, Yu Deng, Sicheng Xu, Zhiyuan Feng, Tong Zhang, Yaobo Liang, and Jiaolong Yang. HiSpatial: Taming hierarchical 3d spatial understanding in vision-language models, 2026. URL https://arxiv.org/abs/2603.25411.

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, 2025. URL https://arxiv.org/abs/2511.10647.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/20e6b4dd2b1f82bc599c593882f67f75-Abstract-Conference.html.

Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3d reconstruction priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16695–16705, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Murai_MASt3R-SLAM_Real-Time_Dense_SLAM_with_3D_Reconstruction_Priors_CVPR_2025_paper.html.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. SpaceR: Reinforcing MLLMs in video spatial reasoning, 2025. URL https://arxiv.org/abs/2504.01805.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs, 2024. URL https://arxiv.org/abs/2406.16860.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-ω, 2026a. URL https://arxiv.org/abs/2605.15195.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Jiajun Wu, Li Fei-Fei, and Manling Li. MindCube: Spatial mental modeling from limited views, 2025b. URL https://arxiv.org/abs/2506.21458.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025c. URL https://arxiv.org/abs/2508.18265.

Wenqi Wang, Reuben Tan, Pengyue Zhu, Jianwei Yang, Zhengyuan Yang, Lijuan Wang, Andrey Kolobov, Jianfeng Gao, and Boqing Gong. SITE: towards spatial intelligence thorough evaluation, 2025d. URL https://arxiv.org/abs/2505.05456.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/11a09e0aaa74867c6b0719c639fc09f8-Abstract-Conference.html.

Yu Wu, Minsik Jeon, Jen-Hao Rick Chang, Oncel Tuzel, and Shubham Tulsiani. RayRoPE: Projective ray positional encoding for multi-view attention, 2026. URL https://arxiv.org/abs/2601.15275.

Chendong Xiang, Jiajun Liu, Jintao Zhang, Xiao Yang, Zhengwei Fang, Shizun Wang, Zijun Wang, Yingtian Zou, Hang Su, and Jun Zhu. Geometry-aware rotary position embedding for consistent video world model, 2026. URL https://arxiv.org/abs/2602.07854.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10632–10643, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Thinking_in_Space_How_Multimodal_Large_Language_Models_See_Remember_CVPR_2025_paper.html.

Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, Yi Lin, and Hengshuang Zhao. Visual spatial tuning, 2025b. URL https://arxiv.org/abs/2511.05491.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, Daohan Lu, Rob Fergus, Yann LeCun, Li Fei-Fei, and Saining Xie. Cambrian-S: Towards spatial supersensing in video, 2025c. URL https://arxiv.org/abs/2511.04670.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. MMSI-Bench: A benchmark for multi-image spatial intelligence, 2025d. URL https://arxiv.org/abs/2505.23764.

Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. From flatland to space: Teaching vision-language models to perceive and reason in 3d. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e3a0db7c0a191854c176af1d20cdec80-Paper-Datasets_and_Benchmarks_Track.pdf.

Jian Zhang, Shijie Zhou, Bangya Liu, Achuta Kadambi, and Zhiwen Fan. SpatialStack: Layered geometry-language fusion for 3d VLM spatial reasoning, 2026a. URL https://arxiv.org/abs/2603.27437.

Yiming Zhang, Jiacheng Chen, Jiaqi Tan, Yongsen Mao, Wenhu Chen, and Angel X. Chang. ReVSI: Rebuilding visual spatial intelligence evaluation for accurate assessment of VLM 3d reasoning, 2026b. URL https://arxiv.org/abs/2604.24300.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3D LLM: Learning position-aware video representation for 3d scene understanding, 2024. URL https://arxiv.org/abs/2412.00493.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing MLLMs with 3d vision geometry priors. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1d8dc55c1f6cf124af840ce1d92d1896-Abstract-Conference.html.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. LLaVA-3D: A simple yet effective pathway to empowering LMMs with 3d-awareness, 2024. URL https://arxiv.org/abs/2409.18125.

