FEELING THE SPACE: EGOMOTION-AWARE VIDEO REPRESENTATION FOR EFFICIENT AND ACCURATE 3D SCENE UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent Multimodal Large Language Models (MLLMs) have shown high potential for spatial reasoning within 3D scenes. However, they typically rely on computationally expensive 3D representations like point clouds or reconstructed Bird's-Eye View (BEV) maps, or lack physical grounding to resolve ambiguities in scale and size. This paper significantly enhances MLLMs with egomotion modality data, captured by Inertial Measurement Units (IMUs) concurrently with the video. In particular, we propose a novel framework, called `Motion-MLLM`, introducing two key components: (1) a cascaded motion-visual keyframe filtering module that leverages both IMU data and visual features to efficiently select a sparse yet representative set of keyframes, and (2) an asymmetric cross-modal fusion module where motion tokens serve as intermediaries that channel egomotion cues and cross-frame visual context into the visual representation. By grounding visual content in physical egomotion trajectories, `Motion-MLLM` can reason about absolute scale and spatial relationships across the scene. Our extensive evaluation shows that `Motion-MLLM` makes significant improvements in various tasks related to 3D scene understanding and spatial reasoning. Compared to state-of-the-art (SOTA) methods based on video frames and explicit 3D data, `Motion-MLLM` achieves competitive accuracy while running $1.30\times$ and $1.61\times$ faster, respectively.

1 INTRODUCTION

Rapid advances in Multimodal Large Language Models (MLLMs) (Liu et al., 2023; Li et al., 2023; Alayrac et al., 2022; Bai et al., 2025) have demonstrated strong capabilities in jointly reasoning over multimodal inputs such as images, videos, and audio to produce contextually grounded responses (Lin et al., 2024; Li et al., 2024a). This progress has led to the adoption of MLLMs in applications such as embodied AI (Zheng et al., 2024; Li et al., 2024b), robotic navigation (Zhang et al., 2024a; Zhou et al., 2024), and autonomous driving (Guo et al., 2024; Chen et al., 2024b). A shared requirement across these applications is *spatial intelligence*, i.e., understanding and reasoning about the 3D structure of the physical world. Despite excelling at textual and 2D visual understanding tasks, existing MLLMs remain limited in 3D spatial reasoning (Yang et al., 2025; Chen et al., 2024a).

Existing studies on 3D spatial reasoning in MLLMs generally follow two directions, as shown in Fig. 1. One line of work incorporates explicit 3D data, such as point clouds (Hong et al., 2023; Xu et al., 2024), depth maps (Zhu et al., 2024; Zheng et al., 2025b), or BEV maps (Qi et al., 2025), into the model. These methods achieve effective spatial awareness but rely on specialized sensors or computation-intensive processing pipelines. Another line of work avoids explicit 3D input and instead extracts geometric features directly from 2D video (Zheng et al., 2025a; Wu et al., 2025). While more practical, they cannot recover reliable metric scale from monocular geometry, thus limiting their capability to resolve ambiguities in distance and size.

In this paper, we introduce egomotion as a new input modality for MLLMs (Fig. 1), captured by low-cost Inertial Measurement Units (IMUs) that are widely deployed on video-capturing platforms such as smartphones, robots, and vehicles (Brossard et al., 2020). Our motivation stems from how humans perceive their surroundings: we do not rely on vision only but continuously integrate *bodily motion cues* to understand the space. For instance, eye movements and head rotation provide information on the relative positions of objects we see, while the distance we travel helps us judge the scale of a

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.

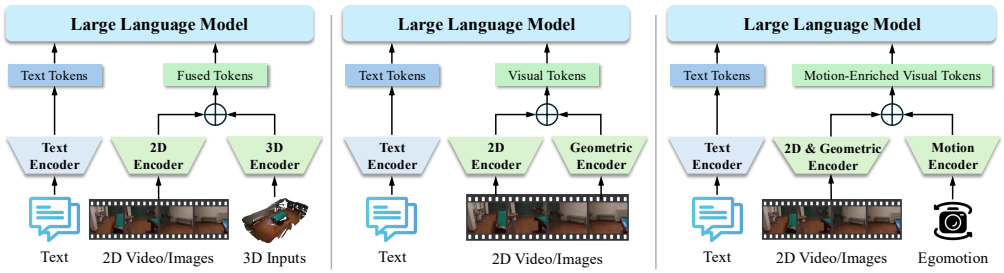


Figure 1: Comparison of (a) 3D-input, (b) 2D-input, and (c) our egomotion-input approaches for spatial reasoning in MLLMs.

room or the size of an object. Similarly, the egomotion data recorded by IMUs measures physical movements, providing reliable metric cues that ground visual observations in the real-world scale, and measuring the rotation and displacement between viewpoints directly rather than inferring them from image correspondence. This allows MLLMs to resolve the distance and size ambiguities that limit vision-only approaches, and to relate observations made from different viewpoints, without incurring the overhead of explicit 3D representations.

We propose `Motion-MLLM`, a novel framework built on two key components designed to synergistically leverage the motion modality. First, we introduce a *cascaded motion-visual keyframe filtering module* that selects a sparse yet representative set of keyframes from the video sequence. It employs a three-stage cascade progressing from lightweight egomotion checks to demanding visual feature analysis, so that computationally expensive operations are performed only for a small subset of informative frames. We then design an *asymmetric cross-modal fusion module* that integrates visual and egomotion features through two-layer cross-attention fusion. Specifically, we adopt a GRU (Gated Recurrent Unit) based encoder to compress variable-length IMU segments between keyframes into motion tokens, which are then fused with visual tokens through a bidirectional attention layer followed by a unidirectional layer. In this design, motion tokens serve as intermediaries that channel egomotion cues and cross-frame context into the visual representation, producing egomotion-enriched visual tokens. By grounding visual content in physical egomotion trajectories, `Motion-MLLM` enables LLMs to reason about absolute scale and spatial relationships across the scene.

Extensive experimentation on five 3D scene understanding and spatial reasoning benchmarks shows that `Motion-MLLM` achieves an overall score of 58.2 on VSI-Bench (Yang et al., 2025), a $+7.6$ improvement over the SOTA despite using only $\sim$4B parameters. On ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2022) benchmarks, `Motion-MLLM` significantly outperforms all 2D-input baselines and is competitive with SOTA 3D-input methods. On visual grounding (ScanRefer (Chen et al., 2020)) and dense captioning (Scan2Cap (Chen et al., 2021)) tasks, `Motion-MLLM` rivals or even outperforms 3D-input methods without requiring explicit 3D data. We further annotate a small benchmark over six TUM-VI (Schubert et al., 2018) sequences carrying real IMU recordings, where `Motion-MLLM` leads by 9.1, confirming that the advantage transfers from synthesized to real sensor data. Moreover, `Motion-MLLM` runs $1.30\times$ and $1.61\times$ faster than SOTA 2D- and 3D-input methods at matched accuracy, respectively, demonstrating that egomotion grounding delivers stronger spatial reasoning than vision-only approaches while avoiding the overhead of explicit 3D representations.

2 Related Work

MLLMs for Scene Understanding. One line of work extends LLMs to 3D scenes through explicit 3D representations. 3D-LLM (Hong et al., 2023), PointLLM (Xu et al., 2024) and Point-Bind (Guo et al., 2023) use point cloud encoders to let LLMs perceive 3D objects, LL3DA (Chen et al., 2024c) further supports interactive visual prompts such as 3D bounding boxes, and Chat3D (Wang et al., 2023) and Chat-Scene (Huang et al., 2024) adopt object-level scene representations. LLaVA-3D (Zhu et al., 2024) projects 2D features into 3D space using depth maps, GPT4Scene (Qi et al., 2025) constructs BEV maps through 3D reconstruction, and Video-3D LLM (Zheng et al., 2025b) injects 3D position encodings into video representations. These inputs are expensive to acquire or computationally heavy. A second line of work avoids explicit 3D input: SpaceR (Ouyang et al., 2025) and VideoChat (Li et al., 2025a) elicit 3D reasoning directly from existing VLMs, while VG-LLM (Zheng et al., 2025a)

and Spatial-MLLM (Wu et al., 2025) add a visual geometry encoder. These approaches reduce the dependency on 3D inputs but remain limited in resolving ambiguities in absolute scale and distance. Our method augments geometric features with egomotion data from IMU sensors, which provides the absolute metric scale that monocular geometry cannot reliably recover, and which measures the relation between viewpoints without depending on visual correspondence.

Motion Modality for Scene Understanding. Motion has long been a key part of 3D vision, especially in areas like autonomous driving and robotics. Traditional systems like ORB-SLAM3 (Campos et al., 2021) and newer neural approaches like SplaTAM (Keetha et al., 2024) or GS-SLAM (Yan et al., 2024) show how tracking camera movement is essential for building high-fidelity maps. In the MLLM field, omni-modality models like PandaGPT (Su et al., 2023) and One-LLM (Han et al., 2024) have begun incorporating IMU data, while SensorLLM (Li et al., 2025b) and LLM4HAR (Hong et al., 2025) align IMU data with text, mostly for recognizing human activities. HIS-GPT (Zhao et al., 2025) and VMRMOT (Lv et al., 2025) integrate motion dynamics to capture human-scene interactions and improve multi-object tracking. In contrast, our work treats IMU data as a critical signal for both visual filtering and physical grounding. To the best of our knowledge, we are the first to use concurrent camera egomotion for scene-level spatial reasoning in MLLMs.

3 DESIGN OF `Motion-MLLM`

We propose a framework, `Motion-MLLM`, that enhances MLLMs for 3D scene understanding by introducing egomotion as an explicit input modality alongside visual observations. Fig. 2 shows the overall architecture of `Motion-MLLM`. It takes two synchronized inputs: 2D video streams and concurrently captured IMU data that records camera motion. IMU sensors are commonly available on widely deployed video-capturing platforms, including smartphones, robotic systems, and autonomous vehicles. We detail two key components of `Motion-MLLM`: (1) a cascaded motion-visual keyframe filtering module (Sec. 3.1) that selects information-rich frames from the video stream in a computationally efficient manner; and (2) an asymmetric cross-modal feature fusion module (Sec. 3.2) that integrates visual and motion features through two-layer cross-modal attention.

3.1 CASCADED MOTION-VISUAL KEYFRAME FILTERING

Due to the limited GPU memory and the high redundancy in consecutive video frames, MLLMs can typically process only a small subset of frames from a video. A common solution (Bai et al., 2025; Yang et al., 2025) is uniform sampling, which often wastes the frame budget on static or repetitive segments and misses brief, dynamic events. Some recent works (Zheng et al., 2025b; Wu et al., 2025) propose to select frames whose geometric features maximally cover the 3D scene. These methods extract 3D features from a large number of candidate frames and rely on additional 3D inputs such as depth maps, incurring substantial overhead.

To overcome these limitations, we introduce a motion-visual keyframe filtering module, which selects frames based on both camera motion and visual content. Our key insight is that egomotion data provides a lightweight criterion for keyframe selection: a keyframe should be selected when it corresponds to a significant change in camera pose. To implement this efficiently, we propose a cascaded filtering pipeline. Instead of evaluating all criteria on every frame, this pipeline progresses from inexpensive checks to demanding analyses, reserving expensive computations (e.g., visual feature extraction) for a small subset of candidate frames. The filtering proceeds in up to three stages per frame (see Sec. A.1 for the algorithm details and threshold settings):

Stage 1: Motion Gate. The IMU sensor records 6-axis inertial data (3-axis accelerometer and 3-axis gyroscope) concurrently with the video, providing measurements between consecutive frames. For a candidate frame f_t, the translational displacement $d(\hat{f}_j, f_t)$ and the rotation angle $\theta(\hat{f}_j, f_t)$ since the most recently selected keyframe $\hat{f}_j$ can be easily obtained by integrating the accelerometer and gyroscope readings. A frame f_t is discarded if $d(\hat{f}_j, f_t) < \tau_d$ and $\theta(\hat{f}_j, f_t) < \tau_\theta$, where τ_d and τ_θ are predefined translation and rotation thresholds. This check is very simple and fast but can discard the vast majority of redundant frames where the camera is static or moves very slowly.

