4DP-QA: Scalable QA for 4D Perception in Vision Language Models

Seokju Cho[1,3*] Abhishek Badki[1] Hang Su[1] Jindong Jiang[1] Ziyao Zeng[1,2*]
Seungryong Kim[3] Sifei Liu[1] Orazio Gallo[1]

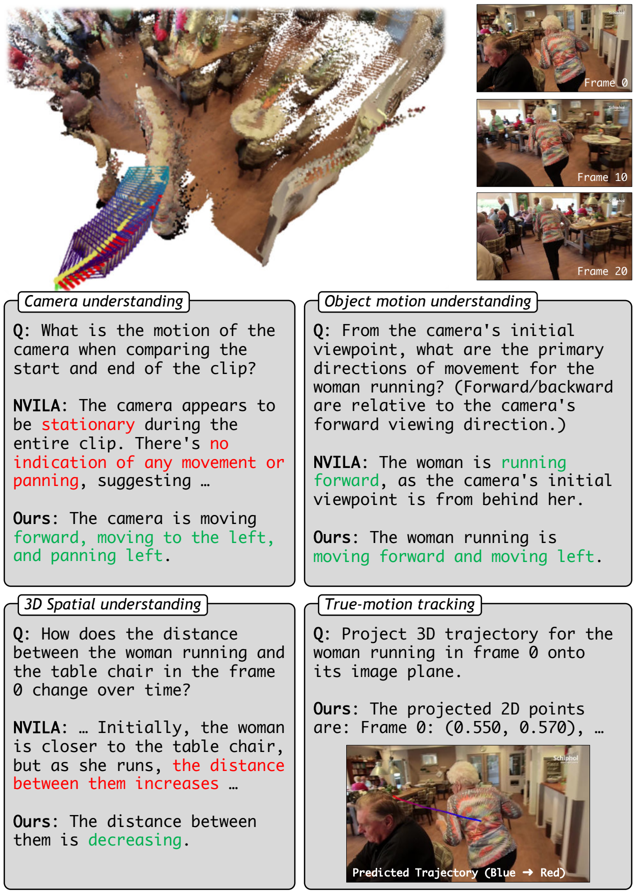
[1]NVIDIA [2]Yale University [3]KAIST AI

Abstract

Despite recent advances, Vision Language Models (VLMs) still struggle to grasp the dynamics of the world. We note that the ability to reason about a 4D scene, challenging in itself, is further complicated by two factors. First, VLMs observe motion indirectly via its projection onto 2D images. Second, existing datasets fail to disentangle object and camera motion. To address these challenges, we present a QA generation pipeline that focuses on motion-related scene understanding. We take particular care of the entanglement of camera and object motion by casting tracking in both the traditional way and in a novel, fixed reference system, dubbed True-Motion Tracking, which provides an intuitive description of motion. From this pipeline, we generate a large-scale training dataset of 400K samples, 4DP-QA (4D Perception QA), and a 2.2K-sample benchmark, 4DP-QA-Bench. Training existing models on our dataset yields performance improvements on an external benchmark, validating the effectiveness of our method.

1. Introduction

Vision Language Models (VLMs) [67] have made great progress in semantic reasoning with images and videos [6, 11, 40, 42, 56], and 3D scene understanding [4, 10, 12, 43]. However, their grasp of motion, ubiquitous in the physical world, remains limited. This is in part rooted in the video capture process itself: the world is 4D (3D+motion), but it is projected onto the 2D sensor of a camera, which is generally also moving. As a result, depth cues are discarded and, even more critically, the absolute motion of objects is entangled with that of the camera, making 4D understanding challenging. Compounding this, most training datasets focus on understanding the apparent motion (motion in the 2D image plane), rather than the true 3D motion. In this work, we address this limitation by introducing a comprehensive framework to equip VLMs with better 4D understanding.



Q: What is the motion of the camera when comparing the start and end of the clip?

NVILA: The camera appears to be stationary during the entire clip. There's no indication of any movement or panning, suggesting …

Ours: The camera is moving forward, moving to the left, and panning left.

Q: From the camera's initial viewpoint, what are the primary directions of movement for the woman running? (Forward/backward are relative to the camera's forward viewing direction.)

NVILA: The woman is running forward, as the camera's initial viewpoint is from behind her.

Ours: The woman running is moving forward and moving left.

Q: How does the distance between the woman running and the table chair in the frame 0 change over time?

NVILA: … Initially, the woman is closer to the table chair, but as she runs, the distance between them increases …

Ours: The distance between them is decreasing.

Q: Project 3D trajectory for the woman running in frame 0 onto its image plane.

Ours: The projected 2D points are: Frame 0: (0.550, 0.570), …

Figure 1. Our framework equips VLMs with better 4D understanding for in-the-wild videos. Training a state-of-the-art VLM (NVILA [42]) on our dataset yields performance gains (NVILA vs. Ours). We also introduce *true-motion point tracking*, a new capability that enables the VLM to isolate true object motion from camera movement, leading to better 4D understanding.

Large models exhibit strong *scaling properties* [30, 47], which allow the *emergence* of new capabilities. Our intuition is that 4D understanding will similarly emerge, even in largely standard VLM architectures, given the right *qual-*

---

[*] Work done during internship at NVIDIA.

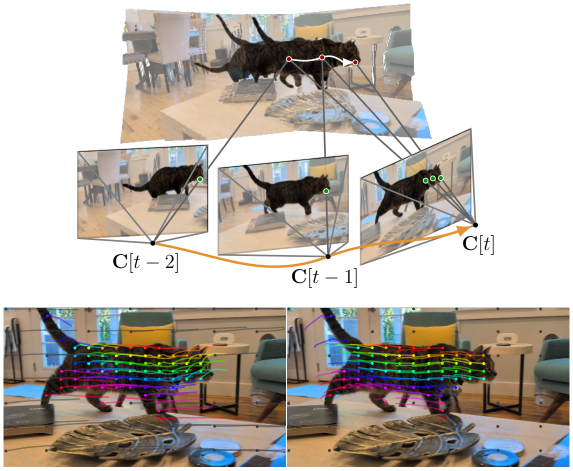
ity and *quantity* of training data. To achieve this, we introduce a scalable spatio-temporal QA generation pipeline that systematically analyzes scene dynamics using accurate geometric information (camera poses, depths, 6D object poses, *etc*.) from multiple data sources, both real and synthetic. We use this pipeline to construct QA pairs that probe different aspects of scene dynamics, such as camera motion, object motion, inter-object dynamics, and spatial reasoning. Our approach employs carefully designed heuristics to translate continuous geometric measurements into natural language descriptions tailored to the different question types.

