SpatialStack: Layered Geometry-Language Fusion for 3D VLM Spatial Reasoning

Jian Zhang[1*] Shijie Zhou[2,3*] Bangya Liu[4*] Achuta Kadambi[2] Zhiwen Fan[5]

[1]XMU [2]UCLA [3]Google [4]UW-Madison [5]TAMU

https://spatial-stack.github.io/

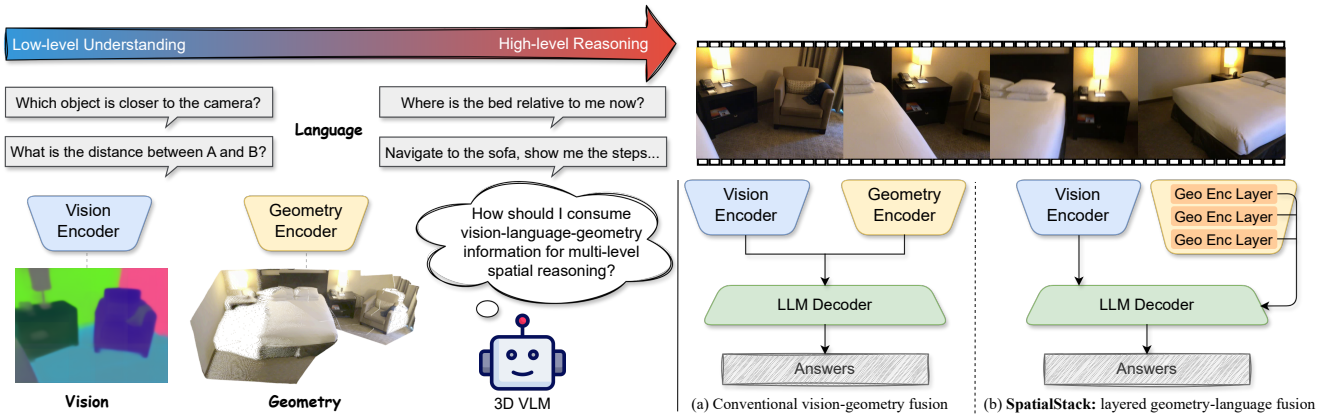

Figure 1. **SpatialStack: Layered Geometry-Language Fusion.** Conventional VLMs (a) fuse only a single deep geometry feature with vision tokens, which limits both fine-grained spatial understanding and high-level spatial reasoning. SpatialStack (b) instead stacks multi-level geometry features and injects them hierarchically into successive LLM decoder layers, yielding stronger 3D spatial understanding across benchmarks.

Abstract

Large vision-language models (VLMs) still struggle with reliable 3D spatial reasoning, a core capability for embodied and physical AI systems. This limitation arises from their inability to capture fine-grained 3D geometry and spatial relationships. While recent efforts have introduced multi-view geometry transformers into VLMs, they typically fuse only the deep-layer features from vision and geometry encoders, discarding rich hierarchical signals and creating a fundamental bottleneck for spatial understanding. To overcome this, we propose SpatialStack, a general hierarchical fusion framework that progressively aligns vision, geometry, and language representations across the model hierarchy. Moving beyond conventional late-stage vision-geometry fusion, SpatialStack stacks and synchronizes multi-level geometric features with the language backbone, enabling the model to capture both local geometric precision and global contextual semantics. Building upon this framework, we develop VLM-SpatialStack, a model that achieves state-of-the-art performance on multiple 3D spatial reasoning benchmarks. Extensive experiments and ablations demonstrate that our multi-level fusion strategy consistently enhances 3D understanding and generalizes robustly across diverse spatial reasoning tasks, establishing SpatialStack as an effective and extensible design paradigm for vision-language-geometry integration in next-generation multimodal physical AI systems.

1. Introduction

Understanding and reasoning about physical space are fundamental capabilities for any intelligent system that aims to perceive, communicate, and act in the physical world. Motivated by this, recent work on spatial reasoning aims to enable embodied agents to interpret scene layouts, predict interactions, and plan actions in 3D environments, forming a cognitive bridge between perception and action [13, 35, 50, 52, 53, 60]. Despite remarkable progress in large vision-language models (VLMs), reliable spatial reasoning remains challenging, as these models often fail to effectively encode 3D geometry and spatial relationships and to associate them with language instructions, which are es-

*Equal contribution.

sential for everyday spatial tasks that require both low-level and high-level reasoning. For instance, they struggle to estimate relative distances in static scenes [28, 51] and cannot reliably distinguish "left" from "right" when reasoning about motion in dynamic environments [65]. In embodied AI applications such as robotic navigation, manipulation, and spatial assistance under XR, such limitations prevent VLMs from grounding their understanding in the complex and dynamic physical world.

Noticing these limitations of conventional VLMs, many recent works still prioritize image-level semantic alignment over the understanding of spatial and geometric structures [22, 37, 42]. Bridging this gap requires unifying geometric awareness with vision-language reasoning within a single framework, which is a key step toward reliable spatial intelligence. This naturally raises a fundamental question: *How can vision–language–geometry be effectively unified in VLMs to enable reliable spatial reasoning?* An initial line of work sought to compensate for these weaknesses by integrating explicit geometric inputs (e.g., precomputed point clouds or depth maps) into VLMs. For instance, early models like 3D-LLM [16] and LEO [17] used external point cloud encoders, while later methods like LLaVA-3D [66] and Video-3D LLM [61] introduced lightweight encoders for RGB-D fusion. However, the reliance on these external, pre-processed inputs significantly limits their applicability. In parallel, rapid advancements in end-to-end multi-view geometry transformers, including DUST3R [48], CUT3R [47], and VGGT [46], have provided a more unified and powerful alternative to map uncalibrated images to 3D point maps. These models can infer rich geometric attributes such as depth, camera pose, and 3D structure directly from multi-view images, thereby bypassing traditional, computationally expensive geometric pipelines (e.g., Structure-from-Motion [41]). Inspired by this progress, recent multimodal models, such as Spatial-MLLM [50], VLM-3R [13], and VG-LLM [60], have begun integrating these geometry encoders into VLM frameworks, showing initial promise in improving spatial reasoning.

Nevertheless, most of these integrations focus only on fusing the final-layer features of geometry transformers with features from vision encoders. This is a critical limitation, as many geometry encoders adopt the DPT architecture [39], which explicitly extracts multi-level representations from different transformer layers to recover detailed geometric information. At the same time, a generalizable spatial-visual fusion mechanism has to account for hierarchical real-world tasks, ranging from low-level depth estimation and surface reconstruction to high-level relational reasoning and goal-directed planning. By sampling only the last layer, existing models discard the rich hierarchical geometric cues embedded in intermediate layers and overlook how different levels of geometric and semantic features contribute to spatial reasoning. Unsurprisingly, this single-level fusion design can improve performance on specific spatial benchmarks but creates a bottleneck that fundamentally constrains 3D understanding.

In this paper, we are motivated by the hierarchical nature of spatial reasoning tasks in 3D environments, and we systematically study how fusion layers across vision encoders, geometry encoders, and large language model (LLM) decoders affect multimodal spatial reasoning. Our analysis first shows that geometry-language fusion in multimodal LLMs follows a hierarchical pattern similar to vision encoding: shallow features enhance fine-grained spatial perception, while deeper features support high-level contextual reasoning. Building on these insights, we introduce SpatialStack, a general hierarchical fusion framework that integrates multi-level geometric features into multimodal LLMs. As shown in Fig. 1, unlike prior methods that fuse geometry only at deep encoder layers, SpatialStack progressively aligns geometric and language representations throughout the model hierarchy, capturing both detailed local geometry and global semantic context. Extensive experiments on multiple benchmarks demonstrate that our approach significantly improves 3D spatial reasoning, achieving strong performance on tasks requiring both detailed perception and holistic spatial understanding.

We summarize our **contributions** as follows:

- We present the first systematic analysis of how fusion layers across vision encoders, geometry encoders, and LLM decoders affect the granularity of spatial reasoning. Our layer-wise study reveals a hierarchical geometry–language correspondence, where shallow layers capture fine spatial details and deeper layers encode global structure and context.
- We propose **SpatialStack**, a general hierarchical fusion framework that progressively aligns multi-level geometric and language features. This design goes beyond conventional final-stage vision-language fusion and supports joint reasoning over local and global spatial context.
- While SpatialStack is model-agnostic and can be applied to any base multimodal LLM, we develop **VLM-SpatialStack** as a concrete realization using the Qwen series. Extensive experiments and ablation studies across multiple benchmarks show that SpatialStack achieves state-of-the-art performance and strong generalization on diverse 3D spatial reasoning tasks.