Stage 2: Lightweight Geometric Change Detection. This stage performs a lightweight check on the frames that pass Stage 1. It employs a sparse feature tracker from a SLAM front end (Campos et al., 2021) to avoid full feature extraction. Specifically, it calculates the average parallax of features

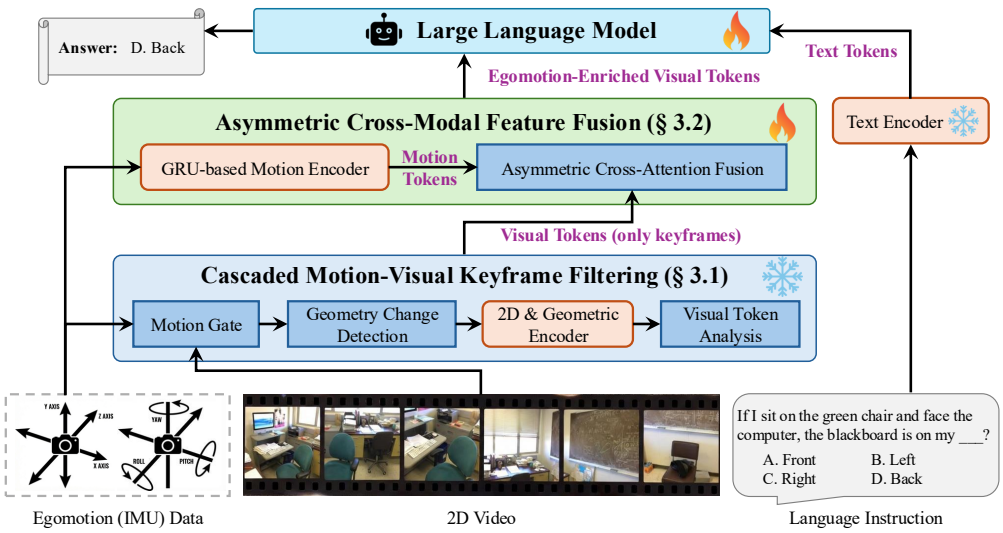


Figure 2: **Architecture of** `Motion-MLLM`**.** The 🔥 and ❄ icons indicate trainable and frozen modules, respectively.

that are tracked in the last keyframe. A high parallax value indicates a substantial geometric change in the frame content. If the parallax is not significant, the frame is discarded. Otherwise, it proceeds to Stage 3.

Stage 3: Visual Token Analysis. This stage first extracts visual features from each candidate frame f_t that passed the preceding checks. Following Wu et al. (2025), we obtain a fused visual token $\mathbf{v}_t$ by integrating 2D features from the MLLM's 2D visual encoder (e.g., Qwen2.5-VL's visual encoder (Bai et al., 2025)) with geometric features from the VGGT backbone (Wang et al., 2025). Since the cascaded pipeline filters out most frames in the first two stages, both 2D and geometric encoders run only on the small subset that passed the inexpensive egomotion and parallax checks. We then compute the cosine distance between $\mathbf{v}_t$ and the last keyframe's token $\mathbf{v}_j$. If the distance exceeds a threshold τ_v, the candidate is selected as a new keyframe. In particular, the N final selected keyframes yield visual tokens $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ that are directly input to the cross-modal fusion module (Sec. 3.2), avoiding redundant feature extraction.

3.2 Asymmetric Cross-Modal Feature Fusion

While the keyframe filtering module reduces redundancy in the input video, we need to integrate visual and egomotion information into an egomotion-aware video representation for 3D scene understanding. As illustrated in Fig. 3, we first encode IMU data segments into compact motion features. We then design an asymmetric cross-attention mechanism that fuses them with the visual features.

3.2.1 GRU-based Motion Encoder.

Between consecutive keyframes $\hat{f}_{i-1}$ and $\hat{f}_i$, the IMU segment $S_i \in \mathbb{R}^{L_i \times 6}$ has variable length L_i because keyframes are non-uniformly selected. We employ a 2-layer bidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) with hidden size 256 to encode each segment. GRU-based encoders naturally handle variable-length temporal sequences and remain the standard choice for IMU sequence encoding in inertial navigation (Chen et al., 2018; Herath et al., 2020). Sec. B.1 compares this choice against preintegration, temporal convolution and Transformer encoders. We take the final hidden state as the motion token $\mathbf{m}_i$, which summarizes the cumulative egomotion integrated between consecutive keyframes. For the first keyframe $\hat{f}_1$, we introduce a learnable start token $\mathbf{m}_1$. The resulting motion tokens form $\mathbf{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_N\}$, where N is the number of selected keyframes. Each motion token has a one-to-one correspondence with the visual tokens from the keyframe filtering stage.

3.2.2 Asymmetric Two-Layer Cross-Attention Fusion.

We design a two-layer cross-attention fusion module to integrate the visual and egomotion modalities. As shown in Fig. 3, this module takes visual tokens $\mathbf{V}$ and motion tokens $\mathbf{M}$ and produces egomotion-enriched visual tokens $\bar{\mathbf{V}}$, which maintains the same dimensionality as $\mathbf{V}$.

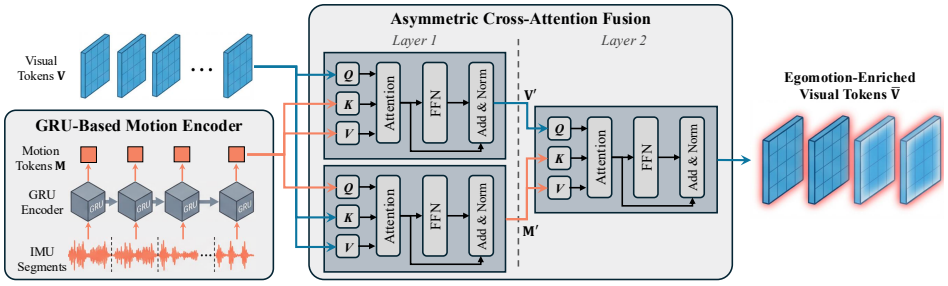


Figure 3: Illustration of asymmetric cross-modal feature fusion.

We first apply a bidirectional cross-attention layer where both modalities query each other. In the motion-to-vision direction, motion tokens serve as keys and values, providing absolute metric scale and egomotion trajectories that complement the geometric information in the visual tokens. In the vision-to-motion direction, visual tokens serve as keys and values to enrich motion features with visual context about the scenes. The two attention operations are computed as:

$$\begin{aligned} \mathbf{V}' &= \mathbf{V} + \mathrm{FFN}(\mathrm{Attn}(\mathbf{V}W_Q^v,\ \mathbf{M}W_K^m,\ \mathbf{M}W_V^m)) \\ \mathbf{M}' &= \mathbf{M} + \mathrm{FFN}(\mathrm{Attn}(\mathbf{M}W_Q^m,\ \mathbf{V}W_K^v,\ \mathbf{V}W_V^v)) \end{aligned} \tag{1}$$

where $\mathrm{Attn}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$, all W are learnable projection matrices, and $\mathrm{FFN}(\cdot)$ denotes a feed-forward network (Vaswani et al., 2017) applied with a residual connection. Both attention layers apply rotary positional embeddings (RoPE) (Su et al., 2024) to queries and keys, with $\mathbf{v}_i$ and $\mathbf{m}_i$ sharing position index i to align each motion token with its paired keyframe.

Next, we fuse $\mathbf{V}'$ and $\mathbf{M}'$ with a unidirectional cross-attention layer. Only visual tokens query motion tokens in this layer, creating a visual→motion→visual information pathway. Since $\mathbf{M}'$ already carries visual context from other keyframes, this unidirectional attention layer enables motion-guided inter-frame visual communication, where motion tokens act as bridges that relay visual information across frames along physically grounded egomotion trajectories. The attention is computed as:

$$\bar{\mathbf{V}} = \mathbf{V}' + \mathrm{FFN}(\mathrm{Attn}(\mathbf{V}'W_Q,\ \mathbf{M}'W_K,\ \mathbf{M}'W_V)) \tag{2}$$

This asymmetric design retains only visual tokens as the output, while motion tokens are discarded after channeling cross-frame information into the visual tokens. Finally, the enriched visual tokens $\bar{\mathbf{V}}$ are fed into the MLLM alongside the text prompt through the standard visual-language pipeline.

4 Experimental Evaluation

4.1 Experimental Setup

Datasets. We evaluate on ScanQA (Azuma et al., 2022), SQA3D (Ma et al., 2022) and VSI-Bench (Yang et al., 2025) for spatial reasoning, ScanRefer (Chen et al., 2020) for visual grounding, and Scan2Cap (Chen et al., 2021) for dense captioning. `Motion-MLLM` takes egomotion data as input, which is absent from these existing datasets. We therefore synthesize camera egomotion data in the format of a typical IMU sensor output. A unified pipeline is used across all five benchmarks: cubic B-splines (Lovegrove et al., 2013) fit to source-dataset ground-truth camera poses are analytically differentiated, sampled at 200 Hz, and augmented with a standard MEMS noise model (Forster et al., 2016). For VSI-Bench, the official per-video metadata (Yang et al., 2025) links each clip to its source scan in ScanNet, ScanNet++ or ARKitScenes, from which we obtain the poses. These validation splits do not overlap with our training data. `Motion-MLLM` consumes only the 2D video frames and the noisy IMU stream at inference, never observing the camera poses from which that stream was synthesized, so all pose-derived information reaches the model through the same six-axis channel an IMU sensor would output. We further validate `Motion-MLLM` on a small-scale benchmark we construct from 6 TUM-VI (Schubert et al., 2018) indoor sequences with real IMU recordings. The synthesized IMU stream is noisier than a real sensor, not cleaner, with noise densities and bias random walks higher than the Allan-variance calibration TUM-VI reports for its Bosch BMI160 IMU. Since TUM-VI provides a mocap trajectory alongside its real IMU recordings, we evaluate the same checkpoint on the same questions under both streams. The two differ by only 0.1, so results

Table 1: **Evaluation Results on ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2022).** "2D", "3D", and "M" specify the model's input type as 2D data, 3D data, and egomotion data, respectively.

Methods	Input			ScanQA (val)					SQA3D (test)	
	2D	3D	M	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	EM@1	EM@R1
Task-Specific Models										
SQA3D (2022)		✓		**30.5**	**11.2**	13.5	34.5	-	46.6	-
3D-VisTA (2023)		✓		-	10.4	**13.9**	**35.7**	**69.6**	**48.5**	-
2D-Input Models										
Qwen2.5-VL-3B (2025)	✓			22.5	3.8	9.7	25.4	47.4	43.4	45.9
Qwen2.5-VL-7B (2025)	✓			27.8	3.0	11.4	29.3	53.9	46.5	49.8
Spatial-MLLM (2025)	✓			**44.4**	**14.8**	**18.4**	**45.0**	**91.8**	**55.9**	**58.7**
3D/2.5D-Input Models										
LL3DA (2024c)		✓		-	13.5	15.9	37.3	76.8	-	-
GPT4Scene-HDM (2025)	✓	✓		44.4	15.5	18.9	46.5	96.3	**60.6**	**63.3**
Video-3D LLM (2025b)	✓	✓		**47.1**	16.2	19.8	49.0	102.1	58.6	-
LLaVA-3D (2024)	✓	✓		-	**16.4**	**20.8**	**49.6**	**103.1**	60.1	-
Egomotion-Input Model										
`Motion-MLLM`**-4B**	✓		✓	**46.5**	**15.8**	**20.3**	**48.0**	**100.4**	**60.2**	**62.4**

obtained with the synthesized stream carry over to a real sensor. Our synthesis pipeline and these measurements are detailed in Sec. A.2.

Implementation and training details. `Motion-MLLM` is built upon Qwen2.5-VL-3B (Bai et al., 2023) (4.3B parameters total), using the 2D visual encoder of Qwen2.5-VL (Bai et al., 2025) together with the VGGT backbone (Wang et al., 2025) to produce visual tokens $\mathbf{V}$, and its LLM backbone for text. We train a single generalist model on the training splits of ScanQA (Azuma et al., 2022), SQA3D (Ma et al., 2022), ScanRefer (Chen et al., 2020) and Scan2Cap (Chen et al., 2021) (Sec. A.3), sampling each batch from a single task type. Question answering uses a standard cross-entropy loss. Visual grounding uses the same pre-extracted object proposals as VG LLM (Zheng et al., 2025a) for fair comparison, and follows Zheng et al. (2025b) in supervising proposal selection with a contrastive loss (Oord et al., 2018). Training runs in two stages of one epoch each. The first freezes the visual encoder, the VGGT backbone and the LLM, training only the motion encoder and cross-attention modules at a learning rate of 1e-4, so the model first learns to encode egomotion and fuse it with visual features. The second unfreezes the LLM and fine-tunes it end-to-end with those modules, keeping both encoders frozen, using Adam (Kingma & Ba, 2014) with a global batch size of 32 and a learning rate warmed to 1e-5 over the first 3% of steps then decayed linearly. We preprocess the video for each scan to 640×480 resolution. Training uses 8 NVIDIA RTX 4000 Ada GPUs and takes $\sim$15 hours. All result numbers are averaged over three independent runs with different random seeds, with a standard deviation of ± 0.5 on accuracy metrics and ± 1.5 on caption metrics.