While these natural language QA pairs provide rich semantic supervision, coarse language descriptions alone are insufficient for fine-grained 4D understanding. To address this, we develop QA pairs aimed at more fine-grained and low-level perceptual abilities. One such task is *visual point tracking*, that is, following points [19, 31] across video frames, crucial for a range of applications like 3D reconstruction [53] and robotics [8, 58]. Visual point tracking allows us to visually associate dynamic regions across frames, but the tracks do not always give a meaningful signal about the *true motion* of the object when the camera is moving. For instance, in Figure 2 the camera is moving to the right faster than the cat. As a result, the visual point tracks (Figure 2(a)) show points on the cat moving backward.

To address this, we introduce a new perceptual task: estimating an object's motion as if it were observed from a stationary reference system, such as the viewpoint of a static camera. Under this formulation, the object's motion is disentangled from the camera's motion, providing a more intuitive representation: now the tracks show the cat moving forward, see Figure 2(b). We term this capability *true-motion point tracking*. Visual point tracking encourages the model to capture dense motion correspondences tied to appearance changes, while true-motion point tracking teaches it to reason about object motion in a stable, fixed reference system. We incorporate these fundamental perceptual tasks into our dataset.

Our work makes the following contributions:

- We introduce a scalable spatio-temporal QA generation pipeline that automatically produces high-quality 4D reasoning data, yielding a training dataset with 400K QA pairs (4DP-QA), and a benchmark with 2.2K QA pairs (4DP-QA-Bench).
- We train multiple VLMs on our training dataset and show improved performance on both ours and external 4D reasoning benchmarks.
- We introduce true-motion point tracking, a new low-level perceptual task that provides an intuitive, image-aligned motion representation. To enable VLMs to learn this task, we develop dedicated QA pairs covering both true-motion and visual point tracking.
- We show that incorporating the point tracking tasks into our training data enhances the spatio-temporal reasoning abilities of VLMs on external benchmarks, underscoring the value of fine-grained perceptual supervision.



(a) Visual Point Tracking (b) True-motion Point Tracking

Figure 2. **True-motion Point Tracking.** Visual point tracking (a) only captures the apparent motion of the object, here making the cat appear to move backward due to the rightward camera motion. True-motion point tracking (b) disentangles camera and object motion, showing the cat moving forward. Background tracks (gray points) show movement in (a) but remain stationary in (b), highlighting that true-motion tracks reflect actual object motion as seen from a fixed viewpoint.

2. Related Work

This work investigates architectural and data strategies to equip VLMs with 3D perception and motion understanding, including robust capabilities for visual and motion-based point tracking. In the following, we review previous work on relevant models, datasets, and point tracking.

General VLMs. The foundation for 3D/4D VLMs rests on 2D models. Early approaches like Flamingo [2] and BLIP-2 [36] focused on aligning visual encoders with LLMs. Subsequent work moved toward instruction tuning [40] and scaling [6, 11, 13, 39]. Video VLMs extend 2D VLMs to temporal understanding, with rapid advances in training strategies [56], computational efficiency [29, 35], and long-range reasoning [27, 61]. Research is also converging on integrated 4D models [69, 70] that fuse video with 3D awareness for physical-world reasoning.

3D VLMs. 3D VLMs aim to equip LLMs with 3D spatial understanding. We focus on scene-level VLMs rather than those operating on raw 3D data like point clouds [46, 63] or meshes [22, 57], which primarily address shape-level geometry and not complex, holistic scene reasoning. These approaches often integrate 3D information via alignment, such as depth map projection [14] and 3D position embeddings [15, 72]. VLM-3R [21] bypasses the reliance on explicit 3D data and adopts CUT3R [55] for 3D reconstructive tokenization. SpatialVLM [10] demonstrates that quantitative spatial reasoning is achievable with only 2D inputs, enabled by a massive, synthesized 2B-sample VQA dataset.

Spatial-temporal understanding datasets. Compared to datasets designed for general image and video understanding [33, 41] or large-scale vision-language instruction tuning [34, 39], datasets available for training and benchmarking VLMs' spatial-temporal capabilities are substantially more limited in scope or scale. Static 3D datasets [4, 12, 43, 68] lack dynamics beyond passive camera motion. General video datasets [24, 54, 65] address scene dynamics and temporal reasoning but neglect explicit 3D geometry and true-motion disentanglement necessary for robust 4D perception. Works targeting dynamic scenes with 3D annotations are either limited to benchmarking scale [37, 71], restricted in temporal complexity [48], or constrained to limited contexts like automotive scenes [52]. ST-VLM [32], a concurrent work, focuses on general scenes but is limited in scale compared to our work. We propose a data pipeline to create a large-scale dataset of fine-grained spatial-temporal QA pairs to overcome these limitations.

Point Tracking. Point tracking has evolved from 2D correspondences [16, 19, 20, 31] to 3D tracking [17, 59] and more recently towards full 4D understanding with joint modeling of camera and object dynamics [60, 66]. While 2D methods capture apparent motion and 3D methods add depth awareness, only 4D approaches aim to recover true object motion by disentangling camera and object motion. However, these solutions are highly specialized, making it challenging to integrate them into general-purpose VLMs.

3. 4D Understanding with VLMs: A Data-Driven Approach

We present a data-driven approach to equip VLMs with fine-grained 4D perception capabilities. We first introduce the perceptual task of true-motion point tracking (Section 3.1), which later serves as a key component in defining novel QA pairs. We then introduce a scalable QA generation pipeline (Section 3.2), with which we create a large-scale 4D understanding QA dataset (Section 3.3) for training and evaluating VLMs.

3.1. Preliminaries: True-Motion Point Tracking

For a 3D point track $\{\mathbf{X}[t]\}$ over a time window $[0, T)$ and a moving camera with extrinsics $\{\mathtt{T}[t]\}$ and intrinsics $\mathtt{K}$, there are two ways to image the 3D point track in 2D:

$$P_{2D} = \{\mathbf{p}[t]\}_{t\in[0,T)} = \{\Pi(\mathtt{K}, \mathtt{T}[t], \mathbf{X}(t))\}_{t\in[0,T)}, \quad (1)$$

$$M_{2D} = \{\mathbf{m}_{t_q}[t]\}_{t\in[0,T)} = \{\Pi(\mathtt{K}, \mathtt{T}[t_q], \mathbf{X}(t))\}_{t\in[0,T)}, \quad (2)$$

where Π is the image projection operator, and t_q is a fixed reference time. Equation 1 gives us the 2D visual point track P_{2D}, while Equation 2 defines what we term the *true-motion* point track M_{2D}. The difference between them is that P_{2D} is imaged by a changing camera, thus entangling the object and the camera motion, while M_{2D} is imaged by a fixed reference system of the camera at time t_q, thus better capturing the true motion of the object. They become equivalent when the camera is stationary, since $\mathtt{T}[t] = \mathtt{T}[t_q]$.