A Additional Method Details

A.1 Geometry Preprocessing

The geometric kernel of DK-VLM uses 3D points and camera poses aligned with visual tokens, without relying on intermediate features from a specific geometry frontend. For each merged visual token, we map the center of its image region to the frontend's dense 3D point map, accounting for resizing, cropping, and padding in both image preprocessing pipelines. Within each video, we filter out low-confidence points using the 20th percentile of finite depth confidence values as the threshold. We obtain each token's 3D coordinates by bilinear interpolation over the retained neighboring points, with the interpolation weights renormalized. All visual tokens from the same frame share that frame's camera pose.

We construct these geometric quantities from the frontend outputs and express them in the coordinate frame of the first camera. Within each video, we center and scale all point coordinates together and apply the same scale factor to camera translations. Point GTA uses the processed scene points as translation anchors and the source frame's rotation relative to the first frame. Camera GTA uses the relative camera poses. This common representation allows different geometry frontends to use the same geometric attention computation.

A.2 Derivation of the Point GTA Relative Transform

In both DK-VLM-7B and DK-VLM-8B, each geometric attention head has 128 channels: 48 for Point GTA, 24 for Camera GTA, and 56 remaining channels. The point and camera subspaces contain 12 and 6 four-dimensional blocks, respectively. For the 12 Point GTA blocks, we use geometrically spaced displacement scales from 0.25 to 4.0, given by $\lambda_b = 0.25 \cdot 16^{b/11}$ for $b = 0, \ldots, 11$.

For the Point GTA transform in Section 3.1,

$$T_{i,b}^{\mathrm{pt}} = \begin{bmatrix} R_i^{\mathrm{rel}} & \lambda_b \mathbf{x}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{9}$$

its inverse is

$$\left(T_{i,b}^{\mathrm{pt}}\right)^{-1} = \begin{bmatrix} {R_i^{\mathrm{rel}}}^\top & -\lambda_b {R_i^{\mathrm{rel}}}^\top \mathbf{x}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{10}$$

Multiplication with the transform of token j gives the complete pairwise relation

$$\left(T_{i,b}^{\mathrm{pt}}\right)^{-1} T_{j,b}^{\mathrm{pt}} = \begin{bmatrix} {R_i^{\mathrm{rel}}}^\top R_j^{\mathrm{rel}} & \lambda_b {R_i^{\mathrm{rel}}}^\top (\mathbf{x}_j - \mathbf{x}_i) \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{11}$$

The upper-left and upper-right blocks are the relative source-view rotation R_{ij}^{view} and scaled point displacement $\mathbf{t}_{ij,b}^{\mathrm{pt}}$ retained in the main text.

A.3 Pairwise Loss Reduction and Boundary Cases

Prediction Head and Pair Sampling. Both models use LayerNorm followed by a two-layer MLP with GELU activation to map the $4d$-dimensional pair features to a log-distance prediction and a three-dimensional direction prediction, which is L2-normalized. The MLP hidden dimension is d, set to 3584 and 4096 for the 7B and 8B variants, respectively.

We use a sampling budget of 256 ordered token pairs per sample, targeting 50% cross-frame pairs for multi-frame inputs. Within-frame and cross-frame candidates are sampled separately by distance tertiles, with unused quotas reassigned to the other group; single-frame inputs use only within-frame pairs. The direction loss includes only pairs with distances greater than $\varepsilon = 10^{-6}$.

For sample n, let $\mathcal{J}_n$ denote the indices of geometry-valid visual tokens. The sampled ordered-pair set and its direction-valid subset are

$$\begin{aligned}\mathcal{P}_n &\subseteq \{(i,j) : i,j \in \mathcal{J}_n,\ i \neq j\},\\ \mathcal{V}_n &= \{(i,j) \in \mathcal{P}_n : d_{ij} > \varepsilon\}.\end{aligned} \tag{12}$$

For each sample with $|\mathcal{P}_n| > 0$, the distance and direction terms are

$$\begin{aligned}\ell_{\text{dist}}^{(n)} &= \frac{1}{|\mathcal{P}_n|} \sum_{(i,j)\in\mathcal{P}_n} \text{Huber}_{\delta=1}(\widehat{y}_{ij}^{d}, y_{ij}^{d}),\\ \ell_{\text{dir}}^{(n)} &= \begin{cases} \dfrac{1}{|\mathcal{V}_n|} \displaystyle\sum_{(i,j)\in\mathcal{V}_n} \left(1 - \widehat{\mathbf{u}}_{ij}^{\top}\mathbf{u}_{ij}\right), & |\mathcal{V}_n| > 0,\\ 0, & |\mathcal{V}_n| = 0, \end{cases}\\ \ell_{\text{pair}}^{(n)} &= \ell_{\text{dist}}^{(n)} + 0.5\,\ell_{\text{dir}}^{(n)}.\end{aligned} \tag{13}$$