4.2 Spatial Reasoning Benchmarks

Evaluation on ScanQA and SQA3D. ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2022) are 3D question-answering benchmarks built on ScanNet (Dai et al., 2017) indoor scenes. We evaluate `Motion-MLLM` on the ScanQA validation set (4,675 QA pairs) and the SQA3D test set (3,519 QA pairs). As shown in Tab. 1, `Motion-MLLM` outperforms task-specific and 2D-input models across all metrics, achieving $+8.6$ CIDEr and $+4.3$ EM@1 over the SOTA 2D-input baseline, which shows that the egomotion modality enhances spatial and positional reasoning. Among 3D/2.5D-input models, only Video-3D LLM (Zheng et al., 2025b) (on ScanQA), GPT4Scene (Qi et al., 2025) (on SQA3D) and LLaVA-3D (Zhu et al., 2024) (on ScanQA) surpass `Motion-MLLM`, and these rely on point clouds or depth maps that incur high data collection and computation costs. `Motion-MLLM` comes within 2.7 CIDEr and 0.4 EM@1 of them using low-cost egomotion data alone. Complete ScanQA and SQA3D results are provided in Sec. C.1, and Sec. B.5 breaks the gain down by answer type.

Evaluation on VSI-Bench. VSI-Bench (Yang et al., 2025) contains over $5,000$ question-answer pairs curated from ScanNet (Dai et al., 2017), ScanNet++ (Yeshwanth et al., 2023) and ARKitScenes (Baruch et al., 2021) across eight subtasks. As shown in Tab. 2, `Motion-MLLM` scores best on five of eight subtasks and raises the overall average to 58.2, a $+7.6$ improvement over the SOTA baseline, outperforming proprietary and open-source models up to 78B with only $\sim$4B parameters. On the other three (Abs. Dist., Route Plan, Appr. Order) it trails a larger model by at most 3.2. Against Spatial-MLLM (Wu et al., 2025), which uses the same VGGT encoder but lacks our IMU

Table 2: **Evaluation Results on VSI-Bench (Yang et al., 2025).** We follow the standard setting of VSI-Bench. **Bold** and underline denote the best and second-best results in each column, respectively.

Methods	Numerical Question				Multiple-Choice Question				Avg.
	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order.	
Proprietary Models (API)									
GPT-4o (2023)	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5	34.0
Gemini-1.5 Pro (2024)	56.2	30.9	64.1	43.6	51.3	46.3	**36.0**	34.6	45.4
Open-source Models									
LLaVA-OneVision-72B (2024a)	43.5	23.9	57.6	37.5	42.5	39.9	32.5	44.6	40.3
Qwen2.5-VL-3B (2025)	24.3	24.7	31.7	22.6	38.3	41.6	26.3	21.2	28.8
Qwen2.5-VL-7B (2025)	40.9	14.8	43.4	10.7	38.6	38.5	33.0	29.8	31.2
InternVL3-78B (2025)	71.2	**53.7**	44.4	39.5	55.9	39.5	28.9	54.5	48.5
Spatial Reasoning Models									
Spacer (2025)	57.8	28.2	59.9	47.1	40.1	45.4	33.5	52.1	45.5
VG LLM-4B (2025a)	66.0	37.8	55.2	59.2	44.6	45.6	33.5	36.4	47.3
VG LLM-8B (2025a)	67.9	37.7	58.6	62.0	46.6	40.7	32.4	**59.2**	50.6
Spatial-MLLM-4B (2025)	65.3	34.8	63.1	45.1	41.3	46.2	33.5	46.3	47.0
`Motion-MLLM`-**4B**	**72.6**	52.0	**66.4**	**65.8**	**61.2**	**55.8**	35.8	56.0	**58.2**

Table 3: **Real IMU validation on TUM-VI (Schubert et al., 2018).** We follow the standard setting of VSI-Bench and utilize the keyframe filtering module to select frames for baselines and `Motion-MLLM`, respectively.

Methods	Numerical Question			Multiple-Choice Question			Avg.
	Obj. Cnt.	Abs. Dist.	Room Size	Rel. Dist.	Rel. Dir.	Appr. Order	
Qwen2.5-VL-3B (2025)	21.7	19.6	20.8	35.0	35.0	20.0	25.4
VG LLM-4B (2025a)	50.4	25.4	48.3	40.0	45.0	30.0	39.9
Spatial-MLLM-4B (2025)	50.0	24.6	38.3	40.0	45.0	35.0	38.8
`Motion-MLLM`-**4B**	**54.6**	**39.2**	**55.0**	**45.0**	**55.0**	**45.0**	**49.0**

Table 4: **Evaluation of 3D scene understanding tasks.** For ScanRefer, scores include proposal refinement following SPAR (Zhang et al., 2025); "()" indicates the score without refinement.

Methods	Input			ScanRefer		Scan2Cap			
	2D	3D	M	Acc@0.25↑	Acc@0.5↑	C@0.5↑	B-4@0.5↑	M@0.5↑	R@0.5↑
2D-Input Models									
VG LLM-8B (2025a)	✓			**57.6 (41.6)**	**50.9 (14.9)**	**80.0**	**41.5**	**28.9**	**62.6**
2.5D/3D-Input Models									
GPT4Scene-HDM (2025)	✓	✓		**62.6**	**57.0**	-	40.6	-	59.3
LLaVA-3D (2024)	✓	✓		54.1	42.4	79.2	**41.1**	**30.2**	**63.4**
Video-3D LLM (2025b)	✓	✓		58.1	51.7	**80.0**	40.2	28.5	61.7
Egomotion-Input Models									
`Motion-MLLM`-**4B**	✓		✓	**61.4 (45.8)**	**55.3 (19.6)**	**79.0**	**41.6**	**30.0**	**64.0**

encoder and cross-modal fusion, `Motion-MLLM` improves by $+17.2$ on the cross-frame Abs. Dist. task but only $+3.3$ on the intra-frame Obj. Size task, attributing the gap to IMU-encoded inter-frame egomotion and asymmetric cross-modal fusion. It also benefits view-integration tasks (e.g., $+9.6$ on relative direction), where motion features bridge observations across viewpoints. Sec. B.5 reports the full per-task and per-source breakdown against the matched-protocol baseline.

Real IMU Validation on TUM-VI. We manually construct a small benchmark of 120 QA pairs over the 6 indoor sequences of TUM-VI (Schubert et al., 2018) (which provides real 200 Hz BMI160 IMU recordings), covering six tasks and following VSI-Bench (Yang et al., 2025) question templates (details are provided in Sec. A.4). As shown in Tab. 3, `Motion-MLLM` reaches an average of 49.0, $+9.1$ over the best-performing baseline, including $+13.8$ on Abs. Dist., confirming that the egomotion advantage transfers from synthesized to real IMU data.

4.3 3D Scene Understanding Tasks

We further assess `Motion-MLLM` on ScanRefer (Chen et al., 2020) and Scan2Cap (Chen et al., 2021), which require linking textual descriptions to specific 3D structures. As shown in Tab. 4, `Motion-MLLM` reaches Acc@0.25/0.5 of $61.4/55.3$ on ScanRefer with only $\sim$4B parameters and

Table 5: **Cost-effectiveness comparison on ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2022).** "MC" represents the maximum coverage sampling (Wu et al., 2025; Zheng et al., 2025b). "MV Filtering" denotes the motion-visual keyframe filtering method we design. "T" represents the end-to-end time consumption.

Methods	Input			Sampling Strategy	# Frames	ScanQA			SQA3D		
	2D	3D	M			EM(%)↑	T(s)↓	CE(s^{-1})↑	EM(%)↑	T(s)↓	CE(s^{-1})↑
Qwen2.5-VL-3B (2025)	✓			Uniform	128	17.1	2.45	0.07	49.4	2.39	0.21
				Uniform	32	15.5	<u>0.72</u>	0.22	45.7	<u>0.71</u>	0.64
Spatial-MLLM (2025)	✓			Uniform	128	28.8	3.06	0.09	<u>62.2</u>	3.01	0.21
				Uniform	32	26.2	0.88	0.30	58.5	0.87	0.67
				MC	∼23	25.9	0.79	<u>0.33</u>	57.6	0.79	<u>0.73</u>
Video-3D LLM (2025b)	✓	✓		Uniform	128	<u>31.7</u>	3.45	0.09	60.8	3.36	0.18
				Uniform	32	30.1	1.22	0.25	58.6	1.19	0.49
				MC	∼18	29.5	0.98	0.30	57.7	0.97	0.59
Motion-MLLM-**4B**	✓		✓	Uniform	128	**31.5**	3.10	0.10	**62.8**	3.05	0.21
				Uniform	32	29.1	0.90	0.32	59.5	0.88	0.68
				MV Filtering	∼21	29.8	**0.61**	**0.49**	60.2	**0.59**	**1.02**

no explicit 3D input, exceeding the SOTA 2D-input baseline VG LLM-8B (Zheng et al., 2025a) by $+3.8$ and $+4.4$ and outperforming 3D-input methods such as LLaVA-3D (Zhu et al., 2024) and Video-3D LLM (Zheng et al., 2025b). Only GPT4Scene-HDM (Qi et al., 2025), which constructs BEV maps through 3D reconstruction, scores higher. On Scan2Cap it sets the best B-4@0.5 (41.6) and R@0.5 (64.0) across all baselines, with C@0.5 (79.0) and M@0.5 (30.0) within 1.0 of the best 3D-input models. Full results and analysis are in Sec. C.2.

4.4 Cost-Effectiveness of Motion-MLLM

We evaluate the cost-effectiveness of Motion-MLLM on ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2022). Beyond the standard Exact Match (EM) accuracy and end-to-end inference time T, we report speedup at matched accuracy against each baseline's best operating point, summarized by the ratio $\mathrm{CE} = \mathrm{EM}\%/T$ (higher is better). The full Pareto analysis is in Sec. C.3. We compare Motion-MLLM against SOTA 2D- and 3D-input models under uniform sampling and method-specific adaptive strategies (Tab. 5).

Superior efficiency under uniform frame sampling. Under uniform sampling, more frames yield higher accuracy but incur significant computational overhead. Motion-MLLM is nonetheless more cost-effective than both. Compared to 2D baselines (e.g., Spatial-MLLM (Wu et al., 2025)), our model is more accurate (29.1% vs. 26.2% on ScanQA with 32 frames) at comparable latency (0.90s vs. 0.88s). Compared to 3D-input models such as Video-3D LLM (Zheng et al., 2025b), Motion-MLLM achieves comparable accuracy (29.1 vs. 30.1 on ScanQA and 59.5 vs. 58.6 on SQA3D) while running $\sim 1.36\times$ faster (0.90s vs. 1.22s). This stems from our lightweight motion encoder and fusion, which avoid heavy 3D processing.

Benefits of egomotion-guided keyframe filtering. As shown in Tab. 5, Motion-MLLM is more accurate and faster than every baseline at its best operating point on both benchmarks. Against the SOTA 2D- and 3D-input baselines, Motion-MLLM runs $1.30\times$ and $1.61\times$ faster on ScanQA with $+3.9$ and $+0.3$ EM accuracy gains, and similar margins on SQA3D. The CE metric summarizes this as a single number. Compared to 32-frame uniform sampling, our motion-visual keyframe filtering achieves comparable accuracy (e.g., 29.8% vs. 29.1% on ScanQA) using only $\sim$ 21 keyframes, showing that egomotion identifies information-rich frames.

4.5 Ablation Study

Effectiveness of egomotion modality. Tab. 6 disentangles the contributions of egomotion and VGGT-derived geometric features. Removing IMU (VGGT-only) drops accuracy by $-3.2\%/-2.0\%$ on ScanQA/SQA3D, while removing VGGT (IMU-only) drops by $-2.3\%/-1.7\%$. The two single-modality configurations are close on these benchmarks, but separate on VSI-Bench, where IMU-only reaches 52.5 against 49.0 for VGGT-only. The egomotion pathway therefore contributes more than the geometric features on tasks that require relating observations across viewpoints.

