GLINT-3D: GEOMETRY-AWARE LANGUAGE-GUIDED INTERLEAVED COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial understanding from multi-view inputs requires vision-language models (VLMs) to preserve both fine-grained visual evidence and geometric relationships that can only be established across views. Feeding per-frame semantic and geometry-aware features directly to the language model is effective but computationally expensive, since the visual token sequence, and with it the decoder KV cache, grows linearly with the number of input frames. In this work we introduce GLINT-3D, a multi-view VLM built on language-guided, interleaved compression of semantic and geometric features. GLINT-3D fuses per-frame features from the VLM vision encoder with multi-view 3D-aware features from a pretrained 3D foundation model, then compresses the result into scene-level and frame-level latent tokens. Rather than compressing once before the language model, GLINT-3D initializes these latent tokens from learnable embeddings and progressively injects visual information through compression layers interleaved with the VLM layers, allowing the question tokens to guide what information is retained. Experiments on a suite of spatial and general visual benchmarks show that one-shot compression causes substantial information loss, while the combination of interleaving and language guidance largely closes the gap, approaching the mean score of the uncompressed model while substantially reducing the visual token count and decoder KV-cache size. GLINT-3D thus offers a favorable trade-off between spatial reasoning performance and scalability to larger image sets and language models.

1 INTRODUCTION

Recent vision-language models (VLMs) have achieved strong performance in visual recognition, captioning, and visual question answering (VQA) but their spatial reasoning capabilities remain substantially weaker. Models that reliably recognize the objects in an image often struggle to estimate metric distances, object sizes, relative orientation, viewpoint changes, or to combine spatial evidence distributed across multiple views. Recent surveys by Lin (2025), Zheng et al. (2025) and Liu et al. (2025a) as well as several benchmarks (Rajabi & Kosecka, 2024; Guo et al., 2024b; Yang et al., 2025a) treat spatial intelligence as a capability distinct from generic visual understanding.

One promising direction is to augment VLMs with representations produced by feed-forward 3D reconstruction models (Wang et al., 2024; Cabon et al., 2025; Keetha et al., 2025; Wang et al., 2025e;d;b). To solve the task of 3D geometry prediction, these models extract strong geometry-aware features directly from multi-view inputs, without requiring explicit depth maps, camera poses, or precomputed 3D scans. These representations provide geometric information that can complement the semantic features of conventional VLM encoders and substantially improve multi-view spatial reasoning.

This type of integration, however, does little to address the scalability issues inherent to per-frame representations. When per-frame semantic and geometry-aware features are passed directly to the language model, the number of visual tokens grows linearly with the number of input views. This rapidly increases attention cost and the decoder KV-cache required to maintain visual context, making the approach impractical for longer sequences and compute-limited settings.

Visual token compression methods such as Perceiver-style fixed-size latent compression (Jaegle et al., 2021; Alayrac et al., 2022), Q-Former (Li et al., 2023), and more recent structured aggregation or token selection (Yang et al., 2026; Li et al., 2024; Shang et al., 2026) promise a potential

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.

solution. Although reasonably effective on general VLM tasks, most of these methods compress information up front, fixing the latent representation before any interaction with the language instruction. The compression must therefore anticipate every possible query, and may consequently discard small objects, local geometric cues, frame-specific evidence, or cross-view relations that later prove essential. This limitation is particularly acute for multi-view spatial reasoning, where questions about the same scene can require very different information: a room-size question depends mainly on global geometry, whereas a relative-direction or appearance-order question may hinge on fine-grained evidence from a few frames.

To address this, we introduce GLINT-3D, a multi-view VLM with query-adaptive visual latent compression. GLINT-3D first merges frame-wise semantic features from a VLM vision encoder with multi-view 3D-aware representations from a pretrained feed-forward 3D reconstruction model such as MapAnything (Keetha et al., 2025). Rather than projecting these features into the LLM space, GLINT-3D carries only a small set of learned latent tokens: a fixed set of scene-level tokens that summarize global scene information, and compressed frame-level tokens that retain view-specific evidence. Crucially, the compressor layers are interleaved with the early layers of the LLM, allowing two-way exchange of information in the latents. This lets the model retrieve evidence relevant to the query at different stages of reasoning, rather than committing to a single compressed representation up front.

Experiments on several spatial intelligence benchmarks confirm this picture. An up-front compressor incurs a substantial loss in spatial reasoning. Interleaving the compressor with the LLM recovers a significant part of this loss, while explicitly conditioning the compression process on the question largely closes the gap to non-compressed methods.

2 Related work

Spatial intelligence of VLMs. As discussed in recent surveys (Lin, 2025; Zheng et al., 2025; Liu et al., 2025a) the role of vision modules has changed in the last few years from isolated perception tasks producing labels, boxes or captions, to multimodal reasoning where the VLM serves as an interface between visual evidence and language-conditioned agents. To enhance the spatial capabilities of VLMs, several strategies were explored.

Early methods focused on textual, visual or hybrid prompting with frozen LLM weights (Wei et al., 2022; Yang et al., 2023; Li et al., 2025), proposed customized training to better capture fine-grained visualspatial information (Wang et al., 2025a; Covert et al., 2025) or injected explicit 2D boxes, masks, or scene graphs into the LLM to improve object disambiguation and reference resolution (Guo et al., 2024a; Zhang et al., 2023; Jain et al., 2024; Liang et al., 2025; Wang et al., 2025c). 2D and 3D features are merged before projecting into LLMs in Huang et al. (2023a); Kang et al. (2025); Yu et al. (2025); Luo et al. (2025); Huang et al. (2023b). More recent methods take only images or a video sequence with poses as input, reconstruct 3D with off-the-shelf SfM methods (Schönberger & Frahm, 2016) and combine 3D with 2D representations (Zhu et al., 2024; Fu et al., 2024; Liu et al., 2025b).

Closer to our setting, where no 3D, camera pose, or depth is available, methods rely on recent feed-forward 3D foundational models (Wang et al., 2024; Cabon et al., 2025; Keetha et al., 2025; Wang et al., 2025e;d;b) to get 3D reconstruction, depth, camera parameters or directly use geometry-aware features from a video or image set. For example, Fan et al. (2026) augment a standard video VLM with instruction-aligned 3D reconstruction, injecting implicit, metric-scale 3D geometry and camera motion tokens into the language model. Wu et al. (2025) combine the visual encoder of Qwen2.5-VL (Bai et al., 2025) with dense 3D VGGT features (Wang et al., 2025b) using a simple connector to integrate the semantic and structural information from both branches. Li et al. (2026a) propose an embodied foundation model trained within a unified spatio-temporal and physically grounded framework, with native 3D grounding supporting spatial reasoning, localization or planning. Zhao et al. (2026) combine semantic and VGGT-based spatial encoders through camera-guided modality fusion, using camera representations to bias, weight, and gate spatial features before they enter the LLM. RoboBrainTeam et al. (2025) pre-process the data to obtain structured scene graphs that are input along with complex task instructions on the language side.