Let $\mathcal{B}_+ = \{n : |\mathcal{P}_n| > 0\}$. The pairwise loss over the minibatch, including the case with no valid pairs, is

$$\mathcal{L}_{\text{pair}} = \begin{cases} \dfrac{1}{|\mathcal{B}_+|} \displaystyle\sum_{n\in\mathcal{B}_+} \ell_{\text{pair}}^{(n)}, & |\mathcal{B}_+| > 0,\\ 0, & |\mathcal{B}_+| = 0. \end{cases} \tag{14}$$

Pairs with $d_{ij} \leq \varepsilon$, including zero-distance pairs, remain in the distance term but do not enter the direction denominator. When $\mathcal{V}_n = \varnothing$, the direction loss is zero but remains connected to the predicted directions in the computation graph. Samples with $\mathcal{P}_n = \varnothing$ do not enter the pairwise batch average; when $\mathcal{B}_+ = \varnothing$, the batch pairwise loss is zero. Otherwise, samples with valid pairs contribute equally to $\mathcal{L}_{\text{pair}}$.

A.4 Reference-Frame Composition Details

Reference-Frame Composition uses planar angular relations extracted from the source annotations. We write $\langle z \rangle_{360} = z \bmod 360^\circ$ for normalization to $[0, 360^\circ)$, with clockwise rotation positive, and use the corresponding circular distance $d_\circ$ for tolerance checks. We first parse and normalize the source relations, then apply consistency checks specific to each source to obtain the verified single-step annotations used for chain construction. After these checks, scenes are partitioned, and chains are constructed only within one scene and split. During partitioning, scene groups that overlap VSI-Bench are isolated from the RC pool. Finally, an independent validator rereads the source annotations, recomputes every derived target, and verifies scene and split membership together with source provenance. These checks cross-validate source angle annotations and their algebraic consistency; they do not use images, XYZ coordinates, or ground-truth 3D states.

VSI-RC. From each VSI-590K `absolute_direction_object` annotation (Yang et al., 2025c), the parser extracts

$$(\text{media}, O, A, B, \text{convention}, \alpha(O,A,B)), \tag{15}$$

where O is the current location, A is the facing anchor, and B is the target landmark. A clockwise raw angle is normalized directly, whereas a counterclockwise angle is converted as $\langle 360^\circ - a_{\text{raw}} \rangle_{360}$. The normalized angle must agree with the corresponding `relative_direction_object` label under the source four-quadrant convention. Inverse relations and angle closure are checked with a 2° circular tolerance:

$$\begin{aligned} d_\circ(\langle \alpha(O,A,B) + \alpha(O,B,A) \rangle_{360}, 0) &\leq 2^\circ,\\ d_\circ(\langle \alpha(O,A,C) - \alpha(O,A,B) \rangle_{360}, \alpha(O,B,C)) &\leq 2^\circ. \end{aligned} \tag{16}$$

Verified relations are connected only within the same media item. A relation extends the current path when its origin is the terminal landmark and its facing anchor is the preceding landmark; paths with

repeated landmarks are rejected. After the initial turn, a landmark path $\mathcal{C} = (v_0, \ldots, v_K)$ induces the ordered reference-state transitions $(v_{i-1}, v_i) \to (v_i, v_{i+1})$ for $1 \le i < K$. Let a_0 be the initial facing anchor. At subsequent steps, the source annotation uses the direction toward the previous landmark as its reference. This reference is opposite to the direction of arrival, requiring a 180° correction. Converting the source facing convention to the traversal-centered convention gives

$$\delta_k^{\mathrm{VSI}} = \begin{cases} \alpha(v_0, a_0, v_1), & k = 1, \\ \langle \alpha(v_{k-1}, v_{k-2}, v_k) - 180^\circ \rangle_{360}, & 2 \le k \le K. \end{cases} \tag{17}$$

The exact-turn target retains the full sequence $(\delta_1^{\mathrm{VSI}}, \ldots, \delta_K^{\mathrm{VSI}})$. The discrete-action target is created only when every turn lies in one of the conservative bins: straight for $[0^\circ, 30^\circ] \cup [330^\circ, 360^\circ)$, right for $[60^\circ, 120^\circ]$, back for $[150^\circ, 210^\circ]$, and left for $[240^\circ, 300^\circ]$. VSI-RC does not serialize cumulative headings. Its prompt specifies the initial reference point, the initial bearing target, and the ordered waypoints while withholding the derived turn sequence.