Table 1. **Data sources for the dataset generation pipeline.** We collect data from a variety of sources, including driving, indoor, and simulation datasets, spanning synthetic and real-world scenes. Once preprocessed, they are standardized to a common format that our generation pipeline (Section 3.2) can use to produce QA pairs.

Dataset	Type	Domain	Number of	
			Videos	Frames
SHIFT [50]	Driving	Synthetic	3,200	1.6M
Virtual KITTI 2 [9]	Driving	Synthetic	50	21.3K
Aria Digital Twin [44]	Indoor, Egocentric	Real	273	644K
HOT3D [7]	Indoor, Egocentric	Real	424	1M
Kubric [26]	Simulation	Synthetic	9.7K	320K
Total			13.6K	**3.6M**

Given a 2D query pixel $\mathbf{p}[t_q]$ at time t_q (where $\mathbf{p}[t_q] = \mathbf{m}_{t_q}[t_q]$), the tasks of visual point tracking and true-motion point tracking are to predict the whole sequences P_{2D} and M_{2D}, respectively. Visual point tracking can be solved using correspondences by comparing the features of $\mathbf{p}[t_q]$ with pixels at other time in the image sequence [19]. On the other hand, to estimate the true-motion point track, one needs to not only solve the visual correspondence task but also recover depth and camera poses. In return, the true-motion track gives an intuitive image-aligned representation that allows reasoning about the dynamic motion in context of its static surrounding from a fixed desired reference system, as shown in Figure 2. We argue that the two are complementary and both are useful for understanding scene dynamics.

Visual point tracking is a well-studied task with specialized solutions [16, 31]. Likewise, true-motion point tracks can be estimated with modern 3D reconstruction methods by explicitly reconstructing 3D point tracks [5, 23, 60, 66]. Integrating these solutions into VLMs requires careful design and brings additional overhead and complexity. Instead, we formulate both tracking tasks as QA pairs and design a large-scale QA generation pipeline to investigate VLMs' capabilities to solve these tasks directly.

3.2. Dataset Generation Pipeline

We design a scalable pipeline to generate 3D motion QA pairs from a wide range of data sources, including auto-

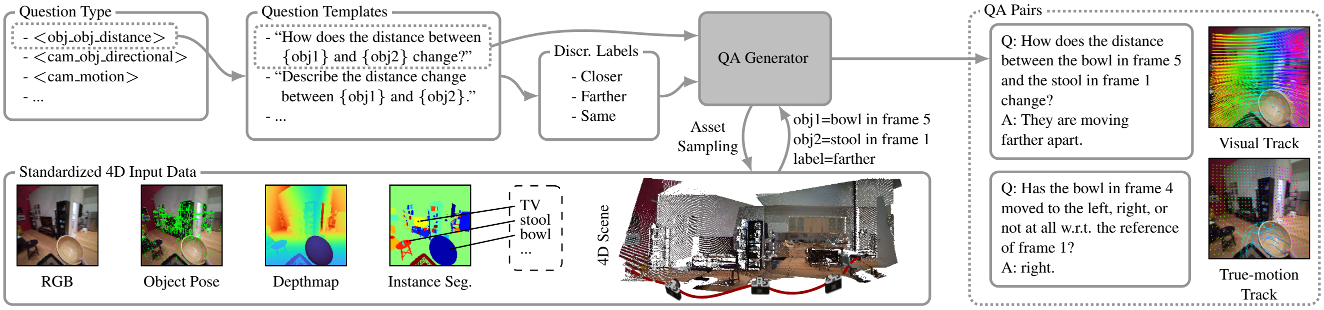


Figure 3. **Dataset Generation Pipeline (Section 3.2).** The pipeline takes as input standardized 4D input data, and produces QA pairs for 13 question types. Each QA pair is generated by instantiating a pre-defined template with the sampled assets and either their discrete labels (for descriptive questions) or continuous measurements (for visual and true-motion point tracking).

motive, physics simulation, and indoor datasets featuring human-object interactions (Table 1). To be consumed by the generation pipeline, each data source is first standardized to a common format of coordinates, image resolution, frame rates, camera parameters, depth, segmentation, metadata, and 6D object poses. 3D point tracks required for many of the QA pairs are extracted from the provided depth maps, camera poses, and 6D object poses. Object labels, when available, were insufficient to uniquely identify objects, so we use a VLM [6] to caption each object using its bounding box and initial object label. After preprocessing, our QA generation pipeline proceeds through the components outlined below. See Figure 3 for an overview. Our pipeline is readily extendable to arbitrary video sources by utilizing off-the-shelf 4D reconstruction methods [38, 55].

QA generator. This is the core component that produces QA pairs for each question type. It first initializes QA pairs by sampling from a set of pre-defined templates. We design multiple templates for each question type and use an off-the-shelf LLM (Gemini-2.5-Pro [18]) to produce diverse phrasing for both questions and answers. Each template contains slots for object references, time instances, coordinate systems, as well as slots in the answers for discrete labels (for descriptive questions) or continuous measurements (for visual and true-motion point tracking). The QA generator then informs the asset sampling process about certain desired characteristics of the assets (*e.g.*, there should be $\geq$ 2 moving objects) for the sampled template. Last, we finalize QA pairs by filling in the slots with the returned assets that meet the requirements.

Assets sampling. Not all video segments display meaningful or sufficient motion for the targeted question types. We first sample random video segments, then assess each segment for suitability according to the specified requirements. To do so, we design a set of heuristics and dataset-specific thresholds to determine whether the camera and objects display the desired characteristics. Depending on the question, it may be necessary for the camera to remain stationary or move in a particular manner. Similarly, objects are filtered based on their motion characteristics (*e.g.*, stationary, moving, changing direction) and the degree to which they are visible within the video segment. We then sample from this filtered object list, discarding the segment if there are insufficient eligible objects for the target question. For certain question types, we also need to choose the appropriate reference system, such as a specific camera coordinate system, the object's canonical reference system or screen-space coordinate system, in order to identify the selected objects. Finally, we sample how to refer to the objects in the questions: by object caption, by coordinate (pixel (x, y) in a specific frame), or by visual region annotation (circle around the region of interest [49]).