2. Related Work

Large Multimodal Models (MLLMs) Early works such as CLIP [38] demonstrated the efficacy of learning joint vision-language representations from web-scale image-text pairs through contrastive pre-training. This paradigm was extended by subsequent models like Flamingo [1], which bridged powerful pre-trained vision and language mod-

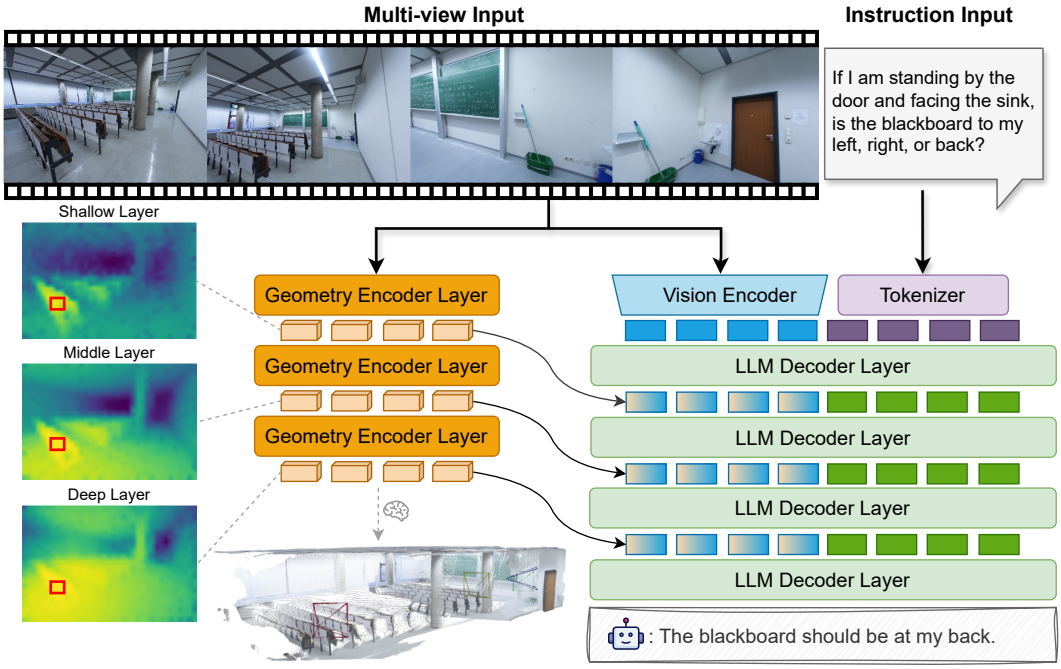

Figure 2. **Architecture of SpatialStack.** A standard VLM backbone is coupled with a multi-view geometry encoder whose layer-wise features are processed by layer-specific projectors and sequentially injected into the LLM decoder, progressively integrating geometric cues. Explanation of the similarity heatmaps on the left is provided in Sec. 3. This multi-level injection preserves both fine-grained geometric structure and high-level spatial context, supporting more reliable low-level understanding and high-level reasoning.

els, and the BLIP series [25, 26], which introduced bootstrapping methods and lightweight querying transformers to unify understanding and generation. A significant advancement in MLLM development was the advent of visual instruction tuning, effectively employed by models such as InstructBLIP [11] to enhance instruction-following capabilities. Models like Qwen2.5-VL [2], LLaVA [31] and MiniGPT-4 [67] popularized a simple and effective architecture for this tuning, connecting a pre-trained vision encoder to a large language model (LLM) using only a simple projection layer. This simple design has spurred research into more effective fusion strategies, such as exploring different visual encoders [21] or deeper token stacking [36]. This architectural blueprint, while powerful for general-purpose multimodal chat and semantic understanding, established a paradigm of fusing only the final-layer visual features with the language backbone. Consequently, as noted by recent analyses [22, 37, 42], these models are trained primarily for semantic alignment and often fail to capture the fine-grained spatial and geometric structures essential for physical reasoning.

Spatial Reasoning in Vision-Language Models The limitations of standard MLLMs in spatial reasoning have been well-documented, prompting recent efforts to quantify these deficiencies through benchmarks like VSI Bench [51], Spar Bench [56], BLINK [15], and Cambrian-1 [45]. Cambrian-S [53] further demonstrated a lack of "implicit 3D spatial cognition", while VLM4D [65] was the first to highlight the challenges of spatiotemporal (4D) reasoning in dynamic scenarios, followed by more recent efforts on dynamic 4D understanding and world modeling [18, 49]. To address these gaps, one line of work focused on injecting explicit 3D data, such as 3D-LLM [16] which processes point clouds, or more lightweight approaches like LLaVA-3D [66] and Video-3D LLM [61] that endow MLLMs with 3D awareness, with applications in embodied tasks [17]. A different strategy enhanced spatial abilities through novel training paradigms. Spatial-SSRL [35] introduced a self-supervised reinforcement learning framework, and Visual Spatial Tuning (VST) [52] proposed a comprehensive tuning framework with a large-scale dataset (VST-P) and a progressive pipeline. These latter methods enhance spatial intelligence but primarily focus on training objectives and data augmentation rather than the core fusion architecture.

Vision-Language-Geometry Fusion The integration of explicit geometric reasoning within MLLMs has been recently catalyzed by the advent of powerful, feed-forward geometry encoders. Models such as DUST3R series [23, 48] and CUT3R [47] can infer dense, consistent point maps from unposed multi-view images, while VGGT [46] introduced a unified transformer to predict diverse 3D attributes from video. The availability of these rich geometric

features has inspired two parallel fusion paradigms. One line of work focuses on building *explicit* spatial semantic representations, such as methods that distill 2D image or video foundation model features into 3D or 4D explicit feature field representations [44, 63, 64], build queryable 3D world models by fusing pixel-aligned features into 3D maps [20], or map images directly to semantic radiance fields [12]. A parallel approach, which our work follows, *implicitly* fuses geometric priors into the latent space of the MLLM. Recent models have shown initial promise in this direction: Spatial-MLLM [50] employs a dual-encoder architecture, VG-LLM [60] fuses features at the patch level, VLM-3R [13] introduces a cross-attention mechanism, and SSR [34] focuses on rationale-guided fusion. However, as identified in our analysis, these integrations typically fuse only the final-layer features from the geometry and vision encoders [50, 60]. This single-level fusion design discards the rich, hierarchical geometric cues embedded in intermediate layers, creating a fundamental bottleneck for fine-grained spatial reasoning. Our work, SpatialStack, directly addresses this limitation by introducing a hierarchical fusion framework that progressively aligns multi-level geometry features with the language backbone.

3. How Multi-level Geometry Features Facilitate Spatial Reasoning

Qualitative Analysis To validate our motivation, we begin by examining why relying solely on deep-layer geometry features is insufficient. As illustrated in Fig. 2, we take one input view and unflatten the tokens from different layers of the geometry encoder back into their original $H \times W$ spatial layout. We then select a small patch (red bounding box) as the region of interest (ROI) and compute patch-wise similarity maps between the ROI and all other spatial locations: brighter regions indicate higher similarity, and darker regions indicate lower similarity. We observe a clear trend: shallow layers retain sharp local structures and well-defined geometric boundaries, while deeper layers produce overly homogeneous activations, where many regions appear similar in latent space despite having distinct physical geometry. This mismatch suggests that deep geometry features lose fine-grained spatial cues critical for reasoning about scene layout and spatial relations. These findings motivate our approach: leveraging multi-level geometry features, especially shallow-layer cues, to enrich spatial grounding and improve fine-grained 3D spatial reasoning in VLMs.

Quantitative Analysis We further investigate how geometric features from different layers influence spatial reasoning performance. Firstly, we follow the difficulty hierarchy defined in SPAR [56] dataset, which categorizes spatial tasks based on the required complexity of spatial understanding. SPAR divides spatial tasks into three cognitive levels: *perception (low)*, *reasoning (medium)*, and *imagination (high)*. Low-level tasks emphasize fundamental geometric perception, such as single-view depth estimation and distance comparison based on local pixel/feature cues; high-level tasks require aggregating spatial information across multiple viewpoints for global spatial reasoning, such as cross-view object spatial relations and path reasoning (see Fig. 3).

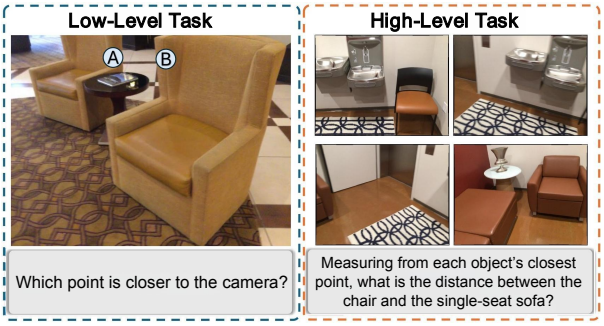

Figure 3. **Examples of spatial tasks at different levels.** The left example (*Low-Level Task*) targets fine-grained geometric perception, such as determining which of two points is closer to the camera. The right example (*High-Level Task*) requires higher-level spatial reasoning, where the model must estimate the distance between two objects by comparing their closest points in 3D space.

Based on this criterion, our **low-level tasks** evaluating fundamental perception include BLINK's *relative depth* [15] and specific tasks in SPAR-Bench [56]: depth (*Depth-OC/OO/OC-MV/OO-MV*) and absolute distance (*Dist-OC/OO/OC-MV/OO-MV*) (see [56] for more details). Conversely, all *VSI-Bench* tasks are categorized as **high-level tasks**, as they require complex multi-view spatial fusion and 3D relational reasoning. We do not include SPAR's medium or high tasks in this study, as we aim to establish a clearer two-level comparison that isolates the distinct effects of geometric feature integration on *basic perception ability* versus *complex spatial reasoning ability*.

Under the task definitions above, we further conduct a quantitative analysis of the performance impact of injecting geometric features from different layers into VLMs. Specifically, following VG-LLM [60], we extract geometric features from a single layer of the geometry encoder (VGGT [46]), project them through a projector, and add them to the last-layer features of the vision encoder. We denote this geometry-vision fusion as GVF in the rest of our paper. The fused geometry-vision features are then concatenated with text tokens and fed into the LLM decoder. We experiment with injecting geometric features from the 4th, 11th, 17th, and 23rd layers, and evaluate the performance on the two task levels.