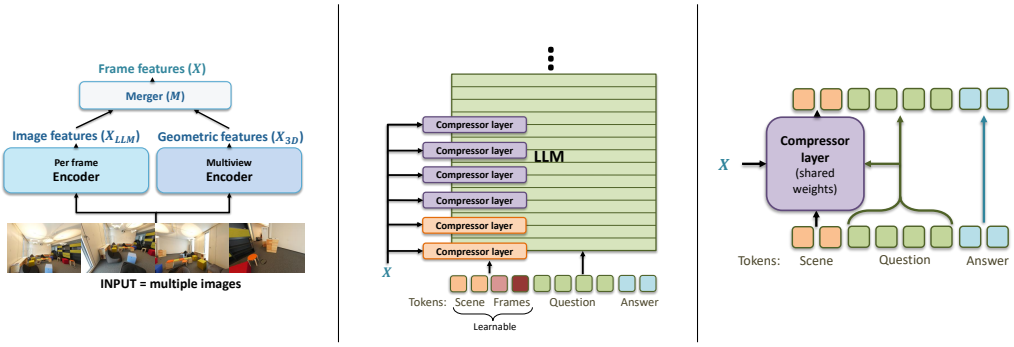


Figure 1: High-level overview of the GLINT-3D architecture. **Left** represents the merger module. **Middle** shows the interleaving of the compressor layers. **Right** represents one of the compressor layers. In practice, 2 sets of learned tokens are used: one per scene and one per frame.

Instead, GLINT-3D focuses on efficiently scaling multi-view reasoning by progressively compressing semantic and geometric frame features through language-guided compressor modules interleaved with the LLM layers, producing compact scene- and frame-level tokens adapted to the input question.

Data also played an important role to improve the spatial reasoning capabilities of LLMs/VLMs through spatially-grounded supervision (Chen et al., 2024a; Deng et al., 2025; Zhang et al., 2025a; Brown et al., 2025a) and iterative data-loops (Liu et al., 2025b). These works establish that targeted spatial data substantially improves spatial competence. In a similar spirit, we also considered data pre-processing and synthetically generating training data. In addition, we design a curriculum learning training procedure where we use dense captioning to learn the alignment between textual and visual/geometric features and progressively include more instruction and question answering examples to adapt with LoRA finetuning of the VLM for such spatial reasoning.

Visual token compression in VLMs. Modern VLMs typically connect a visual encoder to an LLM through an interface that maps hundreds or thousands of patch features into a more compact token sequence. Early approaches use fixed learned-query bottlenecks, such as Perceiver-style resamplers (Jaegle et al., 2021; Alayrac et al., 2022), Q-Former (Li et al., 2023), or lightweight MLP projectors (Liu et al., 2023). More recently, compression is considered an adaptive process rather than a fixed one-shot projection. Hu et al. (2026) propose a tail-token dropping strategy making the query tokens form a Matryoshka structure (Kusupati et al., 2022). TokenPacker (Li et al., 2024) condenses high-resolution features through coarse-to-fine region-to-point aggregation, Shang et al. (2026) select informative visual tokens and merge redundant ones, and Yang et al. (2026) exploit redundancy in vision-encoder representations to retain a compact set of informative tokens. Chen et al. (2024b) propose a versatile plug-and-play method to optimize computational efficiency by learning adaptive attention patterns in early layers and pruning visual tokens in subsequent ones. Despite their effectiveness on general-purpose VLM benchmarks, these approaches largely determine the information exposed to the language model before substantial language reasoning has taken place.

Only a few works have explored adapting the visual token set during language processing. Examples include LVPruning (Sun et al., 2025), which uses interactions between language and visual tokens to progressively prune task-irrelevant visual tokens at intermediate LLM layers, and DUET-VLM (Singh et al., 2026), which combines redundancy-aware visual compression before the LLM with text-guided token dropping throughout the language backbone. QMoP (Li et al., 2026b) instead makes the initial compression query-dependent, dynamically routing visual features through complementary pooling, resampling, and pruning operators according to both the image and textual query. Other methods derive compression decisions from the evolving cross-modal representation itself. Ou et al. (2026) retain visual tokens whose information is not yet explained by the textual subspace, whereas Zhang et al. (2026c) hierarchically distill long visual and textual contexts into compact, layer-specific soft prefixes that are injected throughout the decoder. Unlike these prior methods that progressively discard visual information, our approach preserves dense semantic and geometry-aware features as external memory. Persistent latent tokens can repeatedly retrieve task-relevant evidence at multiple LLM depths, enabling flexible multi-view spatial reasoning across global geometry, view-specific details, and cross-view relationships.

3 METHOD

GLINT-3D, as illustrated in Fig. 1, is a vision-language model designed for multi-view spatial understanding. Given a set of N input images $\{I_1, \ldots, I_N\}$ and a natural-language query Q, it aims to produce an answer conditioned on both the visual appearance and the underlying 3D geometry.

Frame feature merging. We build GLINT-3D upon a base VLM composed of a visual encoder that processes each frame independently and an LLM backbone. Inspired by SpaceMind (Zhao et al., 2026), we enrich the VLM visual representation $\mathbf{X}_{\text{LLM}}$ with multi-layer features $\mathbf{X}_{\text{3D}}$ extracted by a pretrained 3D foundation model. A merger module $\mathcal{M}$ then combines these complementary representations. It takes as its initial input the features $\mathbf{X}_{\text{LLM}}$, which are bilinearly interpolated to match the sequence length of $\mathbf{X}_{\text{3D}}$. The multi-layer features from $\mathbf{X}_{\text{3D}}$ are then incorporated sequentially through AdaLN-Zero (Peebles & Xie, 2023); further implementation details are provided in the Appendix (Fig. 3). We denote the resulting frame-level representation by $\mathbf{X} = \mathcal{M}(\mathbf{X}_{\text{LLM}}, \mathbf{X}_{\text{3D}})$. In our experiments we use Qwen3.5-0.8B (QwenTeam, 2026) as the base VLM, and MapAnything (Keetha et al., 2025) as the geometric feature extractor. Nevertheless, the proposed architecture is model-agnostic and can accommodate other VLMs and 3D foundation models.

Language-guided interleaved compressor.

To alleviate the rapid growth of the input token sequence length with the number of input frames, we adopt a Perceiver-inspired architecture (Jaegle et al., 2021) that compresses the visual context into a compact set of latent tokens. Specifically, we employ two compression modules: a scene-level compressor $\mathbf{P}_{\text{scene}}$, which produces a fixed set of N_{scene} tokens summarizing the complete multi-view scene, and a frame-level compressor $\mathbf{P}_{\text{frame}}$, which produces a fixed set of N_{frame} tokens for each input view. This substantially reduces the visual sequence length and removes redundant information across views.

However, aggressive one-shot compression may discard fine-grained details relevant to answering the query. Furthermore, two full compression modules add a significant number of parameters to the model. We address these limitations through three design choices. First, instead of one-shot compression, we introduce gradual compression by *interleaving* the compressor modules with the LLM layers. Namely, the visual latent tokens are initialized as learnable embeddings and progressively updated by $k = 6$ compressor layers, inserted every $m = 2$ layers of the LLM. This allows visual information to be refined and injected at multiple depths of the language model, while still targeting early injection of spatial information, allowing later layers of the LLM to focus on its interpretation and reasoning.

Second, the compression process is explicitly guided by the input query. Each compressor layer contains an additional cross-attention operation from the visual latent tokens to the question-token representations, *i.e.*the LLM hidden states of the question tokens at the corresponding depth (see the pseudo-code of the compressor layer in Fig. 4 in the Appendix). Consequently, the compressed representation can preserve visual and geometric information that is most relevant to the task at hand, rather than producing a one-size-fits-all summary of the scene. We omit language guidance in the first two compressor layers as the language representations are still very raw and uninformative.

Third, we substantially reduce the number of compressor parameters in two ways. We first tie the weights of the two compressors $\mathbf{P}_{\text{scene}}$ and $\mathbf{P}_{\text{frame}}$. Then, inspired by the success of looping transformers (Dehghani et al., 2019; Yang et al., 2024), we also share weights across different compressor layers: we use one set of weights for all non-language-guided layers and another for all language-guided layers, and add per-depth learnable gates to adapt to the different norms of each layer.