Discrete label generation. For question types requiring coarse descriptive answers, care needs to be taken to map the continuous geometry measurements like distances, translations, rotations, directions, *etc*. computed in a specific reference system into categorical labels. For example, 3D translation may be described using combinations of (forward/backward, left/right, up/down), 3D rotation using combinations of (pan left/right, tilt up/down, roll clockwise/counterclockwise), 3D distances using categories (increasing/decreasing/constant); similar descriptors are used for 2D screen-space motions (see supplementary). We carefully designed the thresholds for each data source to ensure that the selected labels for questions are accurate, unambiguous, and aligned with human perceptual judgments. For questions surrounding the perceptual capabilities of visual and true-motion point tracking, we directly pass the continuous values to the QA generator.

Finally, we perform multiple rounds of human validation to ensure high quality for the QA pairs in our benchmark.

3.3. The 4DP-QA Dataset

With the scalable QA generation pipeline, we construct our dataset, 4DP-QA, with the goal of equipping standard

VLMs [6, 42] with fine-grained 3D motion understanding across a variety of scenes. The QA pairs cover perceptual tasks like visual and true-motion point tracking as well as a broad array of 3D motion understanding challenges—such as camera motion, object motion, and spatial reasoning. **Key characteristics** of 4DP-QA include: (1) consistent use of relative scales to support generalization across diverse scenes, (2) multiple modes of object reference (visual prompts, coordinates, and captions), (3) deliberate use of different reference systems (including camera, object, screen-space, and gravity-aligned) in formulating questions, (4) varied answer formats (free-form descriptive, multiple-choice, and Y/N responses).

4DP-QA includes 13 question types that can be organized into 4 categories as described below. We describe the high-level dataset taxonomy in this section and provide additional details in Section 4.1 and the supplementary.

I. Camera Motion. This category concerns the motion of the camera, as well as how the camera's motion affects its relation to objects. Question types include: (1) *Camera Movement*: how the camera translates and rotates over time. (2) *Camera-Object Distance Change*: how the distance between the camera and an object changes.

II. Object Motion. This category concerns the motion of objects relative to the camera or other objects. Question types include: (3) *Rotation*: how an object rotates from a top-down view. (4) *Direction*: how an object translates relative to the first frame's viewpoint. (5) *Agent Motion*: how an agent (*e.g.*, person or car) moves w.r.t. its own reference system. (6) *Moved Distance*: comparing the distances traveled by two objects. (7) *Object-Object Distance Change*: how the distance between two objects changes over time.

III. 3D Spatial Understanding. This category concerns the understanding of 3D spatial relationships between objects as well as with respect to the camera. The multi-view questions (10 and 11) introduce further complexity, requiring joint reasoning about object motion and depth variation across time. Question types include: (8) *Depth*: comparing the distances of objects from the camera. (9) *Object-Object Distance*: comparing the distances of objects from a reference object within the same frame. (10) *Multi-View Depth*: similar to 8, but with reference to objects in different frames. (11) *Multi-View Object-Object Distance*: similar to 9, but with reference to objects in different frames.

IV. Point Tracking. These questions require recovering the tracks of (12) *Visual Point Tracking*, P_{2D} (Equation 1), and (13) *True-Motion Point Tracking*, M_{2D} (Equation 2), given a query point in a reference frame. In the question, we also specify the target frames (which can be a single or multiple frames) for which we need to recover the track. The answer includes the coordinates and an occlusion flag indicating the visibility of the point in each target frame.

4. Implementation

We present key dataset and training details below, with further information in the supplementary.

4.1. Dataset Details

Using our proposed dataset pipeline, we construct two datasets: the large-scale 4DP-QA dataset for training and a standalone 4DP-QA-Bench dataset for benchmarking.

4DP-QA. For training, we generate 400K QA pairs derived from videos totaling 3.3M frames. We sample 32 frames per video segment and use 448×448 resolution. The dataset contains diverse answer formats: multiple-choice, Y/N, free-form, and point tracking trajectories. Objects are referenced via visual prompts (circles [49]), pixel coordinates, or captions. Tracking coordinates are normalized to $[0, 1]$ and rounded to three decimals. To prevent shortcut learning from autoregressive extrapolation, we randomize the order of trajectory frames in outputs and include frame indices in queries to specify which frames to track.

4DP-QA-Bench. For benchmarking, we generate 2.2K QA pairs from 317K held-out test frames, covering camera motion, object motion, and 3D spatial understanding. Questions are formatted as either multiple-choice with four options or binary Y/N. We evaluate model responses using exact string matching following [65] and report accuracy as the final metric. We provide additional QA quality and bias analysis in the supplementary.

4.2. Training Details

We study two architectures: (1) standard VLMs (NVILA-Lite-8B [39], Qwen2.5-VL-3B/7B [6]) to assess our dataset, and (2) a 4D VLM with an integrated geometry encoder [5] for injecting pre-trained geometric features, inspired by recent 3D VLMs [21].

Standard VLMs are trained with a batch size of 128 for 1 epoch, corresponding to 3.1K training iterations. For NVILA-Lite-8B [42], we use a learning rate of 2×10^{-5} and AdamW optimizer with a cosine learning rate scheduler. Qwen-VL [6, 64] variants are trained using the same hyperparameters, except with a lower learning rate of 1×10^{-5}. We freeze the visual encoder during training. Training of each model takes approximately 9 hours on 32 NVIDIA A100 GPUs with 4-step gradient accumulation.

For our 4D VLM, we integrate L4P [5], a general-purpose ViT-based geometry encoder pre-trained on various low-level 4D perception tasks (such as depth and optical flow estimation, and 2D/3D tracking), to inject geometric features directly into the VLM backbone (NVILA-Lite-8B). We extend L4P by incorporating the true-motion point tracking task. L4P features are projected via an MLP and interleaved with visual tokens from the image encoder.

Table 2. **Quantitative comparison of VLMs on our 4DP-QA-Bench.** We report the accuracy (%) for each task. Alongside comparisons with off-the-shelf VLMs, we also present the performance improvements obtained by training the baseline models on 4DP-QA. Please refer to Section 3.3 for detailed descriptions of each column.