As shown in Fig. 4, the results demonstrate that the choice of injection layer has a significant impact on differ-

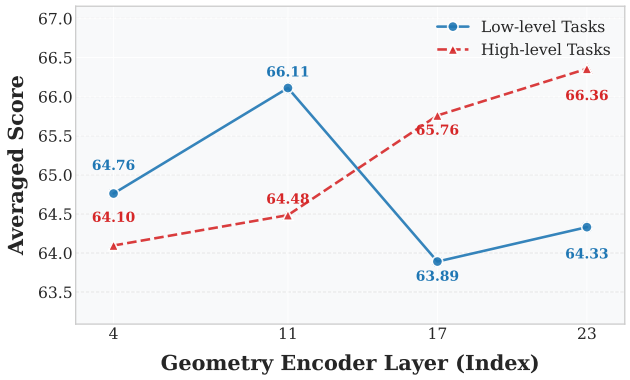

Figure 4. **Effect of Geometry Injection Layers on Spatial Tasks.** Deeper layers improve high-level tasks, while low-level tasks peak at layer 11 and decline at deeper layers, suggesting a trade-off between fine-grained perception and higher-level reasoning.

ent levels of spatial tasks: as the injection layer becomes deeper, the performance on low-level tasks declines, while the performance on high-level tasks improves significantly. This phenomenon suggests that geometric features from different layers play distinct roles in spatial understanding: features from shallower layers provide fine-grained local geometric cues beneficial for basic spatial perception, whereas deeper features encode more global structural and semantic relationships, making them more suitable for complex spatial reasoning.

Given the complementary strengths of shallow and deep features, a multi-layer fusion strategy should intuitively enhance both perception and reasoning. To test this, Tab. 1 compares various fusion strategies against Qwen3.5 [43], the base model fine-tuned on our spatial reasoning datasets without any geometric enhancements. Surprisingly, naive multi-layer fusion fails to achieve the best. Instead, it yields a compromised performance, falling behind the 11th-layer single-fusion on low-level tasks and the 23rd-layer on high-level tasks. This sub-optimal outcome reveals that simply adding hierarchical features into the vision pathway causes feature interference rather than synergy. This highlights that merely extracting multi-level cues is insufficient; the true challenge lies in the fusion strategy—a realization that serves as the primary catalyst for SpatialStack.

4. Where to fuse Multi-level Geometry Features

The observation of feature interference during naive vision-pathway fusion in Sec. 3 prompts a critical question: *where* and *how* should these hierarchical geometric features be integrated to maximally enhance a VLM's spatial reasoning? Should they be confined to the vision encoder, or directly injected into the language model?

Model	Low-Level Avg	High-Level Avg	Overall
Qwen3.5 (fine-tuned)	61.37	64.76	63.07
Single-layer (geo enc: 11)	**66.11**	64.48	65.30
Single-layer (geo enc: 17)	63.89	65.76	64.83
Single-layer (geo enc: 23)	64.33	**66.36**	**65.35**
Multi-Layer Fusion	64.69	65.15	64.92

Table 1. **Ablation Results on Geometry Token Fusion Depth.** Simply fusing multi-layer geometry features to the visual features yields suboptimal performance, while selecting an appropriate single geometry encoder layer achieves better task-specific trade-offs.

4.1. SpatialStack: Geometry-Language Fusion

While most prior works [13, 50, 60] confine geometric enhancements to the vision encoder, we hypothesize that injecting these features directly into the Large Language Model (LLM) provides a more flexible, high-capacity space for multi-scale spatial reasoning. Inspired by DeepStack's [36] success in stacking visual tokens within the LLM, we shift geometry integration to the language side and propose SpatialStack: a novel, first-of-its-kind layered geometry–language fusion framework.

As shown in Fig. 2, SpatialStack performs multi-level fusion between a geometry encoder and an LLM decoder. Its key idea is to inject geometric features from multiple layers of the geometry encoder into corresponding layers of the LLM, forming a hierarchy of geometric representations. This progressive stacking introduces geometric cues throughout the decoding process, strengthening spatial grounding and improving reasoning across tasks of varying difficulty.

Importantly, SpatialStack is a general framework that can be integrated with any open-source VLM. We instantiate VLM-SpatialStack using the latest Qwen3.5 [43] as our primary base model. To ensure a fair comparison with existing baselines [50, 53, 60], we also provide an instantiation based on the same base model they use, Qwen2.5-VL [2].

4.2. VLM-SpatialStack

VLM Architecture. Given K input frames $\{\mathbf{I}_k \in \mathbb{R}^{H\times W\times 3}\}_{k=1}^{K}$, each frame is encoded by a shared vision encoder into tokens $\mathbf{V}_k \in \mathbb{R}^{N\times D_{\text{vis}}}$, where p is the patch size and $N = \frac{H}{p}\times\frac{W}{p}$ is the number of patch tokens per frame. A spatial merger groups every 2×2 neighboring patches (stride factor $s=2$), producing $\tilde{\mathbf{V}}_k \in \mathbb{R}^{N'\times D_{\text{lang}}}$ with $N' = \frac{N}{s^2} = \frac{HW}{(ps)^2}$. Merged tokens from all frames are concatenated along the sequence dimension:

$$\tilde{\mathbf{V}} = [\tilde{\mathbf{V}}_1; \ldots; \tilde{\mathbf{V}}_K] \in \mathbb{R}^{(KN')\times D_{\text{lang}}}.$$

The concatenated visual tokens $\tilde{\mathbf{V}}$ and text tokens $\mathbf{T} \in \mathbb{R}^{M\times D_{\text{lang}}}$ form the multimodal input sequence $\mathbf{H}_0 = [\tilde{\mathbf{V}}; \mathbf{T}]$. This sequence is then processed by L stacked

transformer layers in the LLM decoder:

$$\mathbf{H}_L^{\text{llm}} = f_L^{\text{llm}}\Big(f_{L-1}^{\text{llm}}\big(\cdots f_1^{\text{llm}}(\mathbf{H}_0)\big)\Big), \tag{1}$$

where $\mathbf{H}_L^{\text{llm}}$ denotes the final hidden representations produced by the LLM decoder for downstream prediction.

Geometry Encoder. We employ the Visual Geometry Grounded Transformer (VGGT) [46] as our geometry encoder. Given the same set of K input images $\{\mathbf{I}_k \in \mathbb{R}^{H\times W\times 3}\}_{k=1}^{K}$, each image is divided into non-overlapping patches of size $p \times p$, resulting in $N = (H/p)\times(W/p)$ patch tokens. In addition to the patch tokens, VGGT includes camera and register tokens to encode view-specific and shared geometric context. The initial token sequence for view k is thus

$$\mathbf{Z}_0^{(k)} = [\,\mathbf{c}_k;\, \mathbf{r}_k;\, \mathbf{p}_k\,] \in \mathbb{R}^{(1+R+N)\times D_{\text{geo}}}, \tag{2}$$

where $\mathbf{p}_k$ denotes the patch tokens of image $\mathbf{I}_k$. All view-specific sequences are concatenated and jointly processed by L stacked transformer layers:

$$\mathbf{Z}_L = f_L^{\text{geo}}\Big(f_{L-1}^{\text{geo}}\big(\cdots f_1^{\text{geo}}([\mathbf{Z}_0^{(1)};\ldots;\mathbf{Z}_0^{(K)}])\big)\Big), \tag{3}$$

where $f_l^{\text{geo}}(\cdot)$ denotes the l-th transformer layer in VGGT. While the original VGGT employs Dense Prediction Transformer (DPT) heads [39] for outputs such as depth, point clouds, and camera parameters, we instead extract intermediate hidden states $\mathbf{Z}_l$ from selected layers as multi-view geometric features for fusion with the vision–language model.

Layered Geometry–Language Fusion. As illustrated in Fig. 2, we extract multi-level patch features $\mathbf{Z}_{l_i}$ from the geometry encoder defined in Eq. (3). Specifically, we take the patch-token outputs of the l_i-th layers ($l_i \in \{11, 17, 23\}$, counted from zero) of VGGT after removing camera and register tokens, thereby yielding $\mathbf{Z}_{l_i} \in \mathbb{R}^{(KN)\times D_{\text{geo}}}$ that represent geometric information at different, progressively richer abstraction levels. Each feature $\mathbf{Z}_{l_i}$ is processed by a layer-specific geometry token merger $\mathcal{M}_{\text{geo}}^{(l_i)}$ to align its spatial resolution and embedding dimension with that of $\mathbf{H}$:

$$\mathbf{G}_{l_i} = \mathcal{M}_{\text{geo}}^{(l_i)}(\mathbf{Z}_{l_i}), \quad \mathbf{G}_{l_i} \in \mathbb{R}^{N'\times D_{\text{lang}}}. \tag{4}$$

Finally, the geometry features $\{\mathbf{G}_{l_1}, \mathbf{G}_{l_2}, \mathbf{G}_{l_3}\}$ extracted from VGGT layers $\{11, 17, 23\}$ are injected into LLM decoder layers $\{0, 1, 2\}$ as additive residuals:

$$\mathbf{H}^{(j)'} = \mathbf{H}^{(j)} + \mathbf{G}_{l_j}, \quad j \in \{0, 1, 2\}. \tag{5}$$

Methods	VSI-Bench	SPAR-Bench	BLINK-Spatial	CV-Bench	Overall
Qwen3.5 (fine-tuned)	64.76	68.75	**56.10**	84.49	68.52
GVF-L23 (VG-LLM [60])	66.36	70.83	51.91	84.64	68.43
GVF-L11/17/23	65.15	71.20	51.28	84.33	67.99
SpatialStack	**67.52**	**71.39**	52.12	**85.53**	**69.14**

Table 2. **Cross-benchmark Ablation.** SpatialStack achieves the best cross-task transfer ability, obtaining the highest scores on **VSI-Bench**, **SPAR-Bench**, **CV-Bench**, and the overall average, while the `Qwen3.5` baseline remains strongest on **BLINK-Spatial**. Gray cells denote the highest value in each column.