SPAR-RC. From each SPAR-7M `view_change_infer` annotation with decodable structured camera-motion fields (Zhang et al., 2025), the parser extracts the source and target view identifiers and the yaw component. Yaw is converted to the same clockwise-positive angular convention; pitch and translation fields are not used in RC. Repeated records for the same directed view pair are deduplicated. For three available relations $A \to B$, $B \to C$, and $A \to C$, triangle closure requires

$$d_\circ(\theta_{AC}, \langle \theta_{AB} + \theta_{BC} \rangle_{360}) \le 5^\circ. \tag{18}$$

Accepted directed relations are connected when the target view of one edge matches the source view of the next; paths with repeated views are rejected. For transition $v_{k-1} \to v_k$, $\delta_k^{\mathrm{SPAR}} = \theta_{v_{k-1} v_k}$; cumulative headings follow the recurrence in Section 3.2. SPAR-RC applies no facing-to-traversal 180° correction. Its prompt provides the ordered views and their images and uses the first view as the heading origin while withholding the target sequence.

Supervision Targets. RC produces four target types: VSI `angle` contains exact turn sequences; VSI `action` contains the corresponding discrete actions for eligible chains; SPAR `delta` contains local yaw increments; and SPAR `heading` contains cumulative headings relative to the initial view. Each target sequence is serialized as a JSON array in the assistant response of a separate training sample. No sample jointly emits two target types. The validator independently reconstructs these arrays and checks their provenance against the source relations before they enter autoregressive SFT.

A.5 Output Projection and Gate Initialization

At each selected layer, we initialize the shared output projection ϕ_g^ℓ by copying the native output projection. The two projections have separate parameters. At positions with valid geometry, both kernel outputs use ϕ_g^ℓ; at other positions, the semantic output uses the native output projection.

We initialize $g_\ell = 0$, giving both kernels an initial mixing weight of 0.5. This avoids strongly suppressing the geometric branch at the start of training and keeps the sigmoid gate away from saturation. Each selected layer then learns its own mixing ratio.

A.6 Implementation Details and Parameter Counts

We introduce Dual-Kernel Attention at layers 4, 9, 14, 19, and 24 for DK-VLM-7B and at layers 6, 12, 18, 27, and 33 for DK-VLM-8B, using zero-based indices. The pairwise prediction head reads the pre-fusion geometric features at layer 19 for the 7B model and layer 27 for the 8B model. For both models, the pairwise loss weight increases linearly to 0.1 over the first 5% of training steps.

Both models are trained on 32 NVIDIA H200 GPUs for one epoch. Training takes approximately 11 days for the 7B variant and 10 days for the 8B variant.

Table 8 summarizes the parameter counts of both variants.

Table 8: **Parameter counts of DK-VLM.** Parameter counts are reported in millions (M). Trainable parameters include LoRA, the geometric branch, and the pairwise prediction head. The prediction head is used only during training and removed at inference. All counts exclude the frozen VGGT-Ω frontend.

Component	DK-VLM-7B	DK-VLM-8B
Backbone	8,292.17	8,767.12
LoRA	40.37	61.34
Geometric branch (including gates)	146.82	209.72
Pairwise prediction head	51.43	67.16
Total trainable parameters	**238.62**	**338.22**
Relative to backbone	**2.88%**	**3.86%**

A.7 Training Data Composition

We follow the training mixture reported for GeoThinker's scaled regime (Li et al., 2026b), which contains approximately 1.8M instruction samples. The spatial reasoning data comprise Cambrian-S VSI-590K (590K), SPAR (234K), VLM-3R VSI-Bench instructions (205K), VLM-3R VSTI-Bench instructions (132K), and the MindCube training set (10K). The general video data comprise LLaVA-Hound (64K), PhysGame PhysInstruct (140K), and a general video subset of Cambrian-S-3M (430K).