	Camera Motion			**Object Motion**						**3D Spatial Understanding**					
Model	(1) Cam.	(2) Cam.-Obj. Chg.	**Average**	(3) Rotation	(4) Direction	(5) Agent	(6) Moved Dist.	(7) Obj.-Obj. Chg.	**Average**	(8) Depth	(9) Obj.-Obj. Dist.	(10) MV Depth	(11) MV Obj. Dist.	**Average**	**Overall Avg.**
Random	43.7	39.2	41.5	50.0	25.0	50.0	32.0	31.7	28.5	50.0	50.0	50.0	50.0	50.0	40.8
Proprietary VLMs															
GTP-4o [28]	53.3	51.0	52.1	51.3	40.5	18.2	44.3	32.3	41.7	67.1	67.8	64.2	58.2	65.2	53.8
Gemini-2.5-pro [18]	58.4	67.5	63.2	53.1	46.1	45.5	55.7	56.9	50.8	82.6	83.1	81.8	80.6	82.2	66.8
Gemini-2.5-flash [18]	49.6	54.0	51.6	55.8	37.1	18.2	57.1	50.8	46.8	78.8	82.5	80.4	74.6	79.3	60.5
Gemini-robotics [51]	52.8	51.6	52.4	54.0	41.4	36.4	59.3	51.5	48.8	83.2	86.9	82.4	75.4	82.6	62.5
Open-source VLMs															
Qwen2-VL-7B [6]	48.2	38.8	43.8	23.0	35.3	09.1	35.0	19.2	31.1	56.5	59.6	52.7	52.2	55.8	44.2
Qwen3-VL-8B [64]	54.5	42.1	48.7	50.4	32.3	27.3	43.6	33.8	39.6	54.7	53.0	46.6	51.5	52.3	47.3
Qwen2.5-VL-32B [6]	52.8	52.8	52.9	40.7	44.0	27.3	40.0	38.5	42.9	63.2	69.4	55.4	52.2	61.4	52.9
Qwen3-VL-32B [64]	60.6	53.7	58.2	62.8	43.5	36.4	33.6	38.5	46.9	53.2	50.3	52.0	50.0	51.8	52.1
LLAVA-OneVision-7B [34]	47.7	49.6	48.7	34.5	38.8	18.2	33.6	28.5	36.0	54.1	59.6	44.6	50.0	52.9	46.3
LLAVA-OneVision-1.5-8B [3]	46.2	33.4	40.2	38.9	40.1	36.4	35.0	32.3	39.1	60.0	65.6	51.4	59.7	59.6	46.5
NVILA-Video-8B [42]	45.3	24.2	36.8	26.5	40.5	09.1	42.9	30.0	35.6	52.6	52.5	56.1	50.7	52.9	42.2
Baseline vs. Trained on 4DP-QA (Ours)															
Qwen2.5-VL-3B [6]	50.4	41.5	47.1	33.6	39.2	27.3	40.0	29.2	36.9	58.5	56.3	49.3	47.0	54.4	46.7
Qwen2.5-VL-3B + 4DP-QA	81.0	81.5	81.3	72.6	71.6	90.9	78.6	73.1	73.9	88.8	85.2	91.9	78.4	86.8	81.3
Qwen2.5-VL-7B [6]	45.0	30.7	39.5	54.9	43.5	27.3	40.0	36.9	45.1	60.3	59.6	48.0	50.0	56.1	46.6
Qwen2.5-VL-7B + 4DP-QA	83.7	85.7	84.4	75.2	81.0	72.7	79.3	79.2	79.6	90.0	84.7	91.9	83.6	88.1	84.3
NVILA-Lite-8B [42]	50.1	34.0	42.4	16.8	30.2	18.2	28.6	20.8	26.0	55.6	54.1	58.1	53.7	55.4	42.3
NVILA-Lite-8B + 4DP-QA	82.0	85.1	83.5	76.1	86.6	63.6	76.4	76.9	81.6	90.3	86.3	93.9	81.3	88.6	84.4

Training follows a two-stage procedure, with visual and geometry encoders frozen throughout. In the first stage, all model components are frozen except the MLP projector for the geometry encoder, which is trained for 1.5K iterations on 200K randomly sampled QA pairs from the training set. This stage aligns pre-trained L4P features with the LLM input space. In the second stage, we unfreeze the LLM and both MLP projectors (for the visual and geometry encoders), and continue training for 3.1K iterations on the full training set in the same manner as standard VLM training.

5. Experiments

We train baseline VLM models on our dataset (4DP-QA) as described in Section 4.2 and evaluate them on our benchmark (4DP-QA-Bench). To test generalization from training on our dataset, we further evaluate on VLM4D [71], a 4D reasoning benchmark. Finally, we conduct ablations to analyze the importance of the point tracking tasks and the integration of a geometry encoder into the VLM.

5.1. Results on 4DP-QA-Bench

Table 2 summarizes model performance on 4DP-QA-Bench across different question categories. We evaluate both proprietary models (GPT-4o [28], Gemini-2.5-Pro/Flash [18], Gemini-Robotics [51]) and open-source models (Qwen-VL [6, 64], LLaVA-OneVision [3, 34], NVILA [42]). We also compare VLM baselines fine-tuned on 4DP-QA.

Table 3. **Evaluation on VLM4D [71] Benchmark.** We evaluate how training on our 4DP-QA dataset improves generalization to the VLM4D benchmark.

Model	**Real**	**Synthetic**	**Overall**
Proprietary VLMs			
Gemini-2.5-Pro [18]	62.7	62.9	62.8
Gemini-2.5-Flash [18]	51.4	50.3	51.1
Gemini-robotics [51]	61.9	51.5	59.3
Open-source VLMs			
Qwen2-VL-7B [6]	45.0	45.4	45.1
Qwen3-VL-8B [64]	50.6	53.9	51.4
Qwen2.5-VL-32B [6]	51.5	65.6	55.0
Qwen3-VL-32B [64]	57.0	56.0	56.8
LLAVA-OneVision-7B [34]	46.0	33.3	42.9
LLAVA-OneVision-1.5-8B [3]	48.1	38.9	45.9
NVILA-Video-8B [42]	32.4	55.1	38.0
Baseline vs. Trained on 4DP-QA (Ours)			
Qwen2.5-VL-3B [6]	48.2	35.3	45.0
Qwen2.5-VL-3B + 4DP-QA	55.0	56.9	55.5
Qwen2.5-VL-7B [6]	52.9	50.6	52.3
Qwen2.5-VL-7B + 4DP-QA	60.6	73.0	63.6
NVILA-Lite-8B [42]	43.2	41.4	42.8
NVILA-Lite-8B + 4DP-QA	56.4	73.3	60.5

Performance of off-the-shelf VLMs on our benchmark. Open-source VLMs that are strong on standard semantic benchmarks on average perform only slightly better than random guessing on 4DP-QA-Bench: most 7B–8B mod-

Table 4. **Ablation study on dataset composition.** We ablate the effect of adding tracking tasks to Std-4DP-QA by evaluating the resulting models on both our 4DP-QA-Bench and the external VLM4D dataset [71]. See Section 3.3 for description of each question type.