Optimization. We train the entire model under a single objective, the next-token negative log-likelihood (cross-entropy):

$$\mathcal{L}_{\text{ce}}(\theta) = -\sum_{i=1}^{|o|} \log P_\theta\big(o^{(i)} \,\big|\, o^{(<i)},\, q,\, \mathcal{C}\big), \tag{6}$$

where q denotes the system prompt and question, $o^{(i)}$ is the i-th token of the ground-truth answer, $o^{(<i)}$ are the preceding answer tokens, and $\mathcal{C}$ represents the multimodal context (e.g., input frames). During instruction tuning, we freeze both the vision encoder and the geometry encoder, and update only the multimodal fusion modules and the LLM decoder. This choice preserves the pretrained visual and geometric representations while allowing the model to learn how to align and integrate them effectively for spatial reasoning. No auxiliary objectives or task-specific losses are introduced; spatial priors emerge purely through unified instruction tuning across diverse spatial tasks.

4.3. SpatialStack vs. Geometry-Vision Fusion

To evaluate the effectiveness of SpatialStack, we compare it against three baselines: base model Qwen3.5, a naive single-layer Geometry–Vision Fusion (GVF-L23) equivalent to VG-LLM [60] built on Qwen3.5, and a naive multi-layer fusion (GVF-L11/17/23). Across four spatial reasoning benchmarks in Tab. 2, SpatialStack achieves the best overall average and obtains the highest scores on *VSI-Bench* [51], *SPAR-Bench* [56], and *CV-Bench* [45]. While the base Qwen3.5 model remains strongest on *BLINK-Spatial* [15], the naive geometry-vision fusion approaches (GVF-L23 and GVF-L11/17/23) suffer severe performance drops on this dataset and fail to generalize effectively across tasks. These results highlight that straightforward visual-pathway injection lacks robust generalization, whereas SpatialStack demonstrates superior cross-task transfer ability.

5. Experiments

We describe our training setup in Sec. 5.1, evaluate VLM-SpatialStack against state-of-the-art methods in Sec. 5.2, and provide extensive ablation studies in Sec. 5.3.

Methods	Rank	Avg.	Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order
			Numerical Answer				Multiple-Choice Answer			
Baseline										
Chance Level (Random)	-	-	-	-	-	-	25.0	36.1	28.3	25.0
Chance Level (Frequency)	-	34.0	62.1	32.0	29.9	33.1	25.1	47.9	28.4	25.2
Proprietary Models (API)										
GPT-4o	2	34.0	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5
Gemini-2.5 Pro	1	51.5	43.8	34.9	64.3	42.8	61.1	47.8	45.9	71.3
Open-source Models										
LongVILA-8B	15	21.6	29.1	9.1	16.7	0.0	29.6	30.7	32.5	25.5
Qwen2.5-VL-3B	14	28.7	33.1	19.4	17.4	24.8	37.3	44.3	31.4	22.0
VILA-1.5-8B	13	28.9	17.4	21.8	50.3	18.8	32.1	34.8	31.0	24.8
LongVA-7B	12	29.2	38.0	16.6	38.9	22.2	33.1	43.3	25.4	15.7
VILA-1.5-40B	11	31.2	22.4	24.8	48.7	22.7	40.5	25.7	31.5	32.9
LLaVA-OneVision-7B	10	32.4	47.7	20.2	47.4	12.3	42.5	35.2	29.4	24.4
LLaVA-Video-7B	9	35.6	48.5	14.0	47.8	24.2	43.5	42.4	34.0	30.6
LLaVA-OneVision-72B	8	40.2	43.5	23.9	57.6	37.5	42.5	39.9	32.5	44.6
LLaVA-Video-72B	7	40.9	48.9	22.8	57.4	35.3	42.4	36.7	35.0	48.6
Spatial-MLLM-4B	6	47.0	65.3	34.8	63.1	45.1	41.3	46.2	33.5	46.3
VG-LLM-4B	5	47.3	66.0	37.8	55.2	59.2	44.6	45.6	33.5	36.4
Qwen3.5-4B	4	53.6	56.5	36.5	67.5	53.8	60.3	57.5	34.0	62.3
Cambrian-S-3B	3	57.3	70.7	40.6	68.0	46.3	64.8	61.9	27.3	78.8
SpatialStack-4B (Qwen2.5)	2	60.9	69.2	45.4	63.0	63.2	57.9	68.4	40.2	79.6
SpatialStack-5B (Qwen3.5)	1	67.5	71.0	55.6	69.1	68.2	67.3	84.1	41.2	83.5

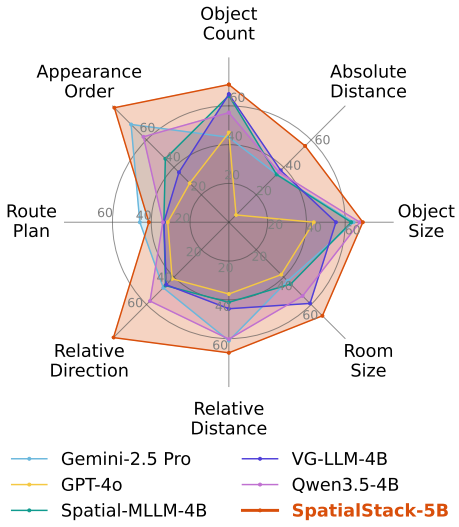

Table 3. **Evaluation on VSI-Bench.** Dark orange cells denote the best *open-source* result in each column, while light orange cells denote the second-best *open-source* result. Group-wise ranks within proprietary and open-source model blocks are highlighted in purple, with dark purple, medium purple, and light purple indicating 1st, 2nd, and 3rd place, respectively.

5.1. Training

Training Datasets Construction. Our training dataset is constructed by sampling from multiple spatial reasoning sources, including the SPAR and LLaVA-Hound subsets used in VG-LLM [60], the ScanNet split adopted in VLM-3R [13], and a selected portion of the VSI-590K corpus [53]. SPAR [56] provides large-scale spatial data generated from reconstructed scenes with 3D ground truth, while LLaVA-Hound [58] offers general-purpose video question–answering samples. VLM-3R reformulates spatial question–answer pairs in a VSI-Bench-style format, producing diverse reasoning tasks such as relative direction, object counting, and absolute distance estimation from real-world 3D–annotated scenes. We sample from these sources to ensure broad coverage of spatial reasoning types and maintain precise alignment between 3D geometry and textual descriptions.

Training Setup. We fine-tune the model using the standard language modeling cross-entropy loss. Training is performed with a batch size of 64 and a learning rate of 1×10^{-5}, optimized using the AdamW optimizer with a warmup ratio of 0.03 and a cosine learning rate schedule. During instruction tuning, the geometry encoder (VGGT) and the vision encoder are kept frozen, while the geometry token merger modules and the LLM decoder are trainable to learn geometry–language alignment.

5.2. Evaluation

We evaluate our model on a diverse set of multimodal benchmarks that test both spatial and general reasoning, including VSI-Bench [51], CV-Bench [45], SPAR-Bench [56], BLINK [15], and Video-MME [14]. These benchmarks cover a wide range of tasks, such as depth and distance estimation, object–relation reasoning, and video-based spatial understanding.

Evaluation on VSI-Bench. We evaluate our model on *VSI-Bench* [51], which contains over 5,000 QA pairs from egocentric indoor videos and includes both Multiple-Choice Answer (MCA) and Numerical Answer (NA) tasks. Following the official protocol, we report mean MCA accuracy and Mean Relative Accuracy for NA across confidence thresholds $C = 0.5, 0.55, \ldots, 0.95$. For comparison, we include representative proprietary models [9, 19], open-source video MLLMs [7, 24, 29, 57, 59], and geometry-aware methods at similar scales [50, 53, 60].

As shown in Tab. 3, we demonstrate that SpatialStack serves as a general and highly effective paradigm for enhancing various VLMs. Applying our framework to both Qwen2.5 [2] and Qwen3.5 [43] yields substantial improvements over their untuned base models. Furthermore, under a fair comparison using the identical Qwen2.5 base model, SpatialStack significantly outperforms other concurrent geometry-aware MLLMs, such as Spatial-MLLM [50], VG-LLM [60], and Cambrian-S [53]. Ultimately, our latest SpatialStack-5B (based on Qwen3.5) establishes a new state-of-the-art among all evaluated open-source models. Notably, despite lacking route-planning data during training, it still surpasses all open-source systems on this task, demonstrating robust zero-shot generalization for high-level spatial reasoning.

Model	2D (%)	3D (%)	Avg. (%)
Proprietary Models (API)			
GPT-4o [19]	74.8	83.0	78.9
Open-source Models			
Mini-Gemini-HD-34B [27]	71.5	79.2	75.4
LLaVA-NeXT-34B [24]	73.0	74.8	73.9
Cambrian-1-34B [45]	74.0	79.7	76.9
SAT-LLaVA-Video-7B [40]	73.0	83.8	78.4
SPAR-8B [56]	72.3	89.1	80.7
Qwen2.5-VL-3B [2]	67.9	70.4	69.2
Qwen3.5-4B [43]	**79.7**	90.2	85.0
Cambrian-S-3B [53]	76.1	76.3	76.2
Dual-Encoder MLLMs			
VG-LLM-4B [60]	71.3	87.7	79.5
SpatialStack-4B (Qwen2.5)	75.4	87.0	81.2
SpatialStack-5B (Qwen3.5)	78.9	**92.2**	**85.5**

Table 4. **Comparison on CV-Bench.** Built on Qwen2.5, SpatialStack-4B outperforms its base model alongside VG-LLM and Cambrian-S. Scaling to Qwen3.5, SpatialStack-5B further improves upon its baseline to set a new state-of-the-art.