4 DATA MIXTURE AND TRAINING

We train the model in two stages with distinct goals and data mixtures, summarized in Fig. 2. In **Stage 1**, we train the merger and compressor modules, which extract semantic and spatial information from the input frames and inject it into the LLM, while keeping the LLM itself frozen. The goal is to align the extracted scene representations with the LLM latent space. Captioning data is particularly well suited for this task: long captions provide far denser semantic supervision than question-answer pairs targeting a single object or measurement, and a broad, varied captioning mixture lets these modules learn a richer correspondence between scene features and the LLM

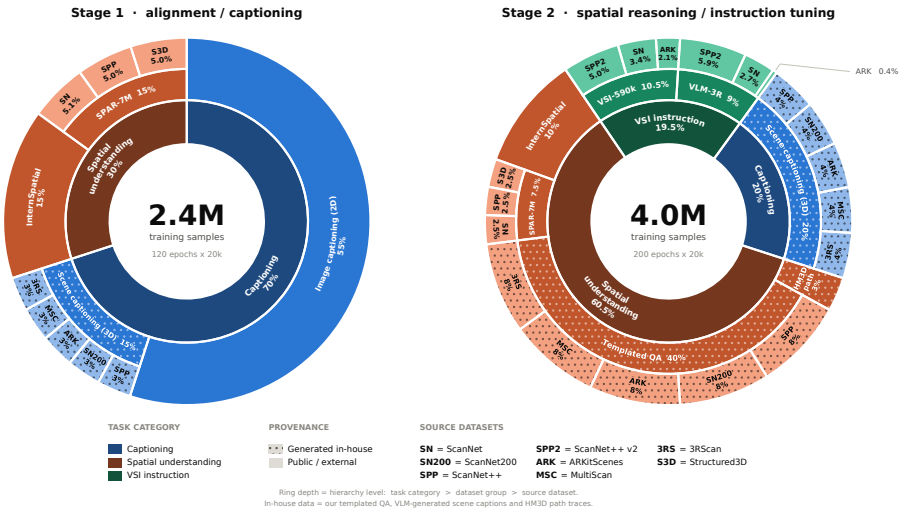


Figure 2: Overview of the data mixture used in each stage of training.

embedding space before the model is exposed to more specialized 3D reasoning tasks. We complement general captioning data with spatial understanding data (30%) from InternSpatial (Deng et al., 2025) and SPAR-7M-RGBD (Zhang et al., 2025a), which adds explicit spatial supervision and, through their grounded inputs (bounding boxes, points), steers feature extraction toward what grounded understanding requires.

In **Stage 2**, we instead target accurate spatial understanding and instruction following. We introduce a broad mixture of task-specific question-answer supervision — spatial understanding (60.5%) and VSI-Bench-style (Yang et al., 2025a) instruction tuning (19.5%) — while retaining multi-view scene captioning (20%) to preserve the alignment established in Stage 1 and to keep the model from over-specializing to short answer formats. The QA data combines open-ended questions with direct answers, including numerical ones, and multiple-choice questions (MCQA), which encourages the model both to produce precise free-form predictions and to discriminate between closely related candidates. Most of the spatial understanding data is in-house templated QA generated from several indoor multi-view 3D datasets. We generate it only from their training scenes; the validation scenes, on which VSI-Bench, VSI-Bench-Debiased and ReVSI are built, are never used for training. We additionally generate a path description dataset from trajectories and scenes in HM3D (Ramakrishnan et al., 2021) to aid route planning.

During training we sample question-answer pairs online from this pre-extracted information, drawing uniformly from nine question types: object presence, object counting, multi-category object counting, object size, distance between objects, relative size of two objects, relative distance of three objects, relative orientation of three objects, and object captioning. Objects are referred to either by their generated referring expression or by class name (only if unique within the scene). We describe each dataset and the corresponding processing steps in detail in the Appendix (Sec. B). We also provide additional implementation details in Sec. A.

5 Experimental Evaluation

Evaluation protocol. For evaluation, we use VLMEvalKit (Duan et al., 2024) and report results on VSI-Bench (Yang et al., 2025a), VSI-Bench-Debiased (Brown et al., 2025b) and ReVSI (Zhang et al., 2026b) using the 32-frame version in each case. In addition, we evaluate on SPAR-Bench (Zhang et al., 2025a), and CV-Bench 2D/3D (Tong et al., 2024) benchmarks. When we evaluate multiple-choice questions we use exact-match accuracy and for the numerical answers we consider mean relative accuracy, following Yang et al. (2025a). In the tables, for our models, the suffix "-nB" refers to the size of the base VLM used.

Table 1: **Ablation results.** All models use Qwen3.5-0.8B as LLM. (a) Component ablations on GLINT-3D v0; Δ is w.r.t. GLINT-3D v0. (b) Changes leading from v0 to the final model, applied cumulatively; Δ is w.r.t. the row above. **First**, second and third best results per column.

Model	VSI-Bench-db	ReVSI	SPAR-Bench	CVB-2D	CVB-3D	Mean	Δ
Qwen3.5-FT	62.1	51.7	62.8	69.1	80.9	65.3	
GLINT-3D w/o compression (=GLINT-3D-NC)	**65.3**	**60.3**	**70.6**	71.9	**83.8**	**70.4**	
GLINT-3D w/ Perceiver compressor	63.7	51.5	58.4	54.5	63.8	58.4	
(a) Ablations on GLINT-3D v0							
GLINT-3D v0	64.5	54.8	66.6	69.5	81.2	67.3	
w/o language guidance	61.9	57.9	61.2	65.7	67.4	62.8	−4.5
w/o language guidance, only $\mathbf{P}_{\text{frame}}$	64.7	53.8	56.6	48.5	58.6	56.4	−10.9
w/o language guidance, only $\mathbf{P}_{\text{scene}}$	62.4	53.3	59.3	67.8	61.4	60.8	−6.5
shared weights between $\mathbf{P}_{\text{frame}}$ and $\mathbf{P}_{\text{scene}}$	65.0	56.3	64.1	67.5	78.5	66.3	−1.0
(b) From v0 to the final model							
GLINT-2D v0	50.5	43.4	57.7	67.8	70.9	58.1	
+ pre-compression layer	50.2	43.5	57.8	69.3	78.3	59.8	+1.7
+ early injection ($m = 2$)	58.1	46.3	57.5	72.5	73.8	61.6	+1.8
+ full weight sharing (= GLINT-2D)	61.4	48.4	60.5	74.3	77.1	64.3	+2.7
GLINT-3D v0	64.5	54.8	66.6	69.5	81.2	67.3	
+ all of the above (= GLINT-3D)	64.6	57.6	65.4	**75.6**	80.5	68.7	+1.4

Table 2: KV-cache size (in MiB) for Qwen3.5 and GLINT-3D w.r.t. input video length (16, 32, 64, 128 frames). Green: reduction factor of GLINT-3D over Qwen3.5.

	Qwen3.5				GLINT-3D			
Model size	**16**	**32**	**64**	**128**	**16**	**32**	**64**	**128**
0.8B or 2B	36	72	144	288	6 ▼6.0×	10 ▼7.2×	17 ▼8.5×	32 ▼9.0×
4B or 9B	96	192	384	768	16 ▼6.0×	26 ▼7.4×	46 ▼8.3×	86 ▼8.9×
27B	192	384	768	1536	33 ▼5.8×	53 ▼7.2×	93 ▼8.3×	173 ▼8.9×

5.1 Ablations and efficiency

References. Table 1 evaluates the components of GLINT-3D against two non-compressed references. Qwen3.5-FT is the naïve fine-tuning of the pure Qwen3.5 architecture, which serves as the uncompressed reference for 2D variants. It has no Stage 1, since there is no merger or compressor to train. GLINT-3D w/o compression projects the merged frame features directly into the LLM and serves as the uncompressed reference for 3D variants. It is shortened to GLINT-3D-NC in most tables. For this baseline we resize the $\mathbf{X}_{\text{3D}}$ features to the sequence length of $\mathbf{X}_{\text{LLM}}$, so its KV cache has the same length as that of Qwen3.5.