		Camera Motion			Object Motion						3D Spatial Understanding						VLM4D		
	Method	(1) Cam.	(2) Cam.-Obj. Chg.	**Average**	(3)Rotation	(4) Direction	(5) Agent	(6) Moved Dist.	(7) Obj.-Obj. Chg.	**Average**	(8) Depth	(9) Obj.-Obj. Dist.	(10) MV Depth	(11) MV Obj. Dist.	**Average**	**Overall Avg.**	Real	Synthetic	**Average**
(I)	Qwen2.5-VL-3B [6]	50.4	41.5	47.1	33.6	39.2	27.3	40.0	29.2	36.9	58.5	56.3	49.3	47.0	54.4	46.7	48.2	35.3	45.0
(II)	**(I)** + Std-4DP-QA	79.6	86.3	82.0	76.1	78.0	81.8	77.1	80.8	78.2	89.7	84.2	92.6	83.6	88.0	83.1	54.4	47.6	52.7
(III)	**(II)** + True Motion Track (TM)	81.0	81.5	81.3	72.6	71.6	90.9	78.6	73.1	73.9	88.8	85.2	91.9	78.4	86.8	81.3	55.0	56.9	55.5
(IV)	**(II)** + Point Track (PT)	81.0	82.7	82.3	72.6	70.3	72.7	79.3	80.0	73.9	88.5	85.8	93.2	79.9	87.3	82.0	51.7	56.9	53.0
(V)	**(II)** + PT + TM	83.0	84.5	82.8	71.7	73.3	72.7	79.3	80.8	75.4	87.9	85.8	92.6	79.9	87.0	82.4	52.3	51.9	52.2
(I)	Qwen2.5-VL-7B [6]	45.0	30.7	39.5	54.9	43.5	27.3	40.0	36.9	45.1	60.3	59.6	48.0	50.0	56.1	46.6	52.9	50.6	52.3
(II)	**(I)** + Std-4DP-QA	83.0	86.9	84.6	74.3	85.8	72.7	80.0	85.4	82.3	88.2	84.2	93.2	85.8	87.8	85.1	60.5	66.3	61.9
(III)	**(II)** + True Motion Track (TM)	83.7	85.7	84.4	75.2	81.0	72.7	79.3	79.2	79.6	90.0	84.7	91.9	83.6	88.1	84.3	60.6	73.0	63.6
(IV)	**(II)** + Point Track (PT)	83.7	83.0	82.9	75.2	81.0	81.8	80.7	76.9	79.4	89.4	86.3	93.9	84.3	88.7	83.9	59.6	69.2	62.0
(V)	**(II)** + PT + TM	85.4	85.7	85.5	72.6	81.0	63.6	74.3	83.1	78.7	91.8	83.1	92.6	82.1	88.3	84.5	57.8	73.7	61.7
(I)	NVILA-Lite-8B [42]	50.1	34.0	42.4	16.8	30.2	18.2	28.6	20.8	26.0	55.6	54.1	58.1	53.7	55.4	42.3	43.2	41.4	42.8
(II)	**(I)** + Std-4DP-QA	84.2	85.1	84.0	77.9	89.2	72.7	77.9	83.1	84.1	92.4	86.9	93.9	83.6	89.9	85.9	54.9	56.4	55.3
(III)	**(II)** + True Motion Track (TM)	82.0	85.1	83.5	76.1	86.6	63.6	76.4	76.9	81.6	90.3	86.3	93.9	81.3	88.6	84.4	56.4	73.3	60.5
(IV)	**(II)** + Point Track (PT)	80.3	85.4	82.8	79.6	84.5	63.6	75.0	80.0	81.4	91.5	84.7	94.6	85.8	89.6	84.5	54.4	63.6	56.7
(V)	**(II)** + PT + TM	82.7	88.1	85.1	73.5	88.8	54.5	77.1	81.5	82.4	91.2	85.2	93.2	82.8	88.8	85.4	55.4	66.3	58.1
(VI)	**(III)** + Geometry Enc. [5]	88.3	91.3	87.9	78.8	88.8	81.8	87.9	88.5	86.5	90.9	86.9	91.9	78.4	88.1	87.7	51.8	85.4	60.0

els lie in the low-to-mid 40% range, compared to a 40.8% random baseline. Interestingly, in several categories, the baseline models are consistently biased toward wrong answers across samples, resulting in a performance worse than the random baseline. For example, Qwen2.5-VL-7B attains only 30.7% on camera-object questions and 27.3% on agent-motion questions, and NVILA-Lite-8B achieves 16.8% on object-rotation and 20.8% on object-object distance, all far below the random baselines for the corresponding categories. This behavior is also evident qualitatively: Figure 1 shows cases where the baseline model systematically mispredicted simple attributes such as whether the camera is moving or whether the distance to an object is increasing or decreasing. These results indicate that current open-source VLMs have limited capabilities for understanding 3D structure or object dynamics in the environment. On the other hand, some proprietary models perform noticeably better: Gemini-2.5-Pro reaches 66.8% overall, substantially outperforming all open-source baselines.

Training on 4DP-QA yields consistent gains. Fine-tuning on 4DP-QA dramatically improves performance for multiple VLM baselines. Qwen2.5-VL-3B improves from 46.7% to 81.3% overall (+34.6 points), Qwen2.5-VL-7B from 46.6% to 84.3% (+37.7), and NVILA-Lite-8B from 42.3% to 84.4% (+42.1). The gains are consistent across all categories: camera-motion averages rise from the low 40%s to above 80%, object-motion averages from 20–40%s to 70–80%s, and 3D spatial understanding averages from $\approx$ 55% to above 86%. Among all evaluated models, NVILA-Lite-8B+4DP-QA achieves the highest overall score (84.4%). All three fine-tuned models perform better than the strongest proprietary baseline (Gemini-2.5-Pro) by about 14–18 points in overall accuracy. These results show that 4DP-QA is effective at teaching standard VLM architectures to perform fine-grained 4D perception.

5.2. Results on VLM4D Benchmark

To assess how well training on our dataset generalizes, we further evaluate on VLM4D [71], a 4D reasoning benchmark. The benchmark consists of two splits: a real split, containing egocentric [25], YouTube [62], and DAVIS [45] videos; and a synthetic split with samples generated by a video generative model [1]. VLM4D covers QA pairs that emphasize translational and rotational motion, perspective awareness, and motion continuity.

We train NVILA-Lite-8B, Qwen2.5-VL-7B, and Qwen2.5-VL-3B on our dataset and evaluate their performance on VLM4D, along with the other models from our benchmark. All models are evaluated using exact string matching, following [65], to handle minor answer variations, and final scores are reported as accuracy.