Method	MMBench	Video -MME	BLINK	Temp Compass	Overall
Qwen3.5-4B	83.25	62.44	**61.12**	66.84	**68.41**
SpatialStack-5B (Qwen3.5)	**83.42**	**63.74**	55.46	**69.37**	68.00

Table 5. **General Capabilities Evaluation.** Our SpatialStack-5B maintains robust general multimodal and spatial-temporal reasoning capabilities, demonstrating no catastrophic forgetting.

Evaluation on CV-Bench. To assess 2D and 3D spatial perception, we evaluate on *CV-Bench* [45], which measures performance via QA tasks constructed from standard vision datasets [4, 30, 62]. We follow the official protocol and report average accuracy across all task types. As shown in Tab. 4, our two versions of SpatialStack surpass all baselines of similar scale and same base models on both 2D and 3D subsets, demonstrating the benefits of multi-level geometry feature stacking for unified spatial perception.

Evaluation on General-purpose Capabilities. We evaluate SpatialStack on a comprehensive suite of benchmarks: MMBench [32] and Video-MME (general multimodal/video understanding), BLINK (fine-grained visual perception), and TempCompass [33] (spatial-temporal reasoning). Tab. 5 shows that our method maintains robust general capabilities while specializing in spatial-temporal tasks, confirming no catastrophic forgetting.

5.3. Ablation Study

VGGT Layer Selection Ablation. Our selection mirrors VGGT's default indices {4, 11, 17, 23}, but with one adjustment: we excluded layer 4 due to poor performance in preliminary testing (insufficient network depth). This set provides a representative spread of shallow, middle, and deep features. Tab. 6 shows that replacing layer L23 with L21 or L22 (either alone or via multi-layer fusion with L11 and L17, denoted by "+") yields no significant performance changes. This confirms that broad sampling across network depth is more critical than specific layer indices.

Metric	L21	L22	L23	+L21	+L22	+L23
Low-Level	63.54	64.87	64.33	65.89	65.45	64.44
High-Level	65.57	66.51	66.36	65.95	66.78	67.52

Table 6. **Layer Selection Ablation.** Performance comparison of extracting geometry features from different deep VGGT layers (L21, L22, L23) and their multi-layer combinations.

Methods	VSI Bench	SPAR Bench	BLINK Spatial	CV Bench	Overall
Qwen3.5	64.76	68.75	56.10	84.49	68.52
Vision Fusion	64.27	69.68	**56.45**	83.11	68.38
SpatialStack (Reverse)	67.22	**71.97**	50.08	84.82	68.52
SpatialStack (final)	**67.52**	71.39	52.12	**85.53**	**69.14**

Table 7. **Geometry-Language Fusion Order Ablation.** Comparison of our progressive hierarchical alignment against a reverse fusion strategy and baseline models.

Geometry-Language Fusion Order Ablation. We analyze the order of layer-wise geometry-language feature fusion in Tab. 7. SpatialStack uses a progressive hierarchical mapping (L11 stands for layer 11): Geo-L11 $\rightarrow$ LLM-L0, Geo-L17 $\rightarrow$ LLM-L1, and Geo-L23 $\rightarrow$ LLM-L2. This is compared against a Reverse configuration (Geo-L11 $\rightarrow$ LLM-L2, Geo-L17 $\rightarrow$ LLM-L1, Geo-L23 $\rightarrow$ LLM-L0). SpatialStack outperforms both of the Reverse baseline and Vision Fusion baseline in 3 out of 4 benchmarks and achieves a higher overall score, confirming our hierarchical alignment is optimal for preserving spatial information.

6. Conclusion

We introduced SpatialStack, a hierarchical fusion framework bridging the gap between vision, geometry, and language for robust 3D spatial reasoning. Our layer-wise analysis reveals a key correspondence: shallow geometry layers preserve fine-grained spatial details, while deeper layers capture global semantic context. We find that naive multi-layer geometry-vision fusion creates a structural bottleneck, leading to feature interference rather than synergy. By progressively aligning multi-level geometric features with the LLM decoder, SpatialStack preserves both local precision and high-level relational semantics. Extensive evaluations across multiple 3D benchmarks show that our approach achieves state-of-the-art performance among open-source models, exhibiting strong zero-shot generalization without compromising general multimodal capabilities. SpatialStack establishes a new paradigm for vision-language-geometry integration, paving the way for AI systems that truly understand and act within the physical 3D world.

References

[1] 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. 2

[2] Shuai Bai et al. Qwen2.5-VL Technical Report, 2025. 3, 5, 7, 8, 12

[3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 13

[4] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13154–13164, 2023. 8

[5] Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9490–9498. IEEE, 2025. 17

[6] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *CVPR*, pages 14455–14465, 2024. 17

[7] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024. 7

[8] 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. 17

[9] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 7

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 13

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

[12] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. *Advances in neural information processing systems*, 37:40212–40229, 2024. 4

[13] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 1, 2, 4, 5, 7, 13, 14

[14] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118, 2025. 7

[15] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024. 3, 4, 6, 7

[16] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023. 2, 3

[17] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 2, 3

[18] Yuzhi Huang, Kairun Wen, Rongxin Gao, Dongxuan Liu, Yibin Lou, Jie Wu, Jing Xu, Jian Zhang, Zheng Yang, Yunlong Lin, Chenxin Li, Panwang Pan, Junbin Lu, Jingyan Jiang, Xinghao Ding, Yue Huang, and Zhi Wang. Thinking in dynamics: How multimodal large language models perceive, track, and reason dynamics in physical 4d world, 2026. 3

[19] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 7, 8

[20] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023. 4

[21] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023. 3

[22] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's" up" with vision-language models? investigating their struggle with spatial reasoning. *arXiv preprint arXiv:2310.19785*, 2023. 2, 3

[23] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 3

[24] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv*, 2024. 7, 8

[25] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 12763–12779. PMLR, 2022. 3

[26] 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*, pages 19730–19742. PMLR, 2023. 3

[27] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 8

[28] Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. Sti-bench: Are mllms ready for precise spatial-temporal world understanding? *arXiv preprint arXiv:2503.23765*, 2025. 2

[29] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26689–26699, 2024. 7

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 8

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

[32] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024. 8

[33] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8731–8772, 2024. 8

[34] Yang Liu, Ming Ma, Xiaomin Yu, Pengxiang Ding, Han Zhao, Mingyang Sun, Siteng Huang, and Donglin Wang. Ssr: Enhancing depth perception in vision-language models via rationale-guided spatial reasoning. *arXiv preprint arXiv:2505.12448*, 2025. 4

[35] Yuhong Liu, Beichen Zhang, Yuhang Zang, Yuhang Cao, Long Xing, Xiaoyi Dong, Haodong Duan, Dahua Lin, and Jiaqi Wang. Spatial-ssrl: Enhancing spatial understanding via self-supervised reinforcement learning. *arXiv preprint arXiv:2510.27606*, 2025. 1, 3

[36] Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. DeepStack: Deeply Stacking Visual Tokens is Surprisingly Simple and Effective for LMMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3, 5

[37] Jianing Qi, Jiawei Liu, Hao Tang, and Zhigang Zhu. Beyond semantics: Rediscovering spatial awareness in vision-language models. *arXiv preprint arXiv:2503.17349*, 2025. 2, 3

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 2

[39] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 2, 6

[40] Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, et al. Sat: Dynamic spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024. 8

[41] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2

[42] Zineng Tang, Long Lian, Seun Eisape, XuDong Wang, Roei Herzig, Adam Yala, Alane Suhr, Trevor Darrell, and David M Chan. Tulip: Towards unified language-image pre-training. *arXiv preprint arXiv:2503.15485*, 2025. 2, 3

[43] Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, 2026. 5, 7, 8, 12

[44] Anh Thai, Songyou Peng, Kyle Genova, Leonidas Guibas, and Thomas Funkhouser. Splattalk: 3d vqa with gaussian splatting. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 4

[45] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024. 3, 6, 7, 8

[46] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 2, 3, 4, 6, 12

[47] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 2, 3

[48] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2, 3

[49] Kairun Wen, Yuzhi Huang, Runyu Chen, Hui Zheng, Yunlong Lin, Panwang Pan, Chenxin Li, Wenyan Cong, Jian Zhang, Junbin Lu, et al. Dynamicverse: A physically-aware multimodal framework for 4d world modeling. *arXiv preprint arXiv:2512.03000*, 2025. 3

[50] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *Advances in neural information processing systems*, 2025. 1, 2, 4, 5, 7

[51] 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 Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025. 2, 3, 6, 7, 16

[52] Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, et al. Visual spatial tuning. *arXiv preprint arXiv:2511.05491*, 2025. 1, 3

[53] Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-s: Towards spatial supersensing in video. *arXiv preprint arXiv:2511.04670*, 2025. 1, 3, 5, 7, 8, 13, 14

[54] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 13

[55] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019. 12

[56] Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *arXiv preprint arXiv:2503.22976*, 2025. 3, 4, 6, 7, 8, 13

[57] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv*, 2024. 7

[58] Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander G Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 694–717, 2025. 7, 13

[59] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 7

[60] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *Advances in neural information processing systems*, 2025. 1, 2, 4, 5, 6, 7, 8, 13, 14, 15, 16

[61] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8995–9006, 2025. 2, 3

[62] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 8

[63] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 4

[64] Shijie Zhou, Hui Ren, Yijia Weng, Shuwang Zhang, Zhen Wang, Dejia Xu, Zhiwen Fan, Suya You, Zhangyang Wang, Leonidas Guibas, et al. Feature4x: Bridging any monocular video to 4d agentic ai with versatile gaussian feature fields. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14179–14190, 2025. 4

[65] Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. Vlm4d: Towards spatiotemporal awareness in vision language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8600–8612, 2025. 2, 3

[66] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4295–4305, 2025. 2, 3

[67] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 3

SpatialStack: Layered Geometry-Language Fusion for 3D VLM Spatial Reasoning

Supplementary Material

In this supplementary material, we provide comprehensive implementation details and additional experimental results for *SpatialStack*. The content is organized as follows:

- Sec. A elaborates on the detailed architectural components, including the geometry token extraction pipeline and the masked additive fusion mechanism.
- Sec. B describes the composition and statistics of our training dataset mixture.
- Sec. C describes the training details, including input processing and specific training configurations.
- Sec. D provides the detailed evaluation protocols, including the specific benchmarks and metrics.
- Sec. E presents additional baseline comparisons on zero-shot spatial reasoning in CV-Bench.
- Sec. F offers qualitative visualizations contrasting the feature responses of geometry and vision encoders.