RC Sample Selection. We construct a pool of approximately 1.6M RC samples from the verified source annotations. For SPAR-RC, we retain one question per view tuple and randomly assign either local yaw increments or cumulative headings as the prediction target. We also retain at most one sample for each image set.

To reduce the concentration of small turns, we group individual turn magnitudes into six 30° bins over the folded angular range from 0° to 180°. Each chain is represented by the multiset of its step-level bins. We use deficit-greedy sampling to favor underrepresented bins, subject to limits of 96 samples per scene and 40 appearances per image.

For VSI-RC, we retain only samples with video inputs and limit each clip to 40 questions. The retained VSI-RC samples are repeated twice to increase their weight in the training mixture. The final RC training set contains 150K entries and includes only two- and three-step chains.

A.8 RC Chain Evaluation

Evaluation Set Construction. We construct an evaluation set using the same RC procedure, with no overlap with the training samples. It includes four target types: local yaw increments and cumulative headings from SPAR-RC, and turn angles and discrete actions from VSI-RC. We evaluate two-, three-, and four-step chains. RC training uses only two- and three-step chains.

For each chain length, we select up to 80 examples per target type. To cover different angular configurations, SPAR-RC examples are grouped by data source (ScanNet, ScanNet++, or S3D) and the 30° bin containing the largest folded turn magnitude in the chain. VSI-RC angle examples are grouped by the same angular criterion, while action examples are grouped by their complete action sequences. Examples are selected in round-robin order across groups within each target type.

This gives 320 examples each for two- and three-step chains. Only eight distinct four-step action examples are available, giving 248 examples for four-step chains. The composition of the evaluation set is shown in Table 9.

Evaluation Metrics. Both models are evaluated on the same examples using the same correctness criteria. For numeric angle targets, a step is correct if the shortest circular difference between the predicted and target angles is at most 10°; discrete actions require exact label matching. We report Progress and Chain Acc. separately for each chain length.

For a target sequence of length n, let k be the number of consecutive correct predictions from the first step. Progress is $100k/n$. Once a prediction is incorrect, that step and all subsequent steps receive no

Table 9: **Composition of the RC evaluation set.** Counts denote evaluation examples for each target type and chain length.

Chain Length	SPAR Delta	SPAR Heading	VSI Angle	VSI Action	Total
2	80	80	80	80	320
3	80	80	80	80	320
4	80	80	80	8	248

credit, even if a later prediction matches its target. For example, a three-step prediction with only its first step correct receives a Progress score of 33.3%.

Chain Acc. assigns a score of 100 only when the predicted sequence has the same length as the target and every step is correct, and 0 otherwise. Both metrics are averaged over all evaluation examples of the corresponding chain length, pooling the four target types.

A.9 Injection Layer Configuration

We examine the number and placement of Dual-Kernel Attention layers in DK-VLM-8B. To reduce training cost, all configurations are trained on the same 10% subset of the full training data, with the remaining training settings held constant. All layer indices are zero-based. Pairwise geometric supervision is applied at the penultimate selected injection layer in each configuration.

Number of Injection Layers. Increasing the number of injection layers from three to five improves VSI-Bench performance by 0.77 percentage points, while increasing it from five to seven yields only a further 0.06-point gain (Table 10). We therefore use five injection layers by default, achieving performance close to the seven-layer configuration with fewer geometric branches.

Table 10: **Effect of the number of injection layers on VSI-Bench.**

Injection Layers	Layer Indices	VSI-Bench
3	6, 18, 33	62.28
5	6, 12, 18, 27, 33	63.05
7	6, 9, 12, 18, 24, 27, 33	**63.11**

Placement of Injection Layers. With the number of injection layers fixed at five, we compare early, distributed, and late placements. The distributed configuration achieves the highest score among the three, outperforming the early and late configurations by 0.62 and 1.76 percentage points, respectively (Table 11). These results support our default choice of distributing geometric injection across model depth.

Table 11: **Effect of injection layer placement on VSI-Bench.** All configurations use five injection layers.