Effectiveness of cascaded motion-visual keyframe filtering. "Full MV Filtering" improves accuracy over uniform sampling to 29.8% on ScanQA but increases latency to 0.96s due to evaluating all

Table 6: **Ablation study on ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2022).** "VGGT-only" is the Spatial-MLLM (Wu et al., 2025) configuration retrained under our pipeline. "Visual-based Sampling" uses visual criteria only, and "Full MV Filtering" evaluates all criteria on every frame. "Concat+MLP" replaces the fusion module with concatenation and an MLP. All rows except "Qwen2.5-VL-3B" are fine-tuned with our two-stage pipeline.

Type	ScanQA			SQA3D			VSI-Bench	
	EM(%)↑	T(s)↓	CE(s^{-1})↑	EM(%)↑	T(s)↓	CE(s^{-1})↑	Avg.↑	Δ
Qwen2.5-VL-3B (Backbone)	15.5	0.72	0.22	45.7	0.71	0.64	–	–
Component Ablation								
VGGT-only (no IMU)	26.6	0.79	0.34	58.2	0.79	0.74	49.0	−9.2
IMU-only (no VGGT)	27.5	0.70	0.39	58.5	0.68	0.86	52.5	−5.7
Frame Sampling Strategy								
Uniform 32-frame	29.1	0.90	0.32	59.5	0.88	0.68	58.0	−0.2
Visual-based Sampling (no IMU)	29.4	0.92	0.32	59.8	0.89	0.67	58.0	−0.2
Full MV Filtering	29.8	0.96	0.31	60.3	0.93	0.65	58.4	+0.2
Feature Connector								
Concat+MLP (95M)	19.2	0.58	0.33	41.5	0.55	0.75	45.6	−12.6
Single-Layer Cross-Attention (57M)	26.1	0.60	0.44	55.2	0.57	0.96	50.5	−7.7
Ours: Cascaded MV Filtering + Asymmetric Cross-Attention (110M)								
Full `Motion-MLLM`	**29.8**	**0.61**	**0.49**	**60.2**	**0.59**	**1.02**	**58.2**	–

criteria on every frame. Replacing IMU-based stages with visual-only criteria yields similar accuracy (29.4%) but only marginally reduces latency (0.92s), since visual feature extraction dominates the cost. Our cascaded MV filtering uses lightweight egomotion checks to discard redundancy before applying visual criteria, achieving comparable accuracy ($29.8\%/60.2\%$) at 0.61s. A detailed ablation study of each stage in the motion-visual keyframe filtering is provided in Sec. A.1.

Effectiveness of asymmetric cross-modal fusion. Concat+MLP reaches only 19.2% on ScanQA, so direct concatenation fails to correlate motion with visual content. Single-layer cross-attention reaches 26.1%. Our asymmetric design achieves the best accuracy ($29.8\%/60.2\%$) with only $+0.03$s additional latency as motion tokens learn visual context before passing it back to the visual stream. Notably, the Concat+MLP baseline is parameter-matched to our asymmetric design (95M vs. 110M) yet underperforms by -10.6% on ScanQA, confirming that the gain is primarily architectural. Sec. B.2 decomposes the module further and tests the relay mechanism causally.

Physical grounding of the egomotion signal. We examine what the model takes from the inertial stream through three interventions at inference, each leaving frame order, token count and positional encoding unchanged, with details in Sec. B.3 and Sec. B.4. Replacing the stream with zeros returns the model to the no-IMU baseline, with every structural element of the pathway still in place. Permuting the per-keyframe segments within a video, which leaves the multiset of motion tokens unchanged, drops the VSI-Bench average 4.8 below that baseline. Re-synthesizing it from trajectories scaled by α, with the video unchanged, moves predicted distances with a log-log slope of 0.73. The gain therefore depends on the physical content of the stream rather than on additional conditioning signals.

5 Conclusion

We proposed `Motion-MLLM`, which enhances MLLMs with egomotion data from IMU sensors for 3D scene understanding through two components: a cascaded keyframe filter that progresses from lightweight IMU checks to demanding visual analysis, and an asymmetric fusion module in which motion tokens channel egomotion cues and cross-frame context into the visual representation. On five benchmarks `Motion-MLLM` outperforms 2D-input baselines and matches 3D-input methods while running $1.30\times$ and $1.61\times$ faster at comparable accuracy.

Limitations and Future Work. Our evaluation covers indoor handheld capture, with the inertial stream perturbed at inference to test robustness to sensor noise, reduced sampling rate and synchronization error. Those perturbations do not cover severe motion, structured mechanical vibration, motion blur or rolling shutter, and outdoor and vehicle-mounted capture fall outside our indoor scope. Real-sensor validation is limited to six TUM-VI sequences, since no public 3D scene understanding benchmark releases paired inertial recordings. Future work will extend the framework to these settings and validate on broader IMU hardware.

AI USE STATEMENT

We used large language models for the following tasks. To implement code: model, evaluation and data-processing code written with LLM assistance was reviewed and tested by the authors, and the design of the method and of every experiment is our own. To clean and reformat data during preprocessing, with the results checked against the source annotations. To generate candidate questions for our TUM-VI benchmark by filling the templates described in Sec. A.4; every question and answer was then manually reviewed, and a second annotator independently verified a 20% random subset. To draft and edit manuscript text from author-supplied results and arguments, with all content reviewed and revised by the authors. We did not use generative AI tools to design the method, to design experiments, or to interpret results.

ETHICS STATEMENT

This work uses publicly released datasets (ScanNet, ScanNet++, ARKitScenes and TUM-VI) under their respective licenses and involves no experiments with human subjects. These indoor scans depict real homes and offices and may contain identifiable content; we use them as released and collect no new imagery. Our real-IMU benchmark adds question-answer annotations over existing TUM-VI recordings only. The method consumes concurrent video and inertial data, a combination already standard on consumer phones and headsets. As with other indoor scene-understanding systems, it could support tracking of people or spaces without consent, and deployment should be governed accordingly.

REPRODUCIBILITY STATEMENT

We will release our code and the TUM-VI question annotations upon acceptance. Sec. 4 specifies the backbone, training mixture, two-stage schedule, optimizer, learning-rate schedule and hardware, and reports all numbers as three-seed means with the observed spread. Sec. A.1 gives the filtering algorithm, all four thresholds and a sensitivity analysis. Sec. A.2 specifies the egomotion synthesis pipeline, including source poses, B-spline differentiation, the noise model and its parameters. Sec. A.3 lists the training data composition, and Sec. A.4 the construction of our real-IMU benchmark. Sec. A.5 reproduces the prompts for every task. Protocols for the analyses in Sec. B are stated in the corresponding table captions.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19129–19139, 2022.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023.

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

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.

Martin Brossard, Axel Barrau, and Silvère Bonnabel. Ai-imu dead-reckoning. *IEEE Transactions on Intelligent Vehicles*, 5(4):585–595, 2020.

Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1168–1176, 2016.

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024a.

Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pp. 202–221. Springer, 2020.

Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14093–14100. IEEE, 2024b.

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26428–26438, 2024c.

Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024d.

Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3193–3203, 2021.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

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*, pp. 5828–5839, 2017.

Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE transactions on robotics*, 33(1):1–21, 2016.

Ziang Guo, Zakhar Yagudin, Artem Lykov, Mikhail Konenkov, and Dzmitry Tsetserukou. Vlm-auto: Vlm-based autonomous driving assistant with human-like behavior and understanding for complex road scenes. *arXiv preprint arXiv:2405.05885*, 2024.

Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, et al. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023.

Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. Onellm: One framework to align all modalities with language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26584–26595, 2024.

Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 3146–3152. IEEE, 2020.

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.

Zhiqing Hong, Yiwei Song, Zelong Li, Anlan Yu, Shuxin Zhong, Yi Ding, Tian He, and Desheng Zhang. Llm4har: Generalizable on-device human activity recognition with pretrained llms. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 4511–4521, 2025.

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. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024.

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.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21357–21366, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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 preprint arXiv:2408.03326*, 2024a.

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

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 68(10):200102, 2025a.

Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18061–18070, 2024b.

Zechen Li, Shohreh Deldari, Linyao Chen, Hao Xue, and Flora D Salim. Sensorllm: Aligning large language models with motion sensors for human activity recognition. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 354–379, 2025b.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 5971–5984, 2024.

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

Steven Lovegrove, Alonso Patron-Perez, and Gabe Sibley. Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In *BMVC*, volume 2, pp. 8, 2013.

Weiyi Lv, Ning Zhang, Hanyang Sun, Haoran Jiang, Kai Zhao, Jing Xiao, and Dan Zeng. Vision-motion-reference alignment for referring multi-object tracking via multi-modal large language models. *arXiv preprint arXiv:2511.17681*, 2025.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE international conference on robotics and automation*, pp. 3565–3572. IEEE, 2007.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv preprint arXiv:2501.01428*, 2025.

Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE transactions on robotics*, 34(4):1004–1020, 2018.

David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The TUM VI benchmark for evaluating visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1680–1687. IEEE, 2018.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*, 2023.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pp. 131–147. Springer, 2024.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19595–19604, 2024.

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*, pp. 10632–10643, 2025.

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*, pp. 12–22, 2023.

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.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024a.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024b.

Jiahe Zhao, Ruibing Hou, Zejie Tian, Hong Chang, and Shiguang Shan. His-gpt: Towards 3d human-in-scene multimodal understanding. *arXiv preprint arXiv:2503.12955*, 2025.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13624–13634, 2024.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *arXiv preprint arXiv:2505.24625*, 2025a.

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*, pp. 8995–9006, 2025b.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024.

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 preprint arXiv:2409.18125*, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2911–2921, 2023.

TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A ADDITIONAL METHOD DETAILS

A.1 DETAILS ABOUT CASCADED MOTION-VISUAL KEYFRAME FILTERING

Our keyframe filtering module applies three cascaded stages to each candidate frame f_t, evaluated against the most recently selected keyframe $\hat{f}_j$. Each stage acts as a progressively more expensive filter: (1) Stage1 uses IMU-derived motion estimates; (2) Stage2 uses sparse feature tracking; and (3) Stage3 uses visual token comparison. Below are the implementation details for each stage.

Stage 1: Motion Gate. The translational displacement between the candidate frame f_t and the last keyframe $\hat{f}_j$ is obtained by double-integrating gravity-corrected accelerometer readings:

$$d(\hat{f}_j, f_t) = \|\Delta \mathbf{p}_{j\to t}\|_2 \,, \tag{3}$$

where $\Delta \mathbf{p}_{j\to t}$ is the position change estimated from double integration. The velocity state is maintained continuously across the entire sequence, so constant-velocity segments (where the accelerometer reads near-zero after gravity correction) produce non-zero displacement. The rotation angle is obtained by first integrating the gyroscope readings to form a relative rotation matrix $\Delta \mathbf{R}_{j\to t}$, then extracting the angle:

$$\theta(\hat{f}_j, f_t) = \arccos\left(\frac{\mathrm{tr}(\Delta \mathbf{R}_{j\to t}) - 1}{2}\right) \in [0, \pi]. \tag{4}$$

A candidate frame f_t is discarded if both conditions hold: $d(\hat{f}_j, f_t) < \tau_d$ and $\theta(\hat{f}_j, f_t) < \tau_\theta$. We set $\tau_d = 0.2\,\mathrm{m}$ and $\tau_\theta = 15^\circ$ based on the geometric properties of typical indoor scenes. At indoor object distances ($1 \sim 5$ m), a translation below 0.2 m produces insufficient parallax for meaningful geometric change. Similarly, a rotation below 15° introduces less than 27% new field of view given the camera's typical horizontal FOV ($\sim 55^\circ$).

Stage 2: Lightweight Geometric Change Detection. This stage follows the sparse feature tracking approach used in visual-inertial odometry systems (Qin et al., 2018; Mourikis & Roumeliotis, 2007). Let $\{p_k^j\}_{k=1}^K$ be K sparse feature points detected in the last keyframe $\hat{f}_j$. These points are tracked to the candidate frame f_t using a widely deployed feature tracker (Campos et al., 2021), yielding matched points $\{p_k^t\}$. The average parallax is computed as:

$$\bar{p}(\hat{f}_j, f_t) = \frac{1}{K'} \sum_{k=1}^{K'} \left\| p_k^t - p_k^j \right\|_2 , \tag{5}$$

where K' is the number of successfully tracked points. A candidate is discarded if $\bar{p}(\hat{f}_j, f_t) < \tau_p$, with $\tau_p = 15$ pixels. This is a conservative threshold that retains candidates with moderate geometric change for further evaluation in Stage 3.