A. Architecture Details

To enable both fine-grained and global spatial reasoning, our architecture integrates multi-level geometric cues extracted from VGGT [46] into the VLM. The overall pipeline consists of three stages: geometry token extraction and spatial alignment (Sec. A.1); geometry merging and projection into the language feature space (Sec. A.2); and masked additive fusion that injects geometry exclusively into the visual-token slice of the decoder state (Sec. A.3). The following subsections describe each component in detail.

A.1. Geometry Token Extraction and Preprocessing

We first outline the end-to-end flow of geometry token extraction and alignment before detailing the reshaping and reordering process. At both training and inference time, images are processed by the vision encoder of the chosen base model (Qwen2.5-VL [2] or Qwen3.5 [43]) to generate visual features. For Qwen2.5-VL, this encoding procedure consists of patch embedding, window based attention, and hierarchical patch merging, and produces a sequence of merged vision tokens; for Qwen3.5, the stock vision encoder produces image embeddings that are inserted into the multimodal sequence. In parallel, VGGT (frozen, evaluation mode) emits geometry tokens or layer-wise geometry features from selected internal aggregator layers. These geometry features are subsequently reshaped and reordered, when needed, to match the layout of the visual tokens before fusion, ensuring spatial consistency.

Token Structuring. VGGT produces a sequence of tokens at multiple internal aggregator layers. Each output contains three types of tokens: a *camera token* encoding global viewpoint information; several *register tokens* acting as global latent slots; and a sequence of *patch tokens* representing per-patch geometric features.

Let $h_{\text{patch}} = H/p$ and $w_{\text{patch}} = W/p$ denote the spatial resolution of the VGGT patch tokens. The patch tokens are originally arranged in a flat row-major sequence of length $h_{\text{patch}} \times w_{\text{patch}}$. To align their traversal order with the vision encoder after the spatial merger step, we partition the spatial grid into windows of size $s \times s$, where $s = \texttt{spatial\_merge\_size}$ (default $s = 2$):

$$(h_{\text{patch}},\, w_{\text{patch}}) \;\to\; \big(\tfrac{h_{\text{patch}}}{s},\, s,\, \tfrac{w_{\text{patch}}}{s},\, s\big), \tag{7}$$

and apply a permutation that moves window indices ahead of within-window positions:

$$\big(\tfrac{h_{\text{patch}}}{s},\, s,\, \tfrac{w_{\text{patch}}}{s},\, s\big) \;\to\; \big(\tfrac{h_{\text{patch}}}{s},\, \tfrac{w_{\text{patch}}}{s},\, s,\, s\big). \tag{8}$$

Finally, the reordered grid is flattened back into a 1D sequence:

$$\big(\tfrac{h_{\text{patch}}}{s},\, \tfrac{w_{\text{patch}}}{s},\, s,\, s\big) \;\to\; h_{\text{patch}} \cdot w_{\text{patch}}. \tag{9}$$

This reordering preserves the total number of tokens while changing their traversal order: tokens are enumerated window-by-window rather than row-by-row. As a result, consecutive groups of s^2 geometry tokens correspond to the same spatial region grouped by one merged visual token, ensuring spatial alignment prior to fusion.

A.2. Geometry-to-Language Projection

After reordering the geometry patch tokens to match the traversal order of the merged vision tokens (Sec. A.1), we obtain a 1D sequence

$$\mathbf{Z} \in \mathbb{R}^{(h_{\text{patch}} \cdot w_{\text{patch}}) \times D_{\text{geo}}}, \tag{10}$$

where $h_{\text{patch}} \cdot w_{\text{patch}}$ denotes the total number of spatial tokens and D_{g} is the geometry feature dimension.

Normalization. Following the design of Qwen2.5-VL [2], token-wise RMS normalization [55] is first applied:

$$\mathbf{Z}_{\text{norm}} = \mathrm{RMSNorm}(\mathbf{Z}). \tag{11}$$

Window-wise merging. The normalized tokens are grouped into non-overlapping spatial windows of size $s \times s$.

Each window is flattened and concatenated along the channel dimension, producing a 1D sequence of merged geometry tokens:

$$\tilde{\mathbf{Z}} \in \mathbb{R}^{\left(\frac{h_{\text{patch}}}{s} \cdot \frac{w_{\text{patch}}}{s}\right) \times (s^2 D_{\text{geo}})}. \tag{12}$$

Projection to language space. Each flattened window token is projected to the language decoder dimension by a two-layer MLP:

$$\mathbf{G} = W_2\, \sigma(W_1 \tilde{\mathbf{Z}} + b_1) + b_2, \tag{13}$$

where $W_1 \in \mathbb{R}^{D_{\text{mlp}} \times (s^2 D_{\text{geo}})}$, $b_1 \in \mathbb{R}^{D_{\text{mlp}}}$, $W_2 \in \mathbb{R}^{D_{\text{lang}} \times D_{\text{mlp}}}$, and $b_2 \in \mathbb{R}^{D_{\text{lang}}}$. The projected geometry representation has shape

$$\mathbf{G} \in \mathbb{R}^{\left(\frac{h_{\text{patch}}}{s} \cdot \frac{w_{\text{patch}}}{s}\right) \times D_{\text{lang}}}. \tag{14}$$

A.3. Additive Fusion via Vision-Token Mask

Let $H_l \in \mathbb{R}^{N_{\text{tot}} \times D_{\text{lang}}}$ denote the decoder hidden states at layer l, where N_{tot} is the token sequence length (including system prompt, instruction text, vision tokens, and autoregressive text), and D_{lang} is the decoder hidden dimension.

The projected geometry features from Sec. A.2 are $\mathbf{G}_l \in \mathbb{R}^{N_p \times D_{\text{lang}}}$, where N_p denotes the number of vision tokens participating in fusion (in the default setting without camera tokens and assuming h_{patch} and w_{patch} are divisible by s, $N_p = \frac{h_{\text{patch}}}{s} \cdot \frac{w_{\text{patch}}}{s}$). To locate the visual portion of the sequence, we define a binary mask $M_{\text{vis}} \in \{0,1\}^{N_{\text{tot}}}$, where $M_{\text{vis}}[i] = 1$ if and only if position i corresponds to a visual token.

Additive fusion updates only the masked positions:

$$\mathbf{H}_l \leftarrow \mathbf{H}_l + \text{scatter}\big(\mathbf{G}_l,\ M_{\text{vis}}\big), \tag{15}$$

where $\text{scatter}(\mathbf{G}_l, M_{\text{vis}})$ distributes rows of $\mathbf{G}_l$ sequentially to locations where $M_{\text{vis}} = 1$ and inserts zeros elsewhere.

Equivalently, for each token index i,

$$\mathbf{H}_l[i] \leftarrow \begin{cases} \mathbf{H}_l[i] + \mathbf{G}_l[k], & \text{if } M_{\text{vis}}[i] = 1, \\ \mathbf{H}_l[i], & \text{if } M_{\text{vis}}[i] = 0, \end{cases} \tag{16}$$

where k enumerates the N_p masked positions.

Thus, geometry information is injected exclusively into the vision-token slice of the decoder state, while non-vision tokens (e.g., system prompt and text tokens) remain unchanged. During autoregressive generation, this fusion is applied at the initial prefill step, after which standard decoding proceeds with the updated hidden states.

B. Dataset Details

We construct a balanced dataset of approximately 200k samples, blending spatial expertise with general instruction-following capabilities to facilitate efficient experimentation. Specifically, we sample subsets from **SPAR-234k** and **LLaVA-Hound-64k** (both from *VG-LLM* [60]), as well as the **ScanNet split** of the *VLM-3R* dataset [13], which provides explicit spatial supervision. To enhance perception of object sequences, we additionally include approximately 2k appearance-order instances from the **VSI-590k** [53] collection. As summarized in Tab. A, this composition ensures broad task coverage suitable for controlled architectural ablations.

B.1. Spatial Instruction-Following Data

SPAR (Spatial Perception and Reasoning). SPAR [56] is a large-scale vision–language dataset designed for *spatial perception and reasoning* in complex indoor scenes, featuring diverse question–answer pairs across 33 spatial task types spanning low-level perception to high-level reasoning, and covering single-view, multi-view, and video formats. We build upon the publicly released **SPAR-234k** subset introduced in [60]; the detailed task-type distribution of our sampled training set is illustrated in Fig. A.