We first compare our approach with an up-front Perceiver-style compression. We keep the same two-Perceiver architecture (per-scene and per-frame, 5 layers each, applied to the merged features before they enter the LLM). This scores 58.4, which is 12.0 below GLINT-3D w/o compression and 6.9 below Qwen3.5-FT, showing that too much information is lost in the compressors.

Ablations on v0. We run a first round of ablations on an early version of GLINT-3D, which we call v0. In this early version we used a stride of $m = 4$ to match the full-attention blocks of Qwen3.5. It also had no compressor block before the LLM, no weight sharing, and language guidance applied in every compressor layer. Assuming that conclusions transfer to the final model, Table 1(a) ablates this model; re-doing all the ablations with the latest model would be too costly.

Without language guidance, interleaving alone reaches 62.8 (+4.4 over the up-front Perceiver), but remains below the 2D-only reference. Language guidance adds +4.5: v0 reaches 67.3, +2.0 above Qwen3.5-FT and 3.1 below the uncompressed model. Both interleaving and language guidance are essential for closing the gap. Additionally, the two compressors are complementary: starting from the model without language guidance, keeping only $\mathbf{P}_{\text{scene}}$ costs 2.0 points and keeping only $\mathbf{P}_{\text{frame}}$ costs 6.4 (the Δ column in Tab. 1 is instead w.r.t. v0). Finally, to reduce the number of new parameters, we shared the weights horizontally, between $\mathbf{P}_{\text{frame}}$ and $\mathbf{P}_{\text{scene}}$ while keeping independent weights per layer. In the v0 model, this did not work and cost 1.0 point.

From v0 to the final model. We then applied the same v0 recipe to GLINT-2D, which compresses the Qwen vision features without MapAnything and is the compressed counterpart of Qwen3.5-

Table 3: Evaluation on the original VSI-Bench (32 frames).

Model	Numerical Question				Multiple-Choice Question				Avg.
	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	App. Order	
Gemini-2.5-Pro	43.8	34.9	64.3	43.8	61.1	47.8	45.9	71.3	51.5
InternVL3-78B	71.2	53.7	44.4	39.5	55.9	39.5	28.9	54.5	48.4
GPT-5 (2025-08-07)	53.5	47.0	73.3	47.5	63.7	48.6	**50.2**	68.9	56.6
Qwen3-VL-8B-Instruct	53.3	47.0	76.3	61.9	58.0	50.9	35.0	66.3	56.1
GeoSR-7B	68.3	38.7	57.4	62.3	48.7	44.4	35.6	59.5	51.9
VLM3R-7B	70.2	49.4	69.2	67.1	65.4	80.5	45.4	40.1	60.9
SpaceMind-8B	73.3	61.4	77.3	74.2	67.2	88.4	44.3	70.6	69.6
SSR-3D-7B	70.5	67.1	76.0	79.5	71.3	**93.4**	48.5	**85.0**	73.9
RynnBrain 1.1-9B	–	–	–	–	–	–	–	–	74.9
RynnBrain 1.1 122B-A10B	–	–	–	–	–	–	–	–	**75.0**
Qwen3.5-0.8B-FT	73.1	50.9	75.7	69.4	69.2	78.4	43.8	81.4	67.7
GLINT-2D-0.8B	71.3	50.0	75.4	70.3	68.2	76.0	43.3	79.8	66.8
GLINT-3D-NC-0.8B	72.6	59.7	76.9	72.6	70.4	86.6	49.0	79.4	70.9
GLINT-3D-0.8B	72.7	59.7	76.5	73.1	68.3	85.8	44.3	78.0	69.8
GLINT-3D-NC-4B	**75.9**	**67.2**	**80.2**	**80.7**	**76.6**	88.0	46.9	83.7	74.9

Table 4: Evaluation on VSI-Bench-Debiased (32 frames).

Model	Numerical Question				Multiple-Choice Question				Avg.
	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	App. Order	
Gemini-2.5-Pro	40.8	34.4	56.7	40.9	61.0	46.8	43.0	69.5	49.1
SSR-3D-7B	–	–	–	–	–	–	–	–	69.9
Qwen3.5-0.8B-FT	60.3	49.1	61.5	69.1	64.8	68.3	43.0	80.8	62.1
GLINT-2D-0.8B	59.4	48.3	61.6	69.5	64.5	68.8	40.4	78.6	61.4
GLINT-3D-NC-0.8B	63.4	56.9	63.8	71.3	67.7	79.8	**44.7**	74.8	65.3
GLINT-3D-0.8B	62.2	57.5	63.5	71.7	64.8	80.8	41.2	75.2	64.6
GLINT-3D-NC-4B	**67.2**	**65.1**	**70.1**	**81.1**	**74.8**	**83.4**	42.1	**81.4**	**70.7**

FT. As Table 1(b) shows, v0 transfers poorly: it scores 58.1, 7.2 below Qwen3.5-FT, with the largest drops on VSI-Bench-db (-11.6), CVB-3D (-10.0) and ReVSI (-8.3). Adding an additional (pre-)compressor block before the LLM helps a little ($+1.7$).

Following prior findings that multi-modal integration happens mostly in the early LLM layers (Yin et al., 2025; Zhang et al., 2025b), we also decided to halve the stride to $m = 2$, to focus injection on early layers. At the same time we disable language guidance in the first two compressor layers, where the language representations are still raw. This early injection approach adds another $+1.8$. Together these two changes bring GLINT-2D to 61.6, $+3.5$ over v0 and 3.7 below Qwen3.5-FT.

The main drawback of the method at this point was the number of parameters in the compressors (33M parameters per layer), which took up much of the memory during training. We therefore shared the weights not only between $\mathbf{P}_{\text{frame}}$ and $\mathbf{P}_{\text{scene}}$, as before, but also across layers. The compressor blocks then have only two distinct sets of weights, one without and one with language guidance. We expected a small loss, as with the shared $\mathbf{P}_{\text{frame}}/\mathbf{P}_{\text{scene}}$ weights in v0. Instead, full weight sharing gains $+2.7$, leaving GLINT-2D only 1.0 below Qwen3.5-FT. We attribute this to reduced per-depth overfitting: tying the weights reaches the same training loss as per-depth weights, but the validation loss of the untied model flattens late in Stage 2, while that of the tied model keeps improving.

Applying all of these changes to GLINT-3D gives a smaller gain of $+1.4$. The resulting model beats Qwen3.5-FT by $+3.4$ and is only 1.7 below GLINT-3D w/o compression. We use this final configuration for GLINT-3D and GLINT-2D in the rest of the paper.

Efficiency. The main benefit of GLINT-3D is its shorter sequence length, which directly translates into a smaller KV-cache size. This advantage becomes particularly important for larger models and long image sequences, as shown in Table 2. Nevertheless, the GLINT-3D geometric branch, merger and compressors all introduce additional computational complexity. In terms of trainable parameters, our merger adds about 50M and 150M parameters in the 0.8B and 4B models respectively, while the compressors add about 65M and 400M parameters.

Training. During Stage 2 training, we use the same configuration: 0.8B base VLM, 4 A100 GPUs, a batch size of 8 per GPU, and 4 gradient accumulation steps. In this configuration, the wall-clock training time for GLINT-2D is 31 hours, almost $2\times$ faster than Qwen3.5-FT at 61 hours. They

Table 5: Evaluation on ReVSI (32 frames).