Placement	Layer Indices	VSI-Bench
Early	6, 9, 12, 15, 18	62.43
Distributed	6, 12, 18, 27, 33	**63.05**
Late	18, 21, 24, 27, 30	61.29

A.10 Geometry Frontends and Input Frame Counts

We examine the sensitivity of DK-VLM-8B to the geometry frontend and the number of input frames on VSI-Bench. Both experiments use the same trained checkpoint, changing only the inference configuration without additional training. We fix the input to 32 frames when comparing frontends and use VGGT-Ω throughout the frame-count comparison.

Geometry Frontends. DK-VLM uses estimated 3D points and camera poses rather than frontend-specific intermediate features. To test whether this interface supports interchangeable frontends, we replace the default VGGT-Ω with other geometry frontends at inference. As shown in Table 12, VGGT-Ω achieves the highest score of 74.33, while the other frontends score between 72.99 and 73.91, a decrease of 0.42–1.34 percentage points from the default setting. These results show that DK-VLM can use different geometry frontends without retraining while maintaining spatial reasoning performance close to that of the default configuration.

Table 12: **Geometry frontend comparison on VSI-Bench.** All configurations use the same DK-VLM-8B checkpoint and 32 input frames, without retraining.

Geometry Frontend	Overall
VGGT-Ω	**74.33**
VGGT	73.91
MASt3R-SLAM	73.64
Pi3X	73.62
DA3-Large	72.99

Input Frame Counts. With VGGT-Ω fixed, we increase the number of input frames at inference from 8 to 128. Performance improves with more frames, but the gains diminish: increasing from 8 to 32 frames yields a 9.04-point improvement, whereas increasing from 32 to 128 frames adds only 0.99 points (Table 13). The 128-frame setting achieves the highest score of 75.32; our default 32-frame setting approaches this performance with fewer input frames.

Table 13: **Effect of input frame count on VSI-Bench.** All configurations use the same DK-VLM-8B checkpoint and VGGT-Ω frontend, without additional training.

Input Frames	Overall
8	65.29
16	69.94
32	74.33
64	75.13
128	**75.32**

A.11 General Video Evaluation Settings

We use up to 32 frames per video for both VideoMME and MVBench, and evaluate VideoMME without subtitles.

A.12 Comparison of Geometry Injection Strategies

We compare different geometry injection strategies based on Qwen3-VL-8B using the same 10% training data subset and optimization settings. All configurations omit the pairwise geometric loss. In addition to Qwen SFT and DK-VLM, we consider three alternatives: adding 3D positional embeddings to visual tokens once at the input; adding 3D positional embeddings to visual tokens at the same five selected decoder layers as DK-VLM, while retaining single-stream attention at each layer; and applying single-kernel GTA at three or five selected decoder layers, directly replacing the positional encoding in the corresponding channels of the native attention. DK-VLM scores 62.78 in this comparison; the score of 63.05 reported in the injection layer experiments (Appendix A.9) corresponds to the five-layer configuration with the pairwise geometric loss retained.

Table 14: **Comparison of geometry injection strategies on VSI-Bench.** All configurations omit the pairwise geometric loss.

Configuration	**VSI-Bench Overall ↑**
Qwen3-VL-8B + SFT	59.48
3D positional embeddings at input	59.21
3D positional embeddings at five layers	59.74
Single-kernel GTA (three layers)	45.98
Single-kernel GTA (five layers)	39.14
DK-VLM-8B (five layers)	**62.78**

Adding 3D positional embeddings at the input and at five layers yields scores of 59.21 and 59.74, respectively, close to the Qwen SFT score of 59.48. In contrast, DK-VLM without the pairwise geometric loss still achieves 62.78, outperforming Qwen SFT and five-layer embedding addition by 3.30 and 3.04 points, respectively (Table 14). This shows that, under the current training settings, dual-kernel geometry injection outperforms the two tested additive positional embedding strategies.

Single-kernel GTA scores 45.98 with three injection layers and 39.14 with five, both below the Qwen SFT baseline. With five injection layers, DK-VLM outperforms single-kernel GTA by 23.64 points. In the five-layer single-kernel configuration, the performance drop is accompanied by garbled outputs and poorer instruction following, including responses that do not follow the required answer format. These observations suggest that directly replacing the native positional encoding can impair model performance and output quality in this setting. Overall, these results provide empirical support for our design of retaining the native attention pathway while introducing spatial relations through a separate geometric kernel.