VLM-3R. VLM-3R is a spatial QA construction framework based on open-source 3D datasets with geometry, semantic labels, and instance-level annotations, including ScanNet [10], ScanNet++ [54], and ARKitScenes [3]. We use only the ScanNet split, which provides six spatial QA task types: Object Counting, Relative Distance, Relative Direction, Object Size, Absolute Distance, and Room Size. This split does not include Route Planning or Appearance Order tasks.

VSI-590K. VSI-590k is a large-scale spatial instruction-tuning dataset consisting of 590k QA examples from real and simulated indoor environments across 12 task types. For training, we extract a 2k subset corresponding to the appearance-order task derived specifically from the ScanNet portion of VSI-590k, which supplements the absence of appearance-order supervision in the VLM-3R ScanNet split.

We refer to this combined compilation of spatial tasks as **VSI-Type Data**. As visualized in Fig. B, these seven tasks are categorized into three major groups: Configuration, Measurement, and Spatiotemporal, following the taxonomy in the VSI-Bench setting.

B.2. General Video Instruction-Following Data

LLaVA-Hound. LLaVA-Hound [58] is a dataset for video captioning, instruction tuning, and preference alignment, curated from 900k videos sourced from WebVid, VIDAL, and ActivityNet. High-quality captions are produced using GPT-4V from uniformly sampled frames, followed by 240k instruction–answer pairs generated using ChatGPT and 17k preference pairs for Direct Preference Optimization. We use the 64k LLaVA-Hound subset released in VG-LLM, from which 60 percent is sampled to retain general instruction-following and object-grounded reasoning capability while

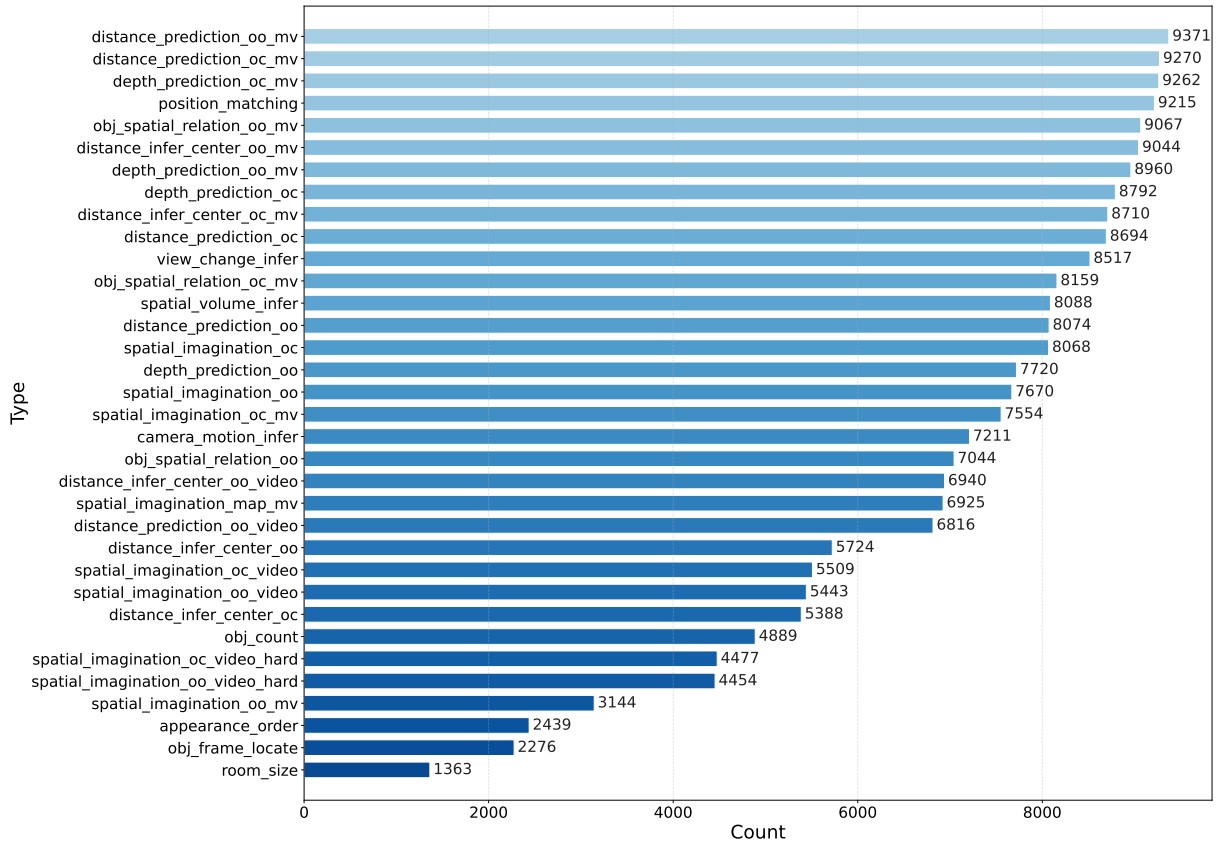

Figure A. **Task-type distribution of the sampled SPAR subset.** The bar chart reports the counts of all 33 spatial task types after randomly sampling 60% of SPAR-234k for training.

Dataset	Raw	Train Subset
SPAR 234k	234k (66.3%)	140k (66.4%)
LLaVA-Hound 64k	63.8k (18.0%)	38.3k (18.1%)
VLM3R-ScanNet	51.8k (14.6%)	31.1k (14.7%)
VSI App-Order	3.8k (1.1%)	1.9k (0.9%)
Total	353k (100%)	212k (100%)

Table A. **Dataset scales and sampled subsets used in our $\sim$ 200k training mixture.** We sample 60% from SPAR-234k, LLaVA-Hound-64k [60], and the ScanNet split of VLM-3R [13], and add $\sim$ 2k appearance-order instances from VSI-590k [53] to compensate for the missing ordering supervision. Percentages indicate each dataset's contribution to the final mixture.

keeping the training scale computationally manageable.

C. Training Details

This section details the implementation of *SpatialStack*, focusing on (1) input processing and (2) training settings. The model is trained via large-scale geometry-aware instruction tuning, where only the language tower and geometry-merger modules are updated, while the vision tower and VGGT remain frozen. All experiments are conducted on 32 NVIDIA A100 GPUs (80GB).

C.1. Input Processing

Videos are first decomposed into individual frame images before entering the multimodal pipeline. A single video token in the prompt is expanded into K consecutive image tokens. For a clip of duration T_{sec} containing F total frames, we uniformly sample $K = \text{clip}(\text{round}(T_{\text{sec}}/\Delta), K_{\min}, K_{\max})$ frame indices from $[0, F-1]$, where Δ denotes the temporal sampling interval.

Each frame (and standalone image) undergoes a unified visual preprocessing pipeline. For SPAR-style training samples, optional task-specific marking is first applied on the original-resolution image: task cues such as points or bounding boxes are drawn according to the provided annotation metadata before any resizing. Transparency is then composited onto a white background and the image is converted to RGB.

Next, we resize the image while preserving aspect ratio to a target size of 518 pixels. In the default crop-based set-

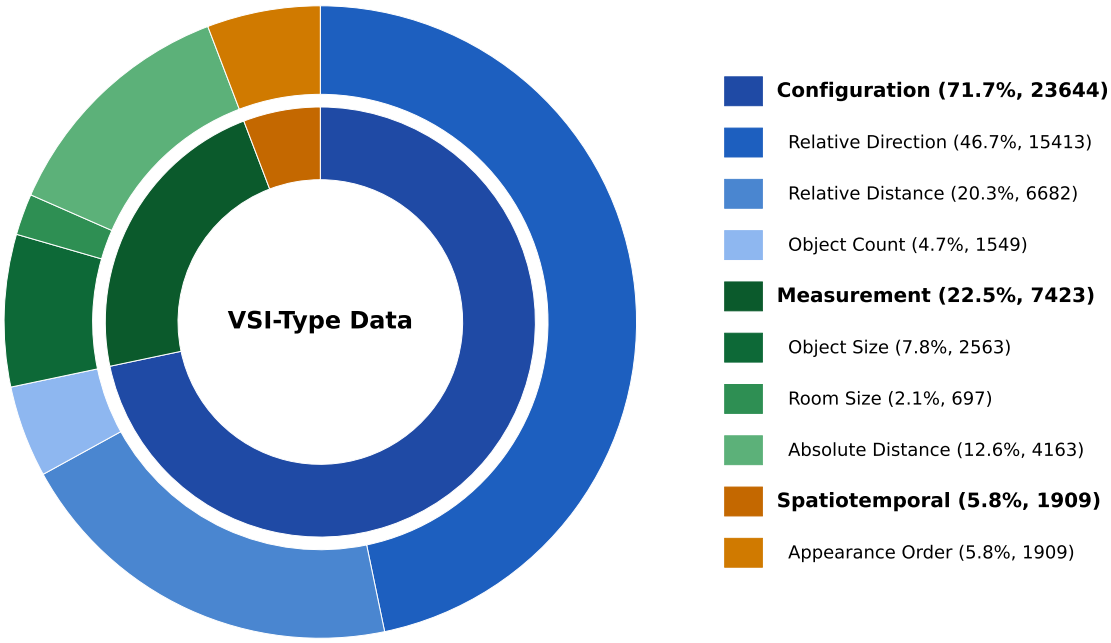

Figure B. **Task-type distribution of the seven tasks in the VSI-Bench setting.** The pie chart summarizes the combined composition from VLM3R-ScanNet and the sampled Appearance-Order subset from VSI-590K, which are merged for unified reporting.

ting, one side is resized to 518 pixels and the other side is scaled proportionally, with center cropping applied when needed. We then apply, when necessary, patch-aligned spatial trimming so that the final height and width satisfy $H \bmod (p \cdot m) = 0$ and $W \bmod (p \cdot m) = 0$, ensuring that the resolution becomes an integer multiple of the effective patch unit $p \cdot m$ (e.g., $14 \cdot 2 = 28$). This alignment is required because the merge stage groups $m \times m$ adjacent patches into a single token.

