When Left Is Right? Camera-Motion Understanding beyond Screen

Anonymous authors
Paper under double-blind review

Abstract

Can a vision–language model tell left from right? Modern VLMs solve visual puzzles and follow hour-long videos, yet when asked whether the camera moved left or right, most do no better than a coin flip. Adding camera-motion data or geometric representations has been tried, separately, and neither closes the gap. We argue that camera-motion understanding needs two conditions at once: the representation must couple off-screen camera motion with on-screen flow dynamics, and it must be interpreted by the language model rather than trained as a standalone pose or depth estimator. We propose CameraVLM, which pairs a motion-state representation with state–language alignment. Camera tokens capture how the camera moves off screen, flow tokens capture how the subject and background move on screen, and a two-stage training strategy keeps language in the loop so that both are learned together with camera-motion semantics. With only 9B parameters, CameraVLM outperforms all open-source and camera-motion-specific models by a wide margin across three benchmarks and more than 20 baselines, and matches or surpasses GPT-6 Astra as well as human viewers in a 48-participant study.

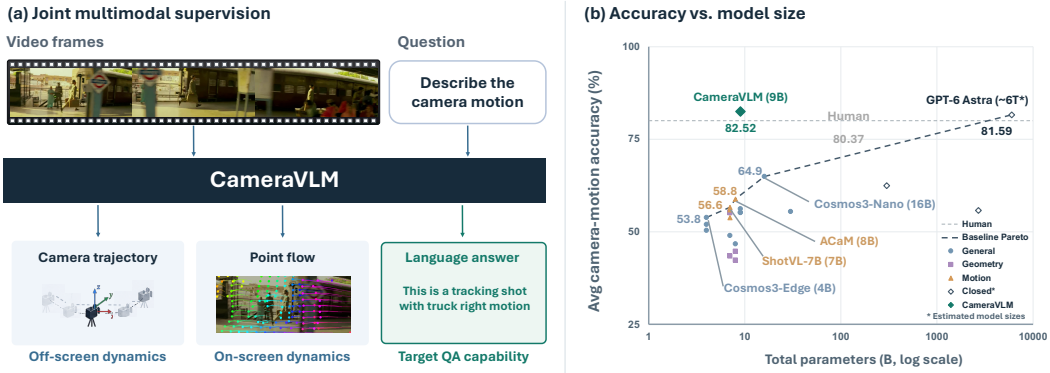

Figure 1: **CameraVLM learns camera motion from off-screen and on-screen dynamics.** (a) Given video frames and a question, CameraVLM is trained to recover the camera trajectory (off-screen dynamics) and the point flow of the scene (on-screen dynamics), and answers the question in language from these states. (b) Average understanding accuracy on CameraBench, CineTechBench, and ShotBench versus total parameters. With 9B parameters, CameraVLM lies above the Pareto front of existing models and surpasses GPT-6 Astra and the human average; sizes of closed-source models are estimated.

1 Introduction

Can a vision–language model tell left from right? The answer seems obvious. Modern VLMs such as Qwen3-VL (Bai et al., 2025a) solve visual puzzles (Song et al., 2025), carry out multi-step visual reasoning (Lu et al., 2023; Yue et al., 2023), and answer questions about hour-long videos (Fu et al., 2024; Li et al., 2025; Wang et al., 2025d), and they increasingly serve as the perception and reasoning core of embodied agents (Li et al., 2026; Wang et al., 2026c; NVIDIA et al., 2026). Yet ask whether the camera in a video moved left or right, and most open-source and commercial VLMs are correct only about half the time, no better than flipping a coin (Lin et al., 2025; Kong et al.,

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.

2025), while humans reach 86% (Fig. 2). Is this failure caused by a lack of camera-motion data, a lack of geometric representations, or something else?

Both have been tried, separately. One line of work adds data: CameraBench (Lin et al., 2025) and ACaM (Tan et al., 2026) improve camera-motion recognition through motion-specific data and instruction tuning, mapping videos to fixed motion labels. The other focuses on representations: DepthLM (Cai et al., 2025) and DepthVLM (Yu et al., 2026) model depth and scene structure, while Cambrian-P (Yang et al., 2026), VLM-3R (Fan et al., 2025), and CamInject (Du et al., 2026) incorporate camera pose or 3D geometric features. Neither is enough on its own (Table 1): camera-motion-specific models improve on the benchmark they are tuned for but often fail to transfer to others, and models with explicit geometry perform no better than general VLMs of the same size, some even worse. In this paper, we argue that this is because camera-motion understanding requires two conditions to hold together, and neither direction provides both: *the representation must couple off-screen 3D geometry with on-screen image motion rather than encode geometry alone, and it must be reasoned over by language rather than merely trained as a pose or depth estimator*.