Model	Numerical Question				Multiple-Choice Question			Avg.
	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	
Chance (Frequency)	51.1	40.3	17.1	20.9	26.2	32.7	30.2	31.2
Qwen3-VL-32B-Instruct	42.2	58.3	68.5	58.6	47.9	36.6	46.5	51.2
InternVL3.5-38B	43.4	56.5	70.0	56.9	55.8	47.8	41.5	53.1
VLM3R-7B	41.6	61.6	64.8	52.5	46.5	49.5	34.1	50.1
Qwen3.5-0.8B-FT	50.3	55.3	66.9	50.4	50.0	59.1	30.2	51.7
GLINT-2D-0.8B	46.4	54.7	65.5	53.1	41.5	44.4	33.3	48.4
GLINT-3D-NC-0.8B	56.4	72.1	68.4	53.7	55.4	**80.5**	35.7	60.3
GLINT-3D-0.8B	52.0	69.0	68.0	55.8	56.4	65.1	37.2	57.6
GLINT-3D-NC-4B	**58.4**	**78.4**	**73.5**	**61.4**	**74.9**	57.7	**47.7**	**64.6**

Table 6: Evaluation on SPAR-Bench.

Model	Avg.	Low	Depth-OC	Depth-OC-MV	Depth-OO	Depth-OO-MV	Dist-OC	Dist-OC-MV	Dist-OO	Dist-OO-MV	Medium	RoomMatch	CamMotion	ViewChange	High	Dist-OO	Dist-OO-MV	ObjRel-OC-MV	ObjRel-OO	ObjRel-OO-MV	Spfmap-OC	Spfmap-OC-MV	Spfmap-OO	Spfmap-OO-MV
Chance Level (Frequency)	32.7	31.2	43.1	43.5	17.4	13.1	41.9	31.0	27.4	32.2	38.2	29.0	26.8	59.0	32.3	52.9	50.6	28.2	26.9	26.6	26.3	26.7	26.5	25.8
GPT-4.1	41.6	42.0	48.2	42.7	20.9	23.6	58.3	52.3	48.6	40.4	36.7	59.3	28.8	22.0	42.9	71.8	67.3	46.2	55.0	41.0	30.4	29.6	20.2	24.9
InternVL2.5-8B+SPAR-mix	63.2	**65.5**	81.5	79.2	38.2	**35.5**	78.9	79.2	68.1	**63.5**	63.0	**78.9**	73.0	37.1	61.3	86.5	79.8	64.0	69.2	59.0	47.3	50.0	42.4	53.5
Qwen3.5-0.8B-FT	62.8	54.7	79.7	70.6	19.7	17.0	74.7	73.7	57.9	44.4	40.9	31.8	69.0	22.0	77.3	85.0	79.5	87.5	86.0	81.7	72.6	69.5	57.0	77.0
GLINT-2D-0.8B	60.5	54.3	78.0	70.3	21.0	19.2	73.1	70.8	53.3	48.6	38.5	42.2	51.8	21.5	73.4	82.6	72.0	84.8	80.8	78.4	66.7	64.2	55.3	76.2
GLINT-3D-NC-0.8B	70.6	62.5	84.3	80.1	40.7	22.8	81.0	81.6	65.2	43.8	60.6	37.7	**89.8**	54.5	81.2	87.6	87.2	92.5	86.0	88.9	**77.2**	75.0	55.0	81.5
GLINT-3D-0.8B	65.4	58.8	81.9	81.0	21.0	17.9	80.2	80.9	64.8	42.8	40.2	38.2	55.0	27.6	79.5	86.5	83.6	90.2	83.2	87.3	76.3	**75.6**	51.7	81.5
GLINT-3D-NC-4B	**73.4**	65.2	**87.2**	**82.9**	**41.3**	24.1	**83.4**	**85.0**	**71.7**	46.0	**67.5**	40.5	89.5	**72.5**	**82.6**	**90.6**	**88.4**	**93.8**	**89.0**	**90.3**	72.8	71.5	**63.2**	**84.0**

peak at 11 and 12 GB of GPU memory respectively. The picture is less favorable for the geometric variants: because GLINT-3D's compression branches keep MapAnything's full sequence length, its peak GPU memory is more than twice that of GLINT-3D-NC (70 *versus* 31 GB), although its wall-clock time remains lower (120 *versus* 155 hours). In both of these, the frozen 1.2B-parameter MapAnything clearly dominates the forward pass.

Inference. On one A100 GPU (0.8B base VLM, bf16, greedy decoding, a VSI-Bench scene resampled to 8–128 views), compression shortens the LLM input from 1.7k–25k to 0.5k–3.1k tokens. The LLM prefill, compressors included, stays at 60–95 ms, while it grows linearly without compression and is 4–5× slower at 128 views; the compressors break even between 16 and 32 views. The end-to-end gain depends on the encoder: GLINT-2D cuts the time to first token by 47% at 128 views, whereas MapAnything takes 90% or more of GLINT-3D's time to first token and limits its gain to 8%. Decoding speed (36–41 ms per token) and peak memory are the same for all variants: at 0.8B, the KV cache (Table 2) is small next to the encoder activations at this size.

5.2 Benchmark results compared to SOTA

We provide below detailed per-benchmark comparisons with SOTA methods in Tables 3 to 7.

VSI-Bench (Yang et al., 2025a). As shown in Table 3, GLINT-3D-0.8B achieves an average score of 69.8, only 1.1 below GLINT-3D-NC-0.8B. It outperforms the Qwen3.5-0.8B-FT baseline by 2.1, performs on par with SpaceMind-8B (Zhao et al., 2026) and clearly outperforms recent methods such as GeoSR-7B (Zhang et al., 2026a) and VLM3R-7B (Fan et al., 2026) despite using a substantially smaller language model. Both versions of GLINT-3D remain behind the SOTA methods on this dataset, which are SSR-3D-7B (Liu et al., 2025b) and RynnBrain (Li et al., 2026a). We attribute this difference to the very small size of our base LLM. When we train GLINT-3D-NC[1] with the 4B base VLM and slightly more data, we obtain 74.9, which is comparable with the best results.

VSI-Bench-Debiased (Brown et al., 2025b). The same overall trend holds on VSI-Bench-Debiased, as reported in Table 4. Both GLINT-2D and GLINT-3D remain very competitive with their uncompressed counterparts, limiting the drop to 0.7 in both cases, while GLINT-3D outperforms both GLINT-2D and Qwen3.5-0.8B-FT by 2–3 points. SSR-3D-7B (Liu et al., 2025b) is once again above GLINT-3D, while GLINT-3D-NC-4B is slightly better, with an average score of 70.7.

[1]Due to a technical issue, the training of the GLINT-3D-4B did not finish before the deadline.

Table 7: Evaluation on CV-Bench.

Model	CV-Bench-2D			CV-Bench-3D			Average
	Overall	Count	Relation	Overall	Depth	Distance	
Cambrian-1-8B	72.3	-	-	72.0	-	-	72.2
RoboBrain-7B-2.0	-	-	-	-	-	-	85.8
Qwen3.5-0.8B-FT	69.1	48.7	89.5	80.9	85.5	76.3	75.0
GLINT-2D-0.8B	74.3	61.2	87.4	77.1	79.5	74.7	75.7
GLINT-3D-NC-0.8B	71.9	57.6	86.2	83.8	87.2	80.3	77.8
GLINT-3D-0.8B	75.6	62.1	89.2	80.5	84.5	76.5	78.1
GLINT-3D-NC-4B	**81.1**	**72.0**	**90.2**	**94.0**	**96.8**	**91.2**	**87.5**

ReVSI (Zhang et al., 2026b). On ReVSI, shown in Tab. 5, the drop on the compressed models GLINT-2D and GLINT-3D is slightly higher than on VSI-Bench-Debiased, 3.3 and 2.7 respectively, due in particular to a very strong drop on the relative-direction (-15.4 in case of GLINT-3D)—an order of magnitude larger than on this same task shown on VSI-Bench and VSI-Bench-Debiased. After a careful analysis, we observed that the drop comes from backward questions, *e.g.* "*facing in the opposite direction of...*", that are absent from our VSI-style training data and therefore systematically inverted by some models. Furthermore, we can also observe the gain of using 3D foundation model features, yielding about +9 point gains for both non-compressed and compressed versions. GLINT-3D-0.8B outperforms all previously reported results for this benchmark.