Finally, the resized image is used to construct inputs for both the vision encoder and the geometry encoder (VGGT), with additional patch/merge alignment applied where needed to maintain spatial consistency between the two branches.

C.2. Training Settings

We train SpatialStack using `torchrun` with DeepSpeed ZeRO-2. Optimization uses AdamW with cosine decay scheduling and warmup. bfloat16 precision is employed for training efficiency and numerical robustness. Tab. B summarizes the configuration.

Category	Setting
Base model	Qwen2.5-VL-3B or Qwen3.5-4B
Geometry encoder	VGGT-1B (frozen)
Fusion strategy	SpatialStack (multi-depth)
Trainable modules	Language tower + fusion modules
Precision	bfloat16
Optimizer	AdamW (wd=0.01)
Learning rate	1×10^{-5}
Scheduler	Cosine decay, warmup 3%
Epochs	1
Batch size	effective 64
Sequence length	12,800 tokens
Frames per video	4–8
Pixels/sample	$16\cdot28^2$–$576\cdot28^2$
Distributed	torchrun + DeepSpeed ZeRO-2
Checkpoint save interval	every 1000 steps
Logging	every 10 steps
Hardware	32×A100 GPUs (80GB)

Table B. **Training hyperparameters for SpatialStack.** Geometry-aware instruction tuning is performed on Qwen2.5-VL-3B or Qwen3.5-4B with VGGT-1B using the proposed SpatialStack fusion. The language tower and fusion modules are trainable, while the geometry encoder remains frozen. Training uses AdamW (bfloat16, cosine schedule) with an effective batch size of 64 under ZeRO-2 parallelism.

D. Evaluation Details

Our evaluation pipeline closely follows established protocols to ensure fair comparison. Specifically, we adopt the data preprocessing methodology from *VG-LLM* [60] and

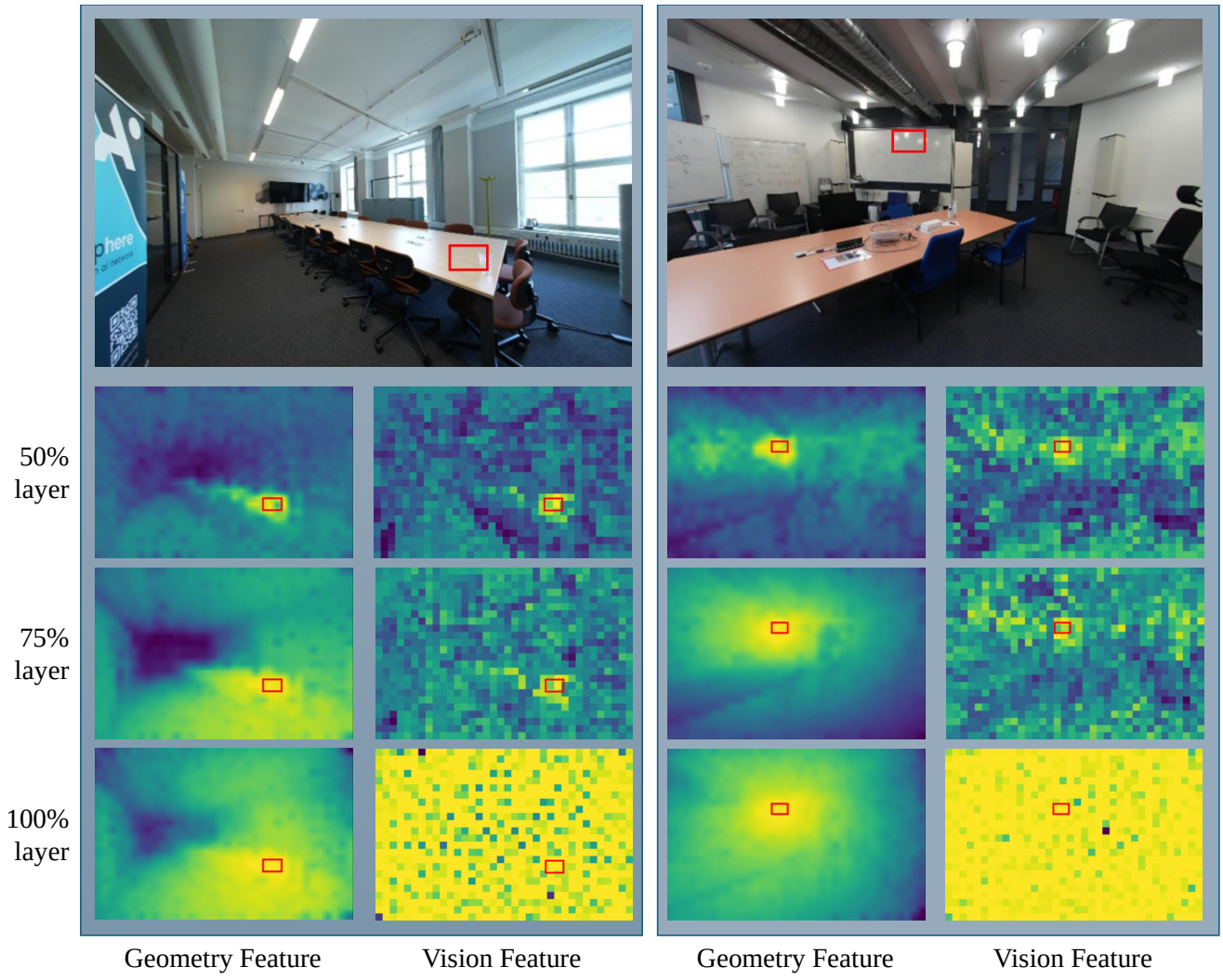

Figure C. **ROI similarity comparison between geometry and vision features across encoder depths.** For two indoor scenes, the top row shows the RGB image with the ROI marked in red. The lower rows display similarity maps (brighter means more similar) at 50%, 75%, and 100% depths of the geometry encoder (left) and the vision encoder (right). Geometry features preserve meaningful spatial structure, while vision features are noisy and become nearly uniform at deeper layers.

adhere to the standard evaluation parameter settings defined in *VSI-Bench* [51].

Implementation Details. Visual inputs (single images, image lists, or videos) are first decomposed into sampled frames with a capped count K, using uniform frame sampling in the evaluation pipeline. Following the preprocessing pipeline of [60], geometry-aware evaluation uses a 518-pixel image preprocessing step. To ensure compatibility with our token merging mechanism, patch/merge alignment is enforced when required so that the patch grid dimensions are divisible by the merge factor m:

$$(W/p) \bmod m = 0 \quad \text{and} \quad (H/p) \bmod m = 0, \tag{17}$$

where p denotes the patch size. When geometry is enabled, the geometry encoder inputs are constructed from the same visual content as the vision branch to maintain spatial correspondence.

Geometry tokens are computed once per sample in evaluation mode. Geometry fusion is injected at predefined decoder layers after self-attention and MLP execution, replacing the vision-aligned slice before decoding continues.

Decoding adopts greedy generation by default (`temperature` $= 0$, `num_beams` $= 1$) with task-specific generation limits unless specified otherwise. Key/value caching is enabled for efficiency, and outputs are trimmed to remove the prompt prefix before evaluation. All benchmark results in the main paper are produced under this evaluation configuration.

E. More Results

We evaluate zero-shot spatial reasoning on CV-Bench in Tab. C. Our method consistently outperforms both SpatialRGPT [8] and Spatialbot [5] across all metrics. Note that SpatialVLM [6] is excluded as its code is unavailable.

Method	Count	Relation	Depth	Distance	Overall
SpatialRGPT	60.4	78.9	80.0	71.3	72.7
Spatialbot	61.4	73.1	76.5	61.0	68.0
Ours	**69.0**	**92.5**	**93.7**	**90.7**	**86.5**

Table C. **Additional Baseline Comparison on CV-Bench.**

F. More Visualizations

F.1. Geometry vs. Vision Feature Responses

To analyze the difference between geometry and vision representations, we visualize ROI-based similarity maps derived from features at different encoder depths, as shown in Fig. C. For each scene, a red box marks a region of interest (ROI) in the RGB image. We compute patch-wise similarity between this ROI and all other spatial locations using features extracted at 50%, 75%, and 100% depth of the geometry encoder and compare them with features from the native vision encoder at corresponding relative depths.

Here, the percentages refer to proportional positions within the encoder stack rather than absolute layer indices. For example, the geometry encoder contains 24 layers, so 50%, 75%, and 100% depths correspond to layers 12, 18, and 24. The vision encoder contains 32 layers, where the same relative depths map to layers 16, 24, and 32. This proportional alignment allows a fair comparison between encoders with different depths.

The similarity maps reveal a consistent trend: shallow geometry layers preserve fine-grained spatial distinctions and clear geometric boundaries, whereas deeper geometry layers become increasingly homogeneous, causing many regions to appear similar despite different physical geometry. In contrast, similarity maps from the native visual encoder are noisy and spatially fragmented across depths, and at the deepest layers they collapse into nearly uniform responses without meaningful spatial differentiation.

These results demonstrate that internal visual features alone lack explicit spatial structure and are insufficient for reasoning about relative geometry. External geometry encoders provide structured spatial cues at different levels that are missing from the native visual pathway, motivating the use of multi-level geometry fusion in spatial reasoning.