Stage 3: Visual Token Analysis. Following the dual-encoder architecture in Spatial-MLLM (Wu et al., 2025), the 2D encoder (Qwen2.5-VL's visual encoder (Bai et al., 2025)) produces features

Algorithm 1 Cascaded Motion-Visual Keyframe Filtering

Require: Video frames $\{f_1, \ldots, f_T\}$, IMU data, thresholds τ_d, τ_θ, τ_p, τ_v
Ensure: Keyframe set $\mathcal{K}$, visual tokens $\mathbf{V}$

$\mathcal{K} \leftarrow \{f_1\}$; $\hat{f}_j \leftarrow f_1$; extract $\mathbf{v}_j$ for f_1 ; $\mathbf{V} \leftarrow \{\mathbf{v}_j\}$
for $t = 2$ **to** T **do**
  Compute $d(\hat{f}_j, f_t)$ and $\theta(\hat{f}_j, f_t)$ from IMU
  if $d(\hat{f}_j, f_t) < \tau_d$ **and** $\theta(\hat{f}_j, f_t) < \tau_\theta$ **then**
    continue {Stage 1: insufficient motion}
  end if
  Compute average parallax $\bar{p}(\hat{f}_j, f_t)$
  if $\bar{p}(\hat{f}_j, f_t) < \tau_p$ **then**
    continue {Stage 2: insufficient geometric change}
  end if
  Extract visual token $\mathbf{v}_t$
  if $1 - \cos(\mathbf{v}_t, \mathbf{v}_j) < \tau_v$ **then**
    continue {Stage 3: visually similar}
  end if
  $\mathcal{K} \leftarrow \mathcal{K} \cup \{f_t\}$; $\hat{f}_j \leftarrow f_t$; $\mathbf{v}_j \leftarrow \mathbf{v}_t$; $\mathbf{V} \leftarrow \mathbf{V} \cup \{\mathbf{v}_t\}$
end for

Table 7: **Stage-wise ablation of the cascaded filter.** The cascade is truncated after each stage, and the frames that survive are used as keyframes. Latency is end-to-end inference time on ScanQA.

Stages applied	# Keyframes	Latency	ScanQA EM	VSI Avg.
Stage 1 (motion gate)	∼80	2.18 s	30.5	58.8
Stages 1+2 (+ parallax)	∼30	0.84 s	30.0	58.3
Stages 1+2+3 (ours)	∼21	0.61 s	29.8	58.2

$\mathbf{v}_{2D}$, and the spatial encoder (VGGT backbone (Wang et al., 2025)) produces features $\mathbf{v}_{3D}$. The 3D features $\mathbf{v}_{3D}$ are rearranged to align with $\mathbf{v}_{2D}$ in spatial and temporal dimensions, yielding $\mathbf{v}'_{3D}$. The fused visual token for candidate frame f_t is:

$$\mathbf{v}_t = \mathrm{MLP}_{2D}(\mathbf{v}_{2D}) + \mathrm{MLP}_{3D}(\mathbf{v}'_{3D}). \tag{6}$$

The cosine distance is computed between globally average-pooled $\mathbf{v}_t$ and $\mathbf{v}_j$, determining whether f_t is selected as a new keyframe. The unpooled spatial tokens are retained for downstream fusion. Specifically, f_t is selected if the cosine distance exceeds $\tau_v = 0.4$ (corresponding to a cosine similarity of 0.6). Since $\mathbf{v}_t$ encodes both 2D appearance and 3D geometric structure, the cosine distance captures changes in both modalities. This threshold is tuned on the ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2022) datasets.

Algorithm and Filtering Statistics. The complete cascaded filtering procedure is summarized in Algorithm 1. On our experimental benchmarks (ScanQA (Azuma et al., 2022), SQA3D (Ma et al., 2022), VSI-Bench (Yang et al., 2025), ScanRefer (Chen et al., 2020), and Scan2Cap (Chen et al., 2021), mostly derived from ScanNet (Dai et al., 2017)), input videos contain on average ∼1,000 frames at 30 FPS. Stage1 retains ∼8% (∼80 frames), Stage2 retains ∼38% of those (∼30 frames), and Stage3 retains ∼70% of those (∼21 keyframes). The cascaded design ensures that expensive visual feature extraction in Stage3 is applied to less than 3% of the original frames.

Stage-wise Ablation. We truncate the cascade after Stage 1 and after Stage 2, and use the surviving frames as keyframes. As shown in Tab. 7, the motion gate alone leaves ∼80 frames and takes 2.18 s. The parallax check reduces this to ∼30 frames and 0.84 s, and the visual token check to ∼21 frames and 0.61 s. Across the full cascade the keyframe count falls by 74% and latency by 72%, at a cost of 0.7 on ScanQA EM and 0.6 on the VSI-Bench average. At a comparable frame count, the ∼30 frames kept after Stage 2 reach 30.0 on ScanQA, above the 29.1 of uniform sampling at 32 frames (Tab. 6).

Threshold Sensitivity. We additionally perturb each of the four thresholds (τ_d, τ_θ, τ_p, τ_v) independently by $\pm 50\%$ from its default value and re-run the keyframe filtering pipeline on ScanQA.

Table 8: **IMU noise model parameters used in egomotion data synthesis.**

Parameter	Symbol	Value
Accelerometer noise density	σ_a	2.0×10^{-2} m/s^2/$\sqrt{\text{Hz}}$
Gyroscope noise density	σ_g	1.5×10^{-3} rad/s/$\sqrt{\text{Hz}}$
Accelerometer bias random walk	σ_{ba}	3.0×10^{-3} m/s^3/$\sqrt{\text{Hz}}$
Gyroscope bias random walk	σ_{bg}	2.0×10^{-5} rad/s^2/$\sqrt{\text{Hz}}$

Across all eight perturbations, the resulting ScanQA EM changes by at most 1.2% and the number of selected keyframes changes by at most 14.3%. This indicates that the cascaded filtering pipeline is robust to moderate threshold variations. We use the same threshold configuration ($\tau_d = 0.2\,\text{m}$, $\tau_\theta = 15^\circ$, $\tau_p = 15\,\text{px}$, $\tau_v = 0.4$) across all five benchmarks without per-dataset tuning, supporting cross-dataset generalization.

A.2 Egomotion Data Synthesis

Existing 3D scene understanding and spatial reasoning benchmarks lack raw IMU sensor data, as they were originally designed for vision-only or point-cloud-based methods. To enable training and evaluation with egomotion input, we synthesize realistic 6-axis IMU measurements (3-axis accelerometer and 3-axis gyroscope) from the per-frame camera pose annotations released by the source datasets. For each scan, we define a world coordinate frame whose origin coincides with the initial camera position at t_0 and whose Z-axis is anti-parallel to gravity. All poses and measurements are expressed relative to this frame, allowing the gravity vector to take the canonical form $\mathbf{g} = [0, 0, -9.81]^\top$ m/s^2. The initial camera's orientation $\mathbf{R}_0$ relative to this frame is derived from the dataset's gravity calibration.

Source Pose Origins. The four ScanNet-derived benchmarks (ScanQA, SQA3D, ScanRefer, Scan2Cap) provide per-frame camera-to-world transformation matrices $\mathbf{T}_i \in SE(3)$ at 30 FPS, which we use directly. VSI-Bench is sourced from the validation splits of three datasets and provides per-video metadata that links each clip back to its source scan (Yang et al., 2025), from which we retrieve the corresponding ground-truth poses: ScanNet poses come from offline RGB-D bundle adjustment (Dai et al., 2017), ScanNet++ poses from ARKit visual-inertial tracking with offline mesh alignment (Yeshwanth et al., 2023), and ARKitScenes poses from ARKit fused visual-inertial tracking (Baruch et al., 2021). All three release at metric scale and are routinely used as ground truth in 3D scene understanding tasks. The same B-spline differentiation procedure described below is applied across all five benchmarks.

B-spline Differentiation. Given the per-scan pose trajectory described above, we fit cubic B-splines (Lovegrove et al., 2013) to the discrete position trajectory $\mathbf{p}(t) \in \mathbb{R}^3$ and rotation trajectory $\mathbf{R}(t) \in SO(3)$ separately, yielding a continuous-time representation that allows analytical computation of derivatives at arbitrary time instants. The noise-free accelerometer and gyroscope readings in the body frame are:

$$\tilde{\mathbf{a}}(t) = \mathbf{R}(t)^\top \left(\ddot{\mathbf{p}}(t) - \mathbf{g}\right), \qquad \tilde{\boldsymbol{\omega}}(t) = \left(\mathbf{R}(t)^\top \dot{\mathbf{R}}(t)\right)^\vee, \tag{7}$$

where $\ddot{\mathbf{p}}(t)$ is the linear acceleration in the world frame obtained from the second derivative of the position spline, and $(\cdot)^\vee$ extracts the angular velocity vector from the skew-symmetric matrix $\mathbf{R}^\top\dot{\mathbf{R}}$. The $-\mathbf{g}$ and $\mathbf{R}(t)^\top$ terms explicitly apply gravity subtraction and the world-to-body rotation, so that $\tilde{\mathbf{a}}(t)$ and $\tilde{\boldsymbol{\omega}}(t)$ match the body-frame format of real MEMS IMU sensors.

Noise Model and Sampling Rate. To simulate realistic sensor characteristics, we augment the ideal readings with additive white noise and slowly drifting bias, following the standard continuous-time IMU noise model (Forster et al., 2016):

$$\begin{aligned} \mathbf{a}(t) &= \tilde{\mathbf{a}}(t) + \mathbf{b}_a(t) + \mathbf{n}_a, \qquad \mathbf{n}_a \sim \mathcal{N}(\mathbf{0},\, \sigma_a^2\mathbf{I}) \\ \boldsymbol{\omega}(t) &= \tilde{\boldsymbol{\omega}}(t) + \mathbf{b}_g(t) + \mathbf{n}_g, \qquad \mathbf{n}_g \sim \mathcal{N}(\mathbf{0},\, \sigma_g^2\mathbf{I}) \end{aligned} \tag{8}$$

where the biases $\mathbf{b}_a(t)$ and $\mathbf{b}_g(t)$ evolve as Brownian motions with $\dot{\mathbf{b}}_a \sim \mathcal{N}(\mathbf{0},\, \sigma_{ba}^2\mathbf{I})$ and $\dot{\mathbf{b}}_g \sim \mathcal{N}(\mathbf{0},\, \sigma_{bg}^2\mathbf{I})$. We use noise parameters representative of a consumer-grade MEMS IMU,

Table 9: **Real versus synthesized IMU on TUM-VI.** The same checkpoint is evaluated on the same 120 questions under the real BMI160 recordings and under a stream synthesized from TUM-VI's mocap trajectory. The three multiple-choice tasks are identical under both conditions and are omitted.

IMU source	Obj. Cnt.	Abs. Dist.	Room Size	Avg.
Real BMI160	54.6	39.2	55.0	49.0
Synthesized from mocap	54.8	39.8	55.2	49.1

Table 10: **Robustness to inertial noise, sampling rate and synchronization error.** The inertial stream is perturbed at inference, without retraining.

Condition	ScanQA EM	VSI Abs. Dist.	VSI Avg.
Nominal (200 Hz)	29.8	52.0	58.2
5× noise	29.5	49.2	57.5
10× noise	29.0	46.8	56.4
50 Hz sampling	29.7	50.2	57.9
±50 ms offset	29.8	50.5	58.1
±100 ms offset	29.7	49.9	57.8

as listed in Tab. 8. All IMU data is synthesized at 200 Hz, matching common consumer sensor sampling rates (Burri et al., 2016). These values follow the EuRoC (Burri et al., 2016) reference configuration, with the white-noise densities raised by a factor of ten. Allan-variance calibration is measured statically and understates in-motion noise, and TUM-VI applies the same inflation to its own recordings. Against the calibration TUM-VI reports for its Bosch BMI160, our accelerometer and gyroscope noise densities are 14× and 19× higher, and our bias random walks 35× and 9× higher.

Synthesis Fidelity. TUM-VI (Schubert et al., 2018) provides a mocap trajectory alongside its real BMI160 recordings, so we evaluate the same checkpoint on the same 120 questions under both inertial streams. As shown in Tab. 9, the two streams differ by 0.1 on average, so synthesis confers no measurable advantage over a real sensor.