SPAR-Bench (Zhang et al., 2025a). Results on SPAR-Bench are presented in Table 6. GLINT-3D achieves an average score of 65.4, exceeding both Qwen3.5-0.8B-FT (62.8) and InternVL2.5-8B trained on SPAR-mix (63.2). The improvement is particularly pronounced for the high-level reasoning category, where GLINT-3D reaches 79.5 compared with 61.3 for InternVL2.5-8B, a gain of +18.2. It performs strongly across object-relation and spatial-imagination tasks, obtaining scores above 83 on several subsets. Compression seems to negatively affect medium-level tasks such as camera-motion and view change performance where the uncompressed variants GLINT-3D-NC, especially the large 4B model keeping all tokens, seem to handle these tasks much better.

CV-Bench (Tong et al., 2024). GLINT-3D achieves scores of 75.6 on CV-Bench-2D and 80.5 on CV-Bench-3D (see Table 7). Compared with the Qwen3.5-FT baseline, it improves the overall CVB-2D score by +6.5 points, primarily through a substantial gain in counting accuracy (+13.4), and it performs on par on CVB-3D. GLINT-3D also surpasses its uncompressed 0.8B counterpart on both 2D tasks, improving the overall score from 71.9 to 75.6. On CV-Bench-3D, however, it falls below the uncompressed variant (83.8), indicating that compression is less effective at preserving the evidence required for depth and distance judgments. The uncompressed 4B model achieves the strongest overall results, reaching 81.1 and 94.0 on CVB-2D and CVB-3D.

6 Conclusion

We introduced GLINT-3D, a multi-view VLM for efficient 3D spatial reasoning that combines semantic features with geometry-aware representations without requiring depth, camera poses, or explicit 3D reconstruction. It compresses dense multi-view features into compact scene- and frame-level visual latents that are progressively updated throughout the LLM and conditioned on the input question. Experiments across five benchmarks show that incorporating 3D features consistently improves spatial reasoning, while interleaved and language-guided compression recovers much of the performance of the uncompressed model with a substantially shorter visual context and reduced KV-cache requirements. Overall, GLINT-3D provides a favorable trade-off between spatial reasoning performance and computational efficiency, and scales better to reasoning over large image sets.

Limitations.
- *Multi-turn conversation*. GLINT-3D compression is adapted to a single input query. In a multi-turn setting, on the same image set, the compressors must be re-run for each incoming query.
- *Reasoning*. Due to the limited scale, our experiments do not make use of internal reasoning (chain-of-thought), a key ingredient of modern LLMs. Reasoning could potentially provide additional guidance to the compression mechanism. How to incorporate our compression mechanism in such settings remains an open question.

AI USE STATEMENT

We used generative AI tools to polish the language and readability of the manuscript, and to improve model and training code, which the authors fully understood, reviewed, and tested. We also used generative AI to support data annotation, specifically scene and object captioning, as described in detail in Section B. Finally, we used AI-assisted search tools to identify potentially relevant gaps in related work and ensure that important references were not overlooked; all works obtained with such methods were manually reviewed and cited. All AI-assisted outputs have been reviewed and validated by the authors, who take full responsibility for the final content of the paper including its writing, code, annotations, figures, and scientific claims.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a Visual Language Model for Few-Shot Learning. In *NeurIPS*, 2022. 1, 3

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, et al. Qwen2.5-VL Technical Report. arXiv:2502.1392, 2025. 2

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes: A Diverse Real-World Dataset for 3D Indoor Scene Understanding Using Mobile RGB-D Data. In *NeurIPS*, 2021. 15

Ellis Brown, Arijit Ray, Ranjay Krishna, Ross Girshick, Rob Fergus, and Saining Xie. SIMS-V: Simulated Instruction-Tuning for Spatial Video Understanding. arXiv:2511.04668, 2025a. 3

Ellis Brown, Jihan Yang, Shusheng Yang, Rob Fergus, and Saining Xie. Benchmark Designers Should "Train on the Test Set" to Expose Exploitable Non-Visual Shortcuts. arXiv:2511.04655, 2025b. 5, 8

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jérôme Revaud, and Vincent Leroy. MUSt3R: Multi-view Network for Stereo 3D Reconstruction. In *CVPR*, 2025. 1, 2

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*, 2024a. 3

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An Image Is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models. In *ECCV*, 2024b. 3

Ian Connick Covert, Tony Sun, James Zou, and Tatsunori Hashimoto. Locality Alignment Improves Vision-Language Models. In *ICLR*, 2025. 2

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. 15

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal Transformers. In *ICLR*, 2019. 4

Nianchen Deng, Lixin Gu, Shenglong Ye, Yinan He, Zhe Chen, Songze Li, Haomin Wang, Xingguang Wei, Tianshuo Yang, Min Dou, Tong He, Wenqi Shao, Kaipeng Zhang, Yi Wang, Botian Shi, Yanting Zhang, Jifeng Dai, Yu Qiao, Hongjie Zhang, and Wenhai Wang. InternSpatial: a Comprehensive Dataset for Spatial Reasoning in Vision-Language Models. arXiv:2506.18385, 2025. 3, 5

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. VLMEvalKit: An Open-Source Toolkit for Evaluating Large Multi-Modality Models. In *ACM MM*, 2024. 5

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, et al. VLM-3R: Vision-Language Models Augmented with Instruction-Aligned 3D Reconstruction. In *CVPR*, 2026. 2, 8, 15

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-LLM: Extending Language Model for 3D Visual Understanding and Reasoning. arXiv:2403.11401, 2024. 2

Qiushan Guo, Shalini De Mello, Hongxu Yin, Wonmin Byeon, Ka Chun Cheung, Yizhou Yu, Ping Luo, and Sifei Liu. RegionGPT: Towards Region Understanding Vision Language Model. In *CVPR*, 2024a. 2

Xuefeng Guo, Rui Zhang, Yifei Duan, Yuxuan He, Chao Zhang, Shuo Liu, and Long Chen. SURDS: Benchmarking Spatial Understanding and Reasoning in Driving Scenarios with Vision Language Models. arXiv:2411.13112, 2024b. 1

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 *ICLR*, 2022. 14

Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka Query Transformer for Large Vision-Language Models. In *NeurIPS*, 2026. 3

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-Scene: Bridging 3D Scene and Large Language Models with Object Identifiers. arXiv:2312.08168, 2023a. 2

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:2311.12871, 2023b. 2

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General Perception with Iterative Attention. arXiv:2103.03206, 2021. 1, 3, 4

Jitesh Jain, Jianwei Yang, and Humphrey Shi. VCoder: Versatile Vision Encoders for Multimodal Large Language Models. In *CVPR*, 2024. 2

Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. SceneVerse: Scaling 3D Vision-Language Learning for Grounded Scene Understanding. In *ECCV*, 2024. 15