Table 3 shows that before training on our dataset, Gemini-2.5-Pro leads with 62.8% accuracy overall, while Qwen3-VL-32B is the best open-source model at 56.8%. Fine-tuning NVILA-Lite-8B, Qwen2.5-VL-7B, and Qwen2.5-VL-3B on our dataset yields gains on both real and synthetic splits: NVILA-Lite-8B jumps from 42.8% to 60.5%, Qwen2.5-VL-7B from 52.3% to 63.6%, and Qwen2.5-VL-3B from 45.0% to 55.5%. After fine-tuning, Qwen2.5-VL-7B outperforms all models overall. Notably, the smaller Qwen2.5-VL-3B model now surpasses the off-the-shelf Qwen2.5-VL-7B and achieves performance on par with the larger 32B models Qwen3-VL-32B and Qwen2.5-VL-32B. These results highlight the effectiveness of our dataset for improving 4D understanding. We show additional benchmark results in the supplementary.

5.3. Ablation Study

We conduct ablations to understand the impact of tracking tasks and geometry features on VLM performance.

True-motion tracking helps 4D reasoning. To better understand the impact of visual point tracking and our newly introduced true-motion point tracking tasks on VLM performance, we conduct a series of ablation studies exploring four distinct training settings. Specifically, we begin with a baseline model trained solely on standard QA questions, which cover camera motion, object motion, and 3D spatial reasoning (Std-4DP-QA in Table 4). We then systematically augment this dataset by: (1) adding visual point tracking (PT) questions only, (2) adding true-motion point tracking (TM) questions only, and (3) adding both PT and TM questions to the training mix. For each of these four settings, models are trained using the same procedures described in Section 4.2, and the overall number of training samples remains constant. Specifically, when introducing tracking questions, we replace 20% of the standard QA samples with tracking-related questions; if both PT and TM questions are included, each type constitutes 10% of the dataset. We set this to 20%, as higher values hurt standard QA performance. We train NVILA-Lite-8B, Qwen2.5-VL-7B, and Qwen2.5-VL-3B on each of these four dataset combinations and evaluate their performance on both our benchmark (4DP-QA-Bench) and VLM4D. Overall, our experiments show that adding tracking tasks enables models to maintain almost all of their performance on standard QA questions in our benchmark, with only a slight decrease, while improving results on the VLM4D benchmark. On average across all baselines, the best results on VLM4D are achieved when we use only the true-motion point tracking task (Row **III** for each baseline). Given that VLM4D is specifically designed to benchmark 4D understanding under camera movement, the observed boost in performance from training on the true-motion tracking task validates the crucial role of this newly introduced task for 4D understanding in VLMs.

Using geometry features improves performance. For the NVILA-Lite-8B model, we additionally train a geometry-augmented variant by adding L4P [5] geometry features to the model, using the same dataset mixture as the model with true-motion point tracking ((**III**)). We observe a noticeable improvement on 4DP-QA-Bench and the VLM4D synthetic split, achieving 85.4% on the VLM4D synthetic split. This highlights the importance of geometry features for 4D understanding in VLMs. However, results on the VLM4D real split show a degradation in performance. We hypothesize that this performance drop is due to the length of the real videos in VLM4D; using a uniform 32-frame sampling does not provide a high enough frame rate for the L4P encoder, which was specifically trained on densely sampled continuous videos.

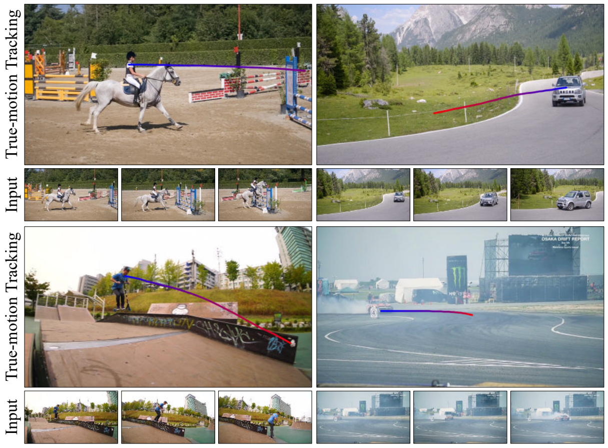


Figure 4. **Visualization of true motion track prediction.** True-motion tracking for dynamic scenes with camera motion (all but bottom right). Estimated tracks disentangle camera motion and summarize object motion as seen in the first frame.

5.4. True-Motion Tracking Visualization

Figure 4 presents qualitative results of true-motion tracking in challenging real-world scenes from [45]. We show outputs from the NVILA-Lite-8B model trained on our dataset, with the prompt: *"Project the 3D trajectory of* $\{$*object name*$\}$ *onto the image plane of frame 0. . . . "*. In each case, the estimated true-motion tracks successfully capture object motion relative to the first frame, providing an intuitive output that effectively summarizes dynamic object behavior despite the presence of camera movement.

6. Conclusion

We present a comprehensive framework to equip VLMs with better 4D understanding. We implement a scalable QA generation pipeline, and collect data from a variety of sources to construct 4DP-QA, a large-scale 4D understanding dataset. We also introduce a new visual perception task, true-motion point tracking, to further encourage fine-grained 4D understanding in the presence of entangled object and camera motion. We demonstrate performance gains from fine-tuning standard VLM architectures on our dataset, and show improved generalization on VLM4D, an external 4D reasoning benchmark. Our ablation studies confirm the effectiveness of including true-motion tracking and further demonstrate that integrating geometric features into the VLM leads to additional improvements.