Sensor Robustness. We further perturb the inertial stream at inference, without retraining, varying the noise level, the sampling rate and the video-IMU synchronization. As shown in Tab. 10, raising the noise to 10× nominal costs 1.8 on the VSI-Bench average, four-fold decimation to 50 Hz costs 0.3, and a 100 ms synchronization offset costs 0.4. The latter is roughly 19× the residual offset TUM-VI reports for its hardware-synchronized rig.

Integration Drift. The fusion path does not integrate the inertial stream. The motion encoder reads the raw readings of each inter-keyframe segment, and each motion token summarizes one segment with no state carried from earlier segments, so drift cannot accumulate in what the model receives. Integration occurs only in Stage 1 of the keyframe filter, which outputs a keep-or-discard decision rather than a measurement. A candidate it wrongly admits is still evaluated by Stages 2 and 3, neither of which uses the IMU. At 10× nominal noise the number of selected keyframes rises by 24%, while the VSI-Bench average falls by 1.8 (Tab. 10).

A.3 Training Data Statistics

We train `Motion-MLLM` on a mixture of four datasets, including ScanQA (Azuma et al., 2022), SQA3D (Ma et al., 2022), ScanRefer (Chen et al., 2020), and Scan2Cap (Chen et al., 2021), all derived from ScanNet (Dai et al., 2017) indoor scans. Tab. 11 summarizes the dataset composition. Following Zheng et al. (2025b), each training batch is randomly sampled from a single task type. VSI-Bench (Yang et al., 2025) is used only for evaluation, as it draws from additional data sources (ScanNet++ (Yeshwanth et al., 2023) and ARKitScenes (Baruch et al., 2021)) and lacks a public training split.

Table 11: **Training and evaluation data statistics.** All training sets are derived from ScanNet (Dai et al., 2017). †Scan2Cap reuses ScanRefer annotations with a different task formulation.

Dataset	Task	#Train	#Eval	#Scenes (train)
ScanQA (2022)	3D QA	25,563	4,675	562
SQA3D (2022)	Situated QA	26,623	3,519	518
ScanRefer (2020)	Visual Grounding	36,665	9,508	562
Scan2Cap† (2021)	Dense Captioning	36,665	9,508	562
Total (training)		**125,516**		
VSI-Bench (2025)	Spatial Reasoning	-	5,131	-

A.4 Real IMU Benchmark Construction

Source Data. We use the six indoor room sequences of TUM-VI (Schubert et al., 2018), which provides hardware-synchronized fisheye video and 200 Hz BMI160 IMU recordings. We undistort the fisheye images to pinhole-equivalent views using the calibration parameters provided by TUM-VI, producing a center-cropped pinhole video at approximately 80° effective field of view. The same undistortion is applied uniformly to all evaluated methods, removing fisheye distortion as a confounder.

Task Selection. Among VSI-Bench's eight tasks, we cover six (Obj. Cnt., Abs. Dist., Room Size, Rel. Dist., Rel. Dir., Appr. Order) and exclude two: Obj. Size, whose reliable manual annotation requires 3D bounding-box annotations not provided by TUM-VI, and Route Planning, whose route definitions are difficult to standardize across short room sequences. Question availability per (sequence, task) cell varies with scene content: tasks with multiple natural object pairs (Rel. Dist., Rel. Dir., Appr. Order) admit more questions per room, while Room Size has limited natural variation (one ground-truth dimension per room). We allocate 24 questions each for Obj. Cnt. and Abs. Dist. (4 per room), 12 for Room Size (2 per room), and 20 each for the three multiple-choice tasks, totaling 120 QA pairs.

Annotation Methodology. We follow the publicly released question templates of VSI-Bench (Yang et al., 2025). Question candidates are generated by filling templates with object pairs identified from each sequence, with optional vision-language model assistance for variety. Ground-truth answers are determined from TUM-VI's mocap-tracked camera trajectory (for distance and dimension queries) and from manual video inspection (for counting, direction, and ordering queries). All annotations are manually reviewed, and a second annotator independently verifies a 20% random subset to confirm reliability.

A.5 Prompts of `Motion-MLLM`

Fig. 4 shows the prompts used in `Motion-MLLM` for each task. We adopt the default system prompt of Qwen2.5-VL (Bai et al., 2025), "You are a helpful assistant," for all tasks. The user prompt contains video frames and IMU data as multimodal input, followed by a task-specific text instruction. All tasks use a unified `<answer>` tag format for response extraction.

For question answering (ScanQA (Azuma et al., 2022), SQA3D (Ma et al., 2022), and VSI-Bench (Yang et al., 2025)), the model is instructed to answer concisely. For SQA3D, the situation description is prepended to the question. For VSI-Bench, the prompt appends a type-specific template depending on the question type (multiple choice, numerical, or verbal). For visual grounding (ScanRefer (Chen et al., 2020)), the model outputs a JSON dictionary containing the frame index and an axis-aligned 3D bounding box in the predicted frame's camera coordinates. For IoU computation, the predicted box corners are transformed to the world frame via the camera-to-world pose of the predicted frame and re-axis-aligned, then compared against the world-frame ground-truth. For dense captioning (Scan2Cap (Chen et al., 2021)), the model describes the object at a given 3D location.

System Prompt: "You are a helpful assistant. You are provided with a video of a scene as a sequence of frames (<image><image>⋯ <image>) and the corresponding IMU sensor data (<imu>) recording the camera's egomotion."

3D Question Answering (ScanQA & SQA3D)

$\mathbf{X}_{\text{User}}$: <image><image>⋯ <image><imu>
What color is the couch near the window? Answer the question concisely within <answer></answer> tags.

$\mathbf{X}_{\text{Agent}}$: <answer> gray </answer>

Dense Captioning (ScanCap)

$\mathbf{X}_{\text{User}}$: <image><image>⋯ <image><imu>
Carefully watch the video and describe the object located at [1.35, -0.62, 0.78] in detail within <answer></answer> tags.

$\mathbf{X}_{\text{Agent}}$: <answer> A rectangular wooden in the center of the room, surrounding by four chairs on each side. </answer>

Spatial Reasoning (VSI-Bench)

$\mathbf{X}_{\text{User}}$: <image><image>⋯ <image><imu>
Question + {Type Template}

Type Template

- **MC**: "Answer with the option's letter from the given choices within <answer></answer> tags."
- **Numerical**: "Answer with the numerical value within <answer></answer> tags."
- **Verbal:** "Answer the question simply within <answer></answer> tags."

Visual Grounding (ScanRefer)

$\mathbf{X}_{\text{User}}$: <image><image>⋯ <image><imu>
Localize the object described below in the video.
Text: The black office chair in front of the desk near the window.
Output a JSON dictionary with the frame index in "frame" and the axis-aligned 3D bounding box in "bbox_3d" in the frame's coordinates.

$\mathbf{X}_{\text{Agent}}$:
```json
[
{"frame": 5. "bbox_3d": [-1.42, 0.37, 0.85, 0.62, 0.58, 0.93]},
...
]
```

Figure 4: **Prompts used for each task in `Motion-MLLM`.** All tasks share the same system prompt and receive video frames along with IMU data as input. Each task uses a task-specific user prompt with a unified `<answer>` tag format for response extraction.

Table 12: **Motion encoder design choice.** All variants share the same fusion module, training mixture and two-stage schedule, differing only in how each IMU segment is encoded. Latency is end-to-end inference time on ScanQA.

Motion encoder	Params	Latency	ScanQA EM	VSI Abs. Dist.	VSI Avg.
IMU preintegration + MLP	0.3M	0.57 s	27.8	46.5	55.1
1D temporal convolution	1.2M	0.58 s	29.1	50.0	57.5
Transformer encoder	2.0M	0.69 s	30.0	51.3	58.4
Bidirectional GRU (ours)	1.6M	0.61 s	29.8	52.0	58.2

B Analysis of the Egomotion Pathway

B.1 Motion Encoder Design Choice

Our motion encoder compresses each variable-length IMU segment into a single token, and several architectures can serve that role. As shown in Tab. 12, we compare our bidirectional GRU against a classical IMU preintegration summary followed by an MLP, a 1D temporal convolution, and a Transformer encoder, holding the rest of the model and the training schedule fixed. Learned sequence encoders outperform the hand-designed summary by a clear margin: preintegration reaches 55.1 on the VSI-Bench average, 3.1 below our GRU. Among the learned encoders the differences are small. The Transformer is 0.2 above our GRU, within the three-seed standard deviation, while adding 0.4M parameters and 0.08 s of latency. We retain the GRU because it consumes variable-length segments without padding or masking and its cost is linear in segment length, but the design does not depend on this choice.

Table 13: **Fusion design ablation.** Trained variants differ only in the fusion module. The final row is an inference-time intervention on the full model, restricting each visual token to attend only to the motion token of its own keyframe, with all parameters unchanged.

Fusion variant	Params	ScanQA EM	VSI Avg.
Trained variants			
Concat+MLP	95M	19.2	45.6
Single cross-attention layer	57M	26.1	50.5
Two layers, no vision-to-motion	73M	27.3	54.2
Full asymmetric fusion (ours)	110M	**29.8**	**58.2**
Inference-time intervention			
Cross-frame relay blocked	110M	27.9	53.8

Table 14: **Egomotion corruption controls.** The inertial stream is replaced at inference. Frame order, token count and positional encoding are unchanged in all three conditions.

Motion input	VSI Avg.	Room Size	Rel. Dist.	Appr. Order
No IMU (VGGT-only)	49.0	48.5	45.2	49.0
Zero	48.6	47.5	45.4	49.1
Shuffled across keyframes	44.2	43.7	41.1	46.0
Mismatched video	44.0	43.5	40.3	45.5
Full `Motion-MLLM`	**58.2**	**65.8**	**61.2**	**56.0**

B.2 Fusion Design Ablation

Tab. 13 decomposes what the fusion module contributes. Additional capacity alone does not explain the gain: Concat+MLP carries 95M parameters and reaches 19.2 on ScanQA, which is 7.4 below the configuration with no motion parameters at all, and it trails our near-parameter-matched design by 10.6. Within the module, adding a second attention layer while leaving motion tokens unmodified raises the VSI-Bench average by 3.7, and enabling the vision-to-motion pathway, which is what allows motion tokens to carry cross-frame context, adds a further 4.0. The specific mechanism therefore contributes more than depth alone.

We test the relay mechanism causally rather than by inspecting attention weights, since attention mass alone does not establish that information flows along a pathway. Restricting each visual token to attend only to the motion token of its own keyframe, with all parameters unchanged, costs 4.4 on the VSI-Bench average. The egomotion modality contributes 9.2 in total, so roughly half of that benefit is carried specifically by visual tokens reading motion tokens belonging to other keyframes.

B.3 Egomotion Corruption Controls

We test whether the model uses the physical content of the inertial stream or merely the presence of an additional token sequence. As shown in Tab. 14, we replace the stream at inference with three controls, leaving frame order, token count and positional encoding unchanged: an all-zero stream, a stream whose per-keyframe segments are permuted within the same video, and a stream taken from a different video. Zeroing the stream returns the model to the no-IMU baseline. It still receives its motion tokens at the correct temporal positions, still runs both fusion layers and still carries all 110M connector parameters, so an account in which the gain comes from an additional conditioning pathway rather than from egomotion would predict no such collapse.

Shuffling or mismatching the stream lands 4.8 to 5.0 below the no-IMU baseline. Wrong displacements attached to the right frames are worse than no displacements at all, which is what follows when the model composes geometry from the stream rather than reading it as a generic cue. Shuffling also preserves the multiset of motion tokens, and hence any aggregate statistic of the video's inertial magnitude, so a learned correlation between inertial magnitude and scene scale would survive it. Room size, the task such a correlation would most directly serve, instead falls from 65.8 to 43.7.

Table 15: **VGGT trajectory error by frame sampling.** Median absolute trajectory error over the 71 ScanNet validation scans used by ScanQA. The per-scan scale is fitted to ground truth and is not available to the model.

VGGT input	ATE, global scale	ATE, per-scan scale
Original video	1.2 m	0.1 m
Uniform 32 frames	2.2 m	0.51 m
Our 21 keyframes	2.5 m	0.62 m

Table 16: **Metric scale intervention.** The inertial stream is re-synthesized from camera trajectories scaled by α, with gravity held at 9.81 m/s^2 so the stream stays physical. The video is unchanged, so the two modalities disagree about scale.