Weitai Kang, Haifeng Huang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Robin3D: Improving 3D Large Language Model via Robust Instruction Tuning. arXiv:2410.00255, 2025. 2

Nikhil Keetha, Norman Mller, Johannes Schnberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bul, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal Feed-Forward Metric 3D Reconstruction. arXiv:2509.13414, 2025. 1, 2, 4, 14

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka Representation Learning. In *NeurIPS*, 2022. 3

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 *ICML*, 2023. 1, 3

Kehan Li, Bohan Hou, Minghao Zhu, Tianyi Zhang, Zesen Cheng, Zhikai Wang, Sicong Leng, Xin Li, Xiao Lin, Biying Yao, et al. RynnBrain 1.1: Towards More Capable and Generalizable Embodied Foundation Model. arXiv:2607.17977, 2026a. 2, 8

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and Ground for Zero-Shot Openvocabulary 3D Visual Grounding. In *CVPR*, 2025. 2

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. TokenPacker: Efficient Visual Projector for Multimodal LLM. arXiv:2407.02392, 2024. 1, 3

Zhongyang Li, Yaqian Li, Faming Fang, Rinyoichi Takezoe, Zi-Hao Bo, Cheng Qian, Mo Guang, Guixu Zhang, and Kaiwen Long. QMoP: Query Guided Mixture-of-Projector for Efficient Visual Token Compression. arXiv:2603.21232, 2026b. 3

Dayong Liang, Changmeng Zheng, Zhiyuan Wen, Yi Cai, Xiao-Yong Wei, and Qing Li. Seeing Beyond the Scene: Enhancing Vision-Language Models with Interactional Reasoning. arXiv:2505.09118, 2025. 2

Fengming Lin. Vision Language Models: a Survey of 26K Papers. arXiv:2510.09586, 2025. 1, 2

Disheng Liu, Tuo Liang, Zhe Hu, Jierui Peng, Yi Xu, Yun Fu, and Yu Yin. Spatial Intelligence in Vision-Language Models: A Comprehensive Survey. techrxiv.176231405.57942913, 2025a. 1, 2

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023. 3

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. In *NeurIPS*, 2025b. 2, 3, 8

Jingzhou Luo, Yang Liu, Weixing Chen, Zhen Li, Yaowei Wang, Guanbin Li, and Liang Lin. DSPNet: Dual-Vision Scene Perception for Robust 3D Question Answering. arXiv:2503.03190, 2025. 2

Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel X. Chang, and Manolis Savva. MultiScan: Scalable RGBD Scanning for 3D Environments with Articulated Object. In *NeurIPS*, 2022. 15

Congyang Ou, Ruike Song, Yang Zhou, Libo Sun, Haokui Zhang, and Zhenbo Luo. When Vision Becomes Text: Visual Token Pruning via Cross-Modal Residual Guidance in VLMs. arXiv:2608.10489, 2026. 3

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 4

QwenTeam. Qwen3.5: Towards Native Multimodal Agents. https://qwen.ai/blog?id=qwen3.5, 2026. 4, 14, 15

Navid Rajabi and Jana Kosecka. GSR-Bench: a Benchmark for Grounded Spatial Reasoning Evaluation via Multimodal LLMs. arXiv:2406.13246, 2024. 1

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI. In *NeurIPS*, 2021. 5

RoboBrainTeam, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, Yi Han, Yingbo Tang, Xiangqi Xu, Wei Guo, Yaoxu Lyu, Yijie Xu, et al. RoboBrain 2.0 Technical Report. arXiv:2507.02029, 2025. 2

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion Revisited. In *CVPR*, 2016. 2

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. arXiv:2403.15388, 2026. 1, 3

Aditya Kumar Singh, Hitesh Kandala, Pratik Prabhanjan Brahma, Zicheng Liu, and Emad Barsoum. DUET-VLM: Dual Stage Unified Efficient Token Reduction for VLM Training and Inference. arXiv:2602.18846, 2026. 3

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Batista-Navarro. LVPruning: An Effective Yet Simple Language-Guided Vision Token Pruning Approach for Multi-modal Large Language Models. arXiv:2501.13652, 2025. 3

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. In *NeurIPS*, 2024. 5, 9

Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In *ICCV*, 2019. 15

Haochen Wang, Anlin Zheng, Yucheng Zhao, Tiancai Wang, Zheng Ge, Xiangyu Zhang, and Zhaoxiang Zhang. Reconstructive Visual Instruction Tuning. arXiv:2410.09575, 2025a. 2

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *CVPR*, 2025b. 1, 2

Jingyi Wang, Jianzhong Ju, Jian Luan, and Zhidong Deng. LLaVa-SG: Leveraging Scene Graphs as Visual Semantic Expression in Vision-Language Models. In *ICASSP*, 2025c. 2

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3D Perception Model with Persistent State. arXiv:2501.12387, 2025d. 1, 2

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *CVPR*, 2024. 1, 2

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π3: Permutation-Equivariant Visual Geometry Learning. arXiv:2507.13347, 2025e. 1, 2

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-Of-thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*, 2022. 2

Luis Wiedmann, Orr Zohar, Amir Mahla, Xiaohan Wang, Rui Li, Thibaud Frere, Leandro von Werra, Aritra Roy Gosthipaty, and Andres Marafioti. FineVision: Open Data Is All You Need. arXiv:2510.17269, 2025. 15

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-MLLM: Boosting MLLM Capabilities in Visual-Based Spatial Intelligence. arXiv:2505.23747, 2025. 2

Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. In *CVPR*, 2025a. 1, 5, 8, 15

Lingfeng Yang, Yueze Wang, Xiang Li, Xinlong Wang, and Jian Yang. Fine-Grained Visual Prompting. In *NeurIPS*, 2023. 2

Liu Yang, Kangwook Lee, Robert D Nowak, and Dimitris Papailiopoulos. Looped Transformers are Better at Learning Learning Algorithm. In *ICLR*, 2024. 4

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer Is Better But Not Necessary in Vision Language Models. arXiv:2412.04467, 2026. 1, 3

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. arXiv:2511.04670, 2025b. 15

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes. In *ICCV*, 2023. 15

Hao Yin, Guangzong Si, and Zilei Wang. Lifting the Veil on Visual Information Flow in MLLMs: Unlocking Pathways to Faster Inference. In *CVPR*, 2025. 7

Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. Inst3D-LMM: Instance-Aware 3D Scene Understanding with Multi-Modal Instruction Tuning. arXiv:2503.08613, 2025. 2

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. arXiv:2503.22976, 2025a. 3, 5, 9, 15

Shihua Zhang, Qiuhong Shen, Shizun Wang, Tianbo Pan, and Xinchao Wang. Make Geometry Matter for Spatial Reasoning. In *ECCV*, 2026a. 8

Shilong Zhang, Peize Sun, Shoufa Chen, Minn Xiao, Wenqi Shao, Wenwei Zhang, Yu Liu, Kai Chen, and Ping Luo. GPT4Roi: Instruction Tuning Large Language Model on Region-of-Interest. arXiv:2307.03601, 2023. 2

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. arXiv:2604.24300, 2026b. 5, 9

Yuqi Zhang, Cheng Chen, Yuyu Guo, Wenjie Yang, Lingchen Meng, Peng Di, Hang Yu, Zuxuan Wu, and Yu-Gang Jiang. VLZip: Unified Visual and Textual Compression for Interleaved Long-Context Modeling. arXiv:2608.08630, 2026c. 3

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-Modal Information Flow in Multimodal Large Language Models. In *CVPR*, 2025b. 7

Ruosen Zhao, Zhikang Zhang, Jialei Xu, Jiahao Chang, Dong Chen, Lingyun Li, Weijian Sun, and Zizhuang Wei. SpaceMind: Camera-Guided Modality Fusion for Spatial Reasoning in Vision-Language Models. In *CVPR*, 2026. 2, 4, 8