Acknowledgments

We would like to thank Ligeng Zhu and Guanqi Zhan for discussions about training for the tracking task, and Zhiding Yu and Jan Kautz for their guidance.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for Physical AI. *ArXiv Preprint*, 2025. 7

[2] 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 (NeurIPS)*, 2022. 2

[3] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *ArXiv Preprint*, 2025. 6

[4] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. ScanQA: 3D question answering for spatial scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3

[5] Abhishek Badki, Hang Su, Bowen Wen, and Orazio Gallo. L4P: Low-level 4D vision perception unified. In *International Conference on 3D Vision (3DV)*, 2026. 3, 5, 7, 8

[6] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL: Technical report. *ArXiv Preprint*, 2025. 1, 2, 4, 5, 6, 7

[7] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[8] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[9] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *ArXiv Preprint*, 2020. 3

[10] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 3

[11] Guo Chen, Zhiqi Li, Shihao Wang, Jindong Jiang, Yicheng Liu, Lidong Lu, De-An Huang, Wonmin Byeon, Matthieu Le, Max Ehrlich, Tong Lu, Limin Wang, Bryan Catanzaro, Jan Kautz, Andrew Tao, Zhiding Yu, and Guilin Liu. Eagle 2.5: Boosting long-context post-training for frontier vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 1, 2

[12] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2Cap: Context-aware dense captioning in rgb-d scans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[13] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[14] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. SpatialRGPT: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3

[15] An-Chieh Cheng, Yang Fu, Yukang Chen, Zhijian Liu, Xiaolong Li, Subhashree Radhakrishnan, Song Han, Yao Lu, Jan Kautz, Pavlo Molchanov, et al. 3D aware region prompted vision language model. *ArXiv Preprint*, 2025. 3

[16] Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local all-pair correspondence for point tracking. *ArXiv Preprint*, 2024. 3

[17] Seokju Cho, Jiahui Huang, Seungryong Kim, and Joon-Young Lee. Seurat: From Moving Points to Depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[18] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *ArXiv Preprint*, 2025. 4, 6

[19] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, João Carreira, Andrew Zisserman, and Yi Yang. TAP-Vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3

[20] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 3

[21] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, et al. VLM-3R: Vision-language models augmented with instruction-aligned 3D reconstruction. *arXiv preprint arXiv:2505.20279*, 2025. 3, 5

[22] Shuangkang Fang, I Shen, Yufeng Wang, Yi-Hsuan Tsai, Yi Yang, Shuchang Zhou, Wenrui Ding, Takeo Igarashi, Ming-Hsuan Yang, et al. MeshLLM: Empowering large language models to progressively understand and generate 3D mesh. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 2

[23] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4RTrack: Simultaneous 4D reconstruction and tracking in the world. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 3

[24] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever

comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[25] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 7

[26] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[27] Tanveer Hannan, Md Mohaiminul Islam, Jindong Gu, Thomas Seidl, and Gedas Bertasius. ReVisionLLM: Recursive vision-language model for temporal grounding in hour-long videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[28] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *ArXiv Preprint*, 2024. 6

[29] Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, et al. Token-efficient long video understanding for multimodal LLMs. *ArXiv Preprint*, 2025. 2

[30] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv Preprint*, 2020. 1

[31] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3

[32] Dohwan Ko, Sihyeon Kim, Yumin Suh, Minseo Yoon, Manmohan Chandraker, Hyunwoo J Kim, et al. ST-VLM: Kinematic instruction tuning for spatio-temporal reasoning in vision-language models. *ArXiv Preprint*, 2025. 3

[33] Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. SEED-Bench: Benchmarking multimodal large language models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[34] 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*, 2024. 3, 6

[35] Chaoyu Li, Tianzhi Li, Fei Tao, Zhenyu Zhao, Ziqian Wu, Maozheng Zhao, Juntong Song, Cheng Niu, and Pooyan Fazli. FrameOracle: Learning what to see and how much to see in videos. *ArXiv Preprint*, 2025. 2

[36] 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 (ICML)*, 2023. 2

[37] Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. STI-Bench: Are MLLMs ready for precise spatial-temporal world understanding? In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 3

[38] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *ArXiv Preprint*, 2025. 4

[39] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On pre-training for visual language models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 5

[40] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2

[41] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision (ECCV)*, 2024. 3

[42] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Vishwesh Nath, Jinyi Hu, Sifei Liu, Ranjay Krishna, Daguang Xu, Xiaolong Wang, Pavlo Molchanov, Jan Kautz, Hongxu Yin, Song Han, and Yao Lu. NVILA: Efficient frontier visual language models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 5, 6, 7

[43] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3D: Situated question answering in 3D scenes. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 3

[44] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria Digital Twin: A new benchmark dataset for egocentric 3D machine perception. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 3

[45] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *ArXiv Preprint*, 2017. 7, 8

[46] Zhangyang Qi, Ye Fang, Zeyi Sun, Xiaoyang Wu, Tong Wu, Jiaqi Wang, Dahua Lin, and Hengshuang Zhao. GPT4Point: A unified framework for point-language understanding and generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 1

[48] Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, et al. SAT:

Dynamic spatial aptitude training for multimodal language models. In *Conference on Language Modeling (COLM)*, 2024. 3

[49] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does CLIP know about a red circle? Visual prompt engineering for VLMs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4, 5

[50] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. SHIFT: A synthetic driving dataset for continuous multi-task domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[51] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *ArXiv Preprint*, 2025. 6

[52] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Zhiyong Zhao, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. DriveVLM: The convergence of autonomous driving and large vision-language models. *ArXiv Preprint*, 2024. 3

[53] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. VGGSfM: Visual geometry grounded deep structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[54] Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. SpatialVID: A large-scale video dataset with spatial annotations. *ArXiv Preprint*, 2025. 3

[55] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3, 4

[56] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. InternVideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision (ECCV)*, 2024. 1, 2

[57] Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. Llama-Mesh: Unifying 3D mesh generation with language models. *ArXiv Preprint*, 2024. 2

[58] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *ArXiv Preprint*, 2024. 2

[59] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Iurii Makarov, Bingyi Kang, Xin Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. SpatialTrackerV2: 3D point tracking made easy. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 3

[60] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. SpatialTrackerV2: 3D point tracking made easy. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 3

[61] Yuan Xie, Tianshui Chen, Zheng Ge, and Lionel Ni. Video-MTR: Reinforced multi-turn reasoning for long video understanding. *ArXiv Preprint*, 2025. 2

[62] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-VOS: A large-scale video object segmentation benchmark. *ArXiv Preprint*, 2018. 7

[63] 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 (ECCV)*, 2024. 2

[64] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *ArXiv Preprint*, 2025. 5, 6

[65] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3, 5, 7

[66] Bowei Zhang, Lei Ke, Adam W Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. *arXiv preprint arXiv:2504.14717*, 2025. 3

[67] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *pami*, 2024. 1

[68] 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*, 2025. 3

[69] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3D LLM: Learning position-aware video representation for 3D scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[70] Hanyu Zhou and Gim Hee Lee. Uni4D-LLM: A unified spatiotemporal-aware VLM for 4D understanding and generation. *ArXiv Preprint*, 2025. 2

[71] Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. VLM4D: Towards spatiotemporal awareness in vision language models. In *IEEE International Conference on Computer Vision (ICCV)*, 2025. 3, 6, 7

[72] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. LLaVA-3D: A simple yet effective pathway to empowering LLLMs with 3D-awareness. *ArXiv Preprint*, 2024. 3