α	Predicted distance, relative to $\alpha = 1$
0.5	0.62
1.0	1.00
2.0	1.70

Table 17: **VSI-Bench gain by task.** `Motion-MLLM` against the matched-protocol retrained VGGT-only baseline. Column order follows Tab. 2.

Configuration	Numerical Question				Multiple-Choice Question				Avg.
	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order	
VGGT-only (retrained)	67.5	35.5	63.3	48.5	45.2	48.3	34.7	49.0	49.0
`Motion-MLLM`	**72.6**	**52.0**	**66.4**	**65.8**	**61.2**	**55.8**	**35.8**	**56.0**	**58.2**
Δ	+5.1	+16.5	+3.1	+17.3	+16.0	+7.5	+1.1	+7.0	+9.2

B.4 Physical Grounding and Complementarity with VGGT

VGGT predicts camera pose, so it and the IMU supply related signals. Two things separate them. VGGT normalizes each scene during training, so its geometry is fixed only up to an unknown scale, while the accelerometer reports in m/s^2. Its accuracy also depends on inter-frame overlap: as shown in Tab. 15, per-scan trajectory error grows from 0.1 m on the original video to 0.62 m on our 21 keyframes, and uniform sampling at 32 frames already reaches 0.51 m. We then re-synthesize the IMU from trajectories scaled by α, leaving the video unchanged. As shown in Tab. 16, predicted distances track α with a log-log slope of 0.73, where a correlation between motion patterns and scene layouts would give a slope near zero. The slope falls below one because visual evidence continues to contribute. A linear probe on the frozen motion tokens recovers each segment's translation magnitude and rotation angle with R^2 of 0.89 and 0.93.

B.5 Task- and Source-Level Breakdowns

Tab. 17 breaks the VSI-Bench gain down by task. Gains concentrate where observations must be related across viewpoints and are smallest where one frame suffices: object size, a metric quantity answerable within a single frame, gains 3.1, while relative and absolute distance gain 16.0 and 16.5. Relative distance is a ranking task in which a global scale factor cancels, so the pattern is not explained by metric scale alone.

The gain is also consistent across the three sources VSI-Bench draws from, at $+9.4$ on ScanNet, $+8.9$ on ScanNet++ and $+9.3$ on ARKitScenes. Our training data comes from ScanNet training scans alone, so the two sources never seen in training gain as much as ScanNet does. The improvement is therefore not explained by familiarity with the training domain.

ScanQA provides no question-type taxonomy, so we partition its validation set by what the answer requires, keying on the ground-truth answer rather than the question wording, since a spatial phrase often only identifies the referent. As shown in Tab. 18, answers that are themselves a position gain 5.4, answers naming an object located through a spatial relation gain 3.4, and the remaining third

Table 18: **ScanQA validation set partitioned by answer type.** A question falls in *Others* if it asks for colour, shape or material. Otherwise it is *Location* if a ground-truth answer contains a spatial-relation term (left, right, front, behind, above, below, next to, between, across, opposite, facing, corner, centre, middle, side, along, against, near, far), *Object via spatial relation* if the question contains one, and *Others* otherwise. Containment terms such as *on* and *in* are excluded, since those relations are usually resolvable within a single frame.

Group	**#Questions**	**VGGT-only**	`Motion-MLLM`	**Δ**
Location	1,477	14.4	19.8	+5.4
Object via spatial relation	1,624	28.7	32.1	+3.4
Others	1,574	36.1	37.1	+1.0
Overall	**4,675**	**26.6**	**29.8**	**+3.2**

Table 19: **Full evaluation Results on ScanQA (Azuma et al., 2022).** We use the val set of ScanQA for evaluation. Reported metrics include EM-1, BLEU-1 to BLEU-4, METEOR, ROUGE-L, and CIDEr. "–" indicates the number is not available. "2D", "3D", and "M" specify the model's input type as 2D data (images/videos), 3D data (point clouds/depth maps), and egomotion data, respectively.

Methods	**Input**			**ScanQA (val)**							
	2D	3D	M	EM-1	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Task-Specific Models											
ScanQA (2022)		✓		21.1	**30.2**	**20.4**	**15.1**	10.1	13.1	33.3	64.9
3D-VisTA (2023)		✓		**22.4**	-	-	-	**10.4**	**13.9**	**35.7**	**69.6**
2D-Input Models											
Qwen2.5-VL-3B (2025)	✓			15.4	22.5	13.1	8.1	3.8	9.7	25.4	47.4
Qwen2.5-VL-7B (2025)	✓			19.0	27.8	13.6	6.3	3.0	11.4	29.3	53.9
LLaVA-Video-7B (2024b)	✓			-	39.7	26.6	9.3	3.1	17.7	44.6	88.7
VideoChat2 (2025a)	✓			19.2	-	-	-	9.6	9.5	28.2	49.2
Spatial-MLLM (2025)	✓			**26.3**	**44.4**	**28.8**	**21.9**	**14.8**	**18.4**	**45.0**	**91.8**
3D/2.5D-Input Models											
3D-LLM (2023)	✓	✓		20.5	39.3	25.2	18.4	12.0	14.5	35.7	69.4
Chat-Scene (2024)	✓	✓		21.6	43.2	29.1	20.6	14.3	18.0	41.6	87.7
LEO (2023)	✓	✓		24.5	-	-	-	13.2	20.0	49.2	24.5
GPT4Scene-HDM (2025)	✓	✓		28.2	44.4	30.3	22.3	15.5	18.9	46.5	96.3
Video-3D LLM (2025b)	✓	✓		30.1	**47.1**	**31.7**	**22.8**	16.2	19.8	49.0	102.1
LLaVA-3D (2024)	✓	✓		**30.6**	-	-	-	**16.4**	**20.8**	**49.6**	**103.1**
Egomotion-Input Model											
`Motion-MLLM`**-4B**	✓		✓	**29.8**	**46.5**	**31.2**	**22.2**	**15.8**	**20.3**	**48.0**	**100.4**

gains 1.0. Exact match is lowest for location answers because they are free-form phrases, so the models should be compared within a group rather than across groups.

C Additional Evaluation Results

C.1 Full Quantitative Results on ScanQA and SQA3D

Tab. 19 extends Tab. 1 with the complete metric suite on ScanQA (Azuma et al., 2022), additionally reporting EM-1, BLEU-2, and BLEU-3. The results are fully consistent with the main findings: `Motion-MLLM` achieves the best EM-1 among 2D-input models (29.8 vs. 26.3 for Spatial-MLLM) and remains comparable with top 3D-input models across all BLEU levels, confirming that the trends in Tab. 1 hold across the full metric set.

Tab. 20 provides a question-type breakdown on SQA3D (Ma et al., 2022). `Motion-MLLM` outperforms all 2D-input models across nearly all question types, with the largest gains on spatially-grounded and situational categories (i.e., *What*, *Is*, *Can*). Compared to 3D/2.5D-input models, `Motion-MLLM` achieves comparable or superior scores on most question types, demonstrating that lightweight egomotion data can match the spatial grounding capability of expensive 3D inputs. On *How* questions, which are typically quantitative count tasks, `Motion-MLLM` slightly underperforms the SOTA 2D-input baseline (Spatial-MLLM, -0.3), suggesting that egomotion cues are less decisive

Table 20: **Evaluation Results on SQA3D (Ma et al., 2022).** We use the test set of SQA3D for evaluation. We additionally report the average EM@1 for different question types, including *What*, *Is*, *How*, *Can*, *Which*, and *Others*. "–" indicates the number is not available. "2D", "3D", and "M" specify the model's input type as 2D data (images/videos), 3D data (point clouds/depth maps), and egomotion data, respectively.

Methods	Input			SQA3D (test)							
	2D	3D	M	What	Is	How	Can	Which	Others	Avg. (EM@1)	Avg. (EM@R1)
Task-Specific Models											
SQA3D (2022)		✓		31.6	**63.8**	**46.0**	69.5	43.9	45.3	46.6	-
3D-VisTA (2023)		✓		**34.8**	63.3	45.4	**69.8**	**47.2**	**48.1**	**48.5**	-
2D-Input Models											
Qwen2.5-VL-3B (2025)	✓			34.8	52.1	39.8	52.7	45.6	47.0	43.4	45.9
Qwen2.5-VL-7B (2025)	✓			39.7	56.6	41.1	55.9	47.6	47.2	46.5	49.8
LLaVA-Video-7B (2024b)	✓			42.7	56.3	47.5	55.3	50.1	47.2	48.5	-
Spatial-MLLM (2025)	✓			**45.9**	**71.6**	**55.1**	**69.5**	**52.0**	**53.0**	**55.9**	**58.7**
3D/2.5D-Input Models											
Chat-Scene (2024)	✓	✓		45.4	67.0	52.0	69.5	49.9	55.0	54.6	57.5
GPT4Scene-HDM (2025)	✓	✓		**55.9**	69.9	50.8	68.7	**53.3**	**60.4**	**60.6**	**63.3**
Video-3D LLM (2025b)	✓	✓		51.1	**72.4**	**55.5**	**69.8**	51.3	56.0	58.6	-
Egomotion-Input Model											
`Motion-MLLM`**-4B**	✓		✓	**54.1**	**72.0**	**54.8**	**70.3**	**53.0**	**58.9**	**60.2**	**62.4**

Table 21: **Full evaluation on ScanRefer and Scan2Cap.** For ScanRefer, scores include proposal refinement following SPAR (Zhang et al., 2025); "()" indicates the score without refinement.

Methods	Input			ScanRefer		Scan2Cap			
	2D	3D	M	Acc@0.25↑	Acc@0.5↑	C@0.5↑	B-4@0.5↑	M@0.5↑	R@0.5↑
Task-Specific Models									
ScanRefer (2020)		✓		37.3	24.3	-	-	-	-
Scan2Cap (2021)		✓		-	-	39.1	23.3	22.0	44.8
3D-VisTA (2023)		✓		**50.6**	**45.8**	**66.9**	**34.0**	-	-
2D-Input Models									
SPAR-7B (2025)	✓			48.8 (31.9)	43.1 (12.4)	-	-	-	-
VG LLM-4B (2025a)	✓			53.5 (36.4)	47.5 (11.8)	78.6	40.9	28.6	62.4
VG LLM-8B (2025a)	✓			**57.6 (41.6)**	**50.9 (14.9)**	**80.0**	**41.5**	**28.9**	**62.6**
2.5D/3D-Input Models									
Grounded 3D-LLM (2024d)		✓		47.9	44.1	70.2	35.0	-	-
GPT4Scene-HDM (2025)	✓	✓		**62.6**	**57.0**	-	40.6	-	59.3
LLaVA-3D (2024)	✓	✓		54.1	42.4	79.2	**41.1**	**30.2**	**63.4**
Video-3D LLM (2025b)	✓	✓		58.1	51.7	**80.0**	40.2	28.5	61.7
Egomotion-Input Models									
`Motion-MLLM`**-4B**	✓		✓	**61.4 (45.8)**	**55.3 (19.6)**	**79.0**	**41.6**	**30.0**	**64.0**

when visual information dominates. These per-type results are consistent with the conclusion in Sec. 4.2.

Answer Distribution. ScanQA and SQA3D have skewed answer distributions, and a language-only model fine-tuned on them recovers a substantial fraction of the reported scores. That floor is shared by our matched-protocol baseline in Tab. 6, which differs from the full model only in the egomotion pathway, so it cancels in the difference between them.

C.2 Full Quantitative Results on ScanRefer and Scan2Cap

Tab. 21 extends Tab. 4 with the full baseline set.