Xu Zheng, Zihao Dongfang, Lutao Jiang, Boyuan Zheng, Yulong Guo, Zhenquan Zhang, Giuliano Albanese, Runyi Yang, Mengjiao Ma, Zixin Zhang, Chenfei Liao, Dingcheng Zhen, Yuanhuiyi Lyu, Yuqian Fu, Bin Ren, Linfeng Zhang, Danda Paudel, Nicu Sebe, Luc Van Gool, and Xuming Hu. Multimodal Spatial Reasoning in the Large Model Era: a Survey and Benchmarks. arXiv:2510.25760, 2025. 1, 2

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. LlaVa-3D: a Simple Yet Effective Pathway to Empowering LMMs with 3D-Awareness. arXiv:2409.18125, 2024. 2

Appendix

A Implementation details

Unless otherwise specified, we use Qwen3.5-0.8B (QwenTeam, 2026) as the base VLM and MapAnything (Keetha et al., 2025) to extract geometric features. For Qwen3.5, we set the temporal patch size to 1 and resize the input images to 512 pixels, using a patch size of 16. For MapAnything, we resize the images to 518 pixels and use a patch size of 14.

The multi-view features $\mathbf{X}_{\text{3D}}$ are a list containing the outputs from several layers of MapAnything, namely the output of MapAnything's encoder and the outputs of layers 7, 11, and 14 and of the final layer of its decoder, together with the scale tokens expanded to match the sequence length. The scene compressor uses $N_{\text{scene}} = 200$, whereas the frame compressor uses $N_{\text{frame}} = 20$. Both compressors consist of $k = 6$ and $k = 8$ layers for the 0.8B and 4B base VLMs, respectively, starting right before the LLM and integrated after every $m = 2$ intermediate LLM layers. In all cases, the first two compressor layers, for both compressor types, are without language guidance; the remaining layers use language guidance. Layers are grouped by their type (with/without language guidance), and each group shares the same set of weights.

The pseudo-code of the feature merging module is provided in Figure 3 and for the language-guided, interleaved scene compressor layer in Fig. 4.

We train the 0.8B model on 4 A100 GPUs using the standard next-token prediction objective. The VLM vision encoder and the 3D foundation model remain frozen throughout training. In the first stage, we freeze the LLM and train only the merger and compressors for 18k optimizer steps. In the second stage, we continue training the merger and compressors while also applying LoRA (Hu et al., 2022) to the LLM, with rank 16, for 30k optimizer steps. In both stages we use a batch size of 128, a learning rate of 10^{-4} and a 10% warm-up period.

```
def merge_block(self, feat, cond):
   residual = feat
   cond = self.cond_proj(self.cond_norm(cond))
   shift, scale, gate = cond.chunk(3, dim=-1)
   feat = self.feat_norm(feat)
   feat = feat * (1 + scale) + shift
   feat = self.feat_proj(feat)
   return residual + gate * feat

def merger(self, vlm_features,
           fm3d_features_list):
   # B, N, seqlen, D
   num_tokens = fm3d_features_list[0].shape[2]
   feats = resize(vlm_features, num_tokens)
   for fm3d_feature in fm3d_features_list:
      feats = self.merge_block(feats,
                               fm3d_feature)
   return feats
```

Figure 3: Pseudo-code for our merger.

```
def compressor_layer(self, latents,
                     frame_features,
                     question_features):
   x = latents
   x = x + self.ca1(x, frame_features)
   x = x + self.ca2(x, question_features)
   x = x + self.mlp1(x)
   x = x + self.sa(x)
   x = x + self.mlp2(x)
   return self.merge_block(latents, x)
```

Figure 4: Pseudo-code for our compressor layer. We use standard pre-norm (omitted).

B Detailed description of the training dataset

2D image captioning data. We source image captioning data from FineVision (Wiedmann et al., 2025), a large-scale open VLM training corpus of 24M samples unified from over 200 public sources into 185 subsets. We retain 7M of them: real images paired with captions or descriptions, excluding charts, mathematics, OCR, VQA, *etc.* .

VSI-590k and VLM-3R. These datasets, introduced by Yang et al. (2025b) and Fan et al. (2026), contain 590K and 340K video QA pairs respectively. Both follow the question structure of VSI-Bench (Yang et al., 2025a) and are commonly used for instruction fine-tuning on that benchmark. We sample 32 frames from the videos.

SPAR-7M-RGBD. SPAR-7M-RGBD (Zhang et al., 2025a) is a 33-task spatial suite built on ScanNet, ScanNet++, and Structured3D. We retain only the single- and multi-view splits of these three partitions, discarding the 10 video-based tasks.

Templated QA. Our templated spatial reasoning QA data is built on top of five 3D indoor scan datasets: ScanNet (Dai et al., 2017), ScanNet++ (Yeshwanth et al., 2023), ARKitScenes (Baruch et al., 2021), 3RScan (Wald et al., 2019), and MultiScan (Mao et al., 2022). Each question is generated from an instance-labeled 3D mesh together with video frames of the same scene. Except for ScanNet++, which already provides images aligned to its labeled mesh, we take the instance-labeled meshes from SceneVerse (Jia et al., 2024) and align the video frame poses to them. From the resulting collection of roughly 4.6K scenes we sample nearly 100K sub-scenes of 3–15 frames each, growing every sub-scene by iteratively adding frames that have sufficient overlap with at least one already selected frame.

For each sub-scene we then extract object-level 3D information from the mesh: position and size from a 3D oriented bounding box, per-view 2D bounding boxes, rendered segmentation masks, and pairwise object distances measured between nearest points. We prompt a strong VLM (Qwen3.5-27B (QwenTeam, 2026)) with the frame set, marking the target object with 2D bounding boxes across views (see Fig. 5), to obtain descriptive captions and referring expressions. A VLM judge is used to filter these annotations on object identifiability, bounding-box consistency, and presence of distractors. The pipeline yields approximately 1.5M object-level descriptions covering object appearance, attributes, and local context.

Scene captioning (3D). For each of the 100K sub-scenes extracted for Templated QA, we additionally prompt the same VLM (Qwen3.5-27B) for a set of detailed scene descriptions, each targeting one of eight prescribed aspects (see examples in Fig. 5): a general description; room geometry; relationships between objects; object orientations; spatial descriptions from imagined viewpoints; the functional or zonal organization of the room; salient objects; and object affordances, *i.e.*the actions the objects support.

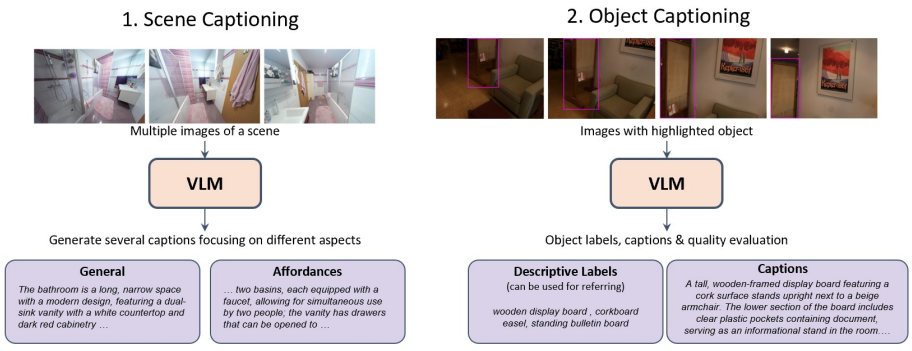


Figure 5: Scene- and object-level multi-view annotation with VLM (Qwen3.5-27B).