Visual Grounding on ScanRefer. ScanRefer (Chen et al., 2020) contains $36,665$ object descriptions paired with axis-aligned bounding boxes across 562 indoor scans. We follow the standard protocol with Acc@0.25 and Acc@0.5, the percentage of samples whose IoU exceeds the threshold. As shown in Tab. 21, `Motion-MLLM` achieves Acc@0.25/0.5 of $61.4/55.3$ (refined) and $45.8/19.6$

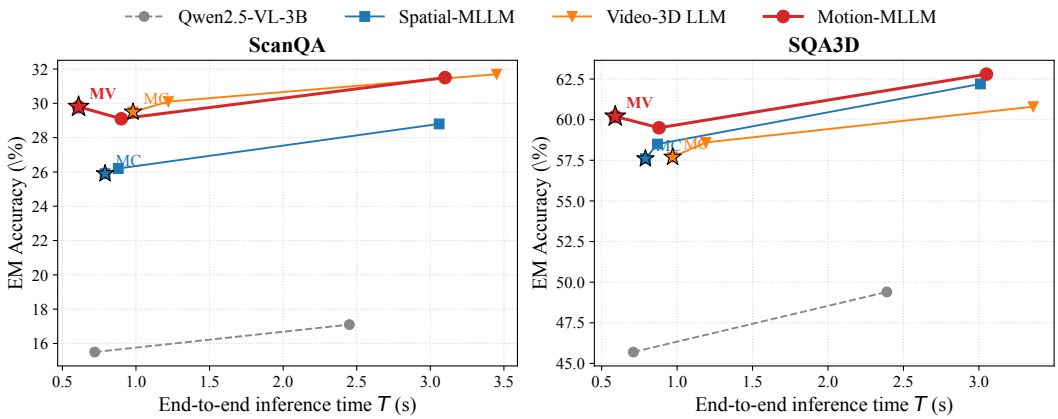


Figure 5: **Latency-accuracy trade-off on ScanQA (left) and SQA3D (right).** Each baseline polyline traces its frame-sampling configurations. `Motion-MLLM` uses MV-filtering.

(raw) on the ScanRefer benchmark with only ∼4B parameters, without requiring external 3D data such as point clouds or depth maps. It significantly outperforms the SOTA 2D-input baseline (VG LLM-8B (Zheng et al., 2025a)), surpassing its refined scores by $+3.8$ and $+4.4$, respectively. More importantly, `Motion-MLLM` rivals and even outperforms methods relying on heavy explicit 3D inputs, such as LLaVA-3D (54.1 Acc@0.25) and Video-3D LLM (58.1 Acc@0.25). This gain highlights the effectiveness of our asymmetric cross-modal fusion: motion features ground visual content in physical space, allowing `Motion-MLLM` to infer distances and sizes from egomotion cues without depth maps or point clouds.

Dense Captioning on Scan2Cap. We evaluate `Motion-MLLM` on the val split of Scan2Cap (Chen et al., 2021), which contains 9,508 object descriptions across 141 indoor scans. We report CIDEr (C), BLEU-4 (B-4), METEOR (M), and ROUGE (R) scores at IoU ≥ 0.5 against the ground truth. As shown in Tab. 21, `Motion-MLLM` achieves a B-4@0.5 score of 41.6 and an R@0.5 score of 64.0, setting a new best across all baselines, including those utilizing explicit 3D data. Its C@0.5 (79.0) and M@0.5 (30.0) are competitive, nearly matching top-performing 3D-input models such as Video-3D LLM (Zheng et al., 2025b) and LLaVA-3D (Zhu et al., 2024). The improvement is smaller than on visual grounding, as dense captioning relies less on absolute scale estimation. However, gains on B-4, M, and R over 2D-input baselines demonstrate that leveraging the egomotion modality enables better object localization and spatial reasoning, yielding more precise and contextually grounded captions.

C.3 Pareto Analysis of Cost-Effectiveness

The CE ratio used in Sec. 4.4 is a single-number summary that implicitly weighs accuracy against latency. Fig. 5 reports the full latency-accuracy trade-off, with each baseline shown as a polyline through its sampling configurations (uniform 128, uniform 32, and adaptive MC where applicable). `Motion-MLLM`'s MV-filtering point sits at the low-latency end of the Pareto frontier on both benchmarks, strictly dominating every baseline at its best CE point. Baselines reach higher peak accuracy only at $\sim 5\times$ `Motion-MLLM`'s latency, confirming that the efficiency advantage in Sec. 4.4 is a genuine latency-accuracy property.

C.4 Qualitative Results

We provide qualitative examples to visualize the evaluation results of `Motion-MLLM` on ScanQA (Azuma et al., 2022) (Fig.6), SQA3D (Ma et al., 2022) (Fig.7), VSI-Bench (Yang et al., 2025) (Fig.8), ScanRefer (Chen et al., 2020) (Fig. 9), and Scan2Cap (Chen et al., 2021) (Fig. 10). These examples show that `Motion-MLLM` correctly answers spatial questions, localizes described objects, and generates grounded captions across diverse indoor scenes.

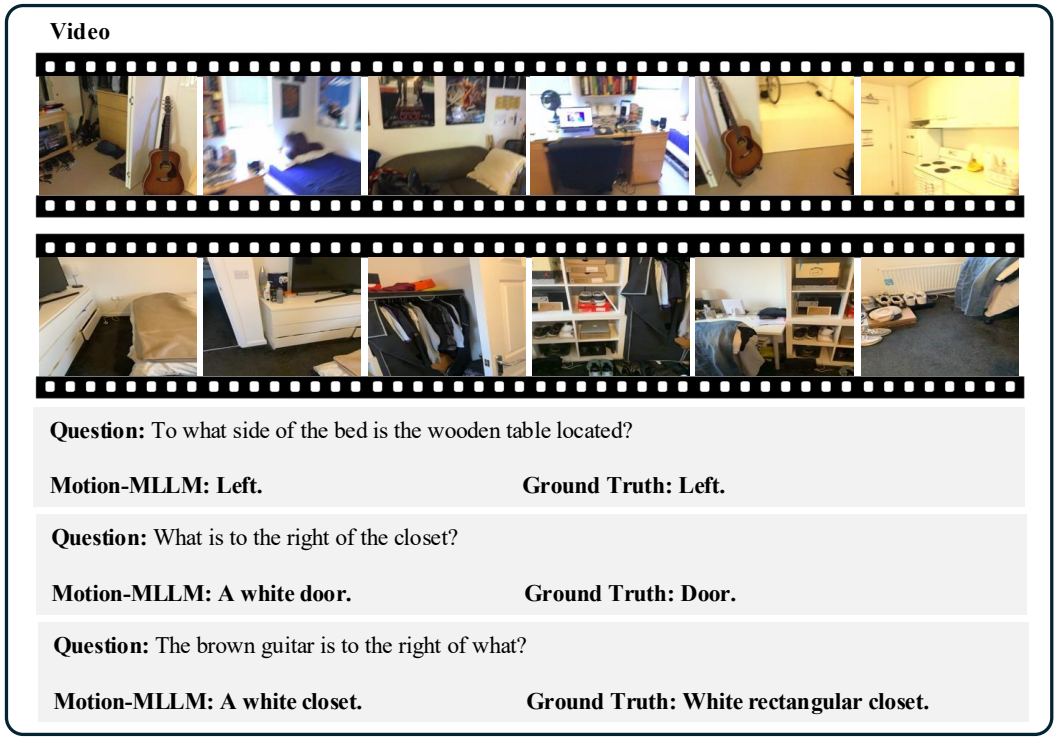


Figure 6: **Qualitative examples on ScanQA (Azuma et al., 2022).**

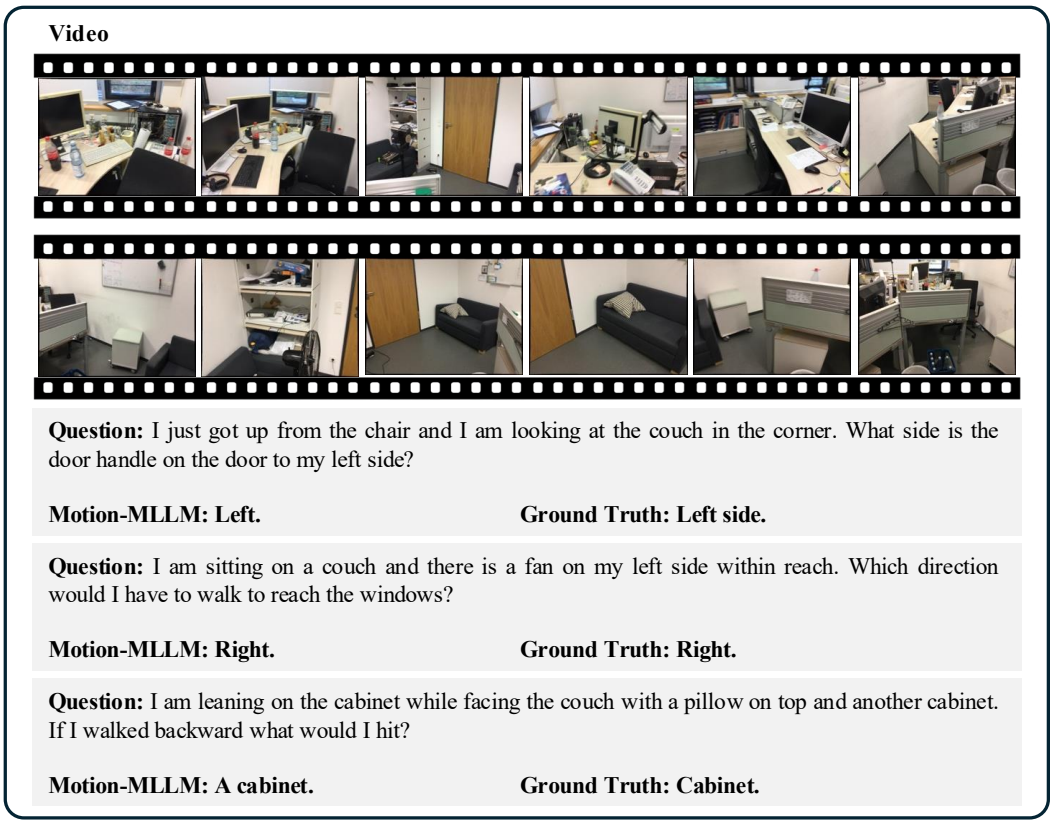


Figure 7: **Qualitative examples on SQA3D (Ma et al., 2022).**

Video



Question: You are a robot beginning at the printer and facing the window. You want to navigate to the lamp. You will perform the following actions (Note: for each [please fill in], choose either 'turn back,' 'turn left,' or 'turn right.'): 1. Go forward until the desk chair. 2. [Please fill in] 3. Go forward until the sofa. 4. [please fill in] 5. Go forward until the other sofa by the window. 6. [please fill in] 7. Go forward until the lamp. You have reached the final destination.

A. Turn Right, Turn Left, Turn Right
B. Turn Left, Turn Left, Turn Right
C. Turn Back, Turn Right, Turn Left
D. Turn Back, Turn Right, Turn Right

Motion-MLLM: A **Ground Truth:** A

Question: What is the size of this room (in square meters)? If multiple rooms are shown, estimate the size of the combined space.

Motion-MLLM: 24.5 **Ground Truth: 26.4**

Video



Question: If I am standing by the tv and facing the bed, is the table to my front-left, front-right, back-left, or back-right? The directions refer to the quadrants of a Cartesian plane (if I am standing at the origin and facing along the positive y-axis).

A. Back Left
B. Back Right
C. Front Right
D. Front Left

Motion-MLLM: C **Ground Truth: C**

Question: Measuring from the closest point of each object, what is the distance between the table and the bed (in meters)?

Motion-MLLM: 2.05 **Ground Truth: 1.9**

Figure 8: **Qualitative examples on VSI-Bench (Yang et al., 2025).**

Video



Question: The recycling bin has a kitchen cabinet to its right. the recycling bin is boxy and blue. Please locate its position with the coordinate of center x, y, z and its length, width and height.

Motion-MLLM: [1.25, 1.51, 0.32, 0.54, 0.36, 0.85]
Ground Truth: [1.48, 1.64, 0.41, 0.57, 0.32, 0.77]

Question: The object is a small black microwave oven situated under a shelf . It looks to have a digital timer. Please locate its position with the coordinate of center x, y, z and its length, width and height.

Motion-MLLM: [1.60, -1.20, 0.89, 0.40, 0.43, 0.32]
Ground Truth: [1.54, -1.06, 1.00, 0.44, 0.51, 0.30]

Figure 9: **Qualitative examples of visual grounding on ScanRefer (Chen et al., 2020).**

Video



Question: Carefully watch the video and describe the object located at [-0.95, 0.48, 0.12] in detail.

Motion-MLLM: A blue trash bin is located besides a white heater. It is in front of a white table.
Ground Truth: A blue trash bin. It is under a white table.

Question: Carefully watch the video and describe the object located at [1.28, -1.38, 0.67] in detail.

Motion-MLLM: A white dresser is located against the wall on the right side of the bed, and it is located to the left of the white door.
Ground Truth: A white dresser is between the wall and the bed. It is in the right corner.

Figure 10: **Qualitative examples of dense captioning on Scan2Cap (Chen et al., 2021).**