First, camera motion is observed through its interaction with scene content. Camera translation and rotation can produce similar **on-screen** changes, while subject motion may dominate the apparent motion within the frame. At the same time, limited on-screen motion does not imply a stationary camera relative to the surrounding or off-screen scene. The same camera move looks different depending on what is in front of it: a dolly toward a distant scene barely changes the image, and a pan and a truck are hard to tell apart unless the scene has both near and far objects.[1] The representation must therefore capture how the on-screen subject and background move in the image, together with the **off-screen** camera geometry. Second, spatial descriptions depend on a reference, and the question specifies it. When A and B face each other, a cup on A's left is on B's right; when the camera follows a walking person, the camera moves relative to the street but not to the person. Camera-motion understanding therefore requires a representation that jointly encodes off-screen camera motion and on-screen image motion, and that the language model can interpret under the reference specified by the query.

We propose CameraVLM (Fig. 1), which addresses these two conditions through motion-state representation and state–language alignment. On the representation side, we add two sets of tokens to the VLM's token sequence, processed by the backbone together with the video and question tokens. Pose-grounded camera tokens, whose hidden states are decoded into camera pose, encode the off-screen camera motion, namely its translation, rotation, and viewpoint change; flow tokens, pooled from the flow pathway, encode the on-screen motion, namely the cross-frame motion of points on the subject and background and whether they remain visible. Together, they let the model separate camera motion from subject motion and tell apart camera moves that look alike on screen.

On the alignment side, we construct joint training data with language, camera-pose, and flow supervision, and adopt a two-stage training strategy that keeps language in the loop, so the new states are learned together with their camera-motion semantics and remain interpretable by the language model. The first stage fits the new tokens to the language model: with it frozen, they are trained with pose, flow, and QA losses, and the QA gradient through the frozen model aligns them with its existing representation space. The second stage teaches the model to use them: the language model is unfrozen and trained jointly on language, pose, and flow objectives, with general video QA mixed in to preserve language abilities.

We evaluate on three camera-motion benchmarks (CameraBench, CineTechBench, and ShotBench) against more than 20 models, spanning open-source, geometry-aware, camera-motion-specific, and commercial VLMs, together with a 48-participant human study. With only 9B parameters, CameraVLM outperforms all open-source and camera-motion-specific models by a wide margin and, on average, matches or surpasses GPT-6 Astra and human viewers.

Our contributions are as follows:

- We argue that camera-motion understanding in VLMs requires two conditions to hold together: a representation that couples off-screen camera motion with on-screen flow dynamics, and interpretation of that representation by the language model.

[1] In cinematography, a *pan* rotates the camera in place, like turning one's head; a *truck* moves it sideways; a *dolly* moves it toward or away from the scene.



Figure 2: **Spatial motion semantics depend on reference and spatio-temporal state.** The same scene can yield opposite directions under different references or similar appearances from different camera motions. CameraVLM models these relations through camera pose and flow. On left/right questions, most existing VLMs still perform poorly, while CameraVLM approaches human performance.

- We propose CameraVLM, which pairs a motion-state representation of camera and flow tokens with state–language alignment through a two-stage training strategy that keeps language in the loop.
- Across three camera-motion benchmarks and a 48-participant human study, CameraVLM, with only 9B parameters, outperforms all open-source and camera-motion-specific models by a wide margin and matches or surpasses GPT-6 Astra and human viewers on average.

2 Related Work

Camera-motion understanding. CameraBench (Lin et al., 2025) evaluates video VLMs on camera-motion classification. CineTechBench (Wang et al., 2025c) and ShotBench (Liu et al., 2025) broaden the scope to cinematographic attributes, while ACaM (Tan et al., 2026) further decomposes camera motion into fine-grained atomic movements. Another line introduces explicit camera information through camera pose, camera tokens, or external motion estimates (Yang et al., 2026; Du et al., 2026; Feng et al., 2026). These methods improve camera-motion recognition but mainly operate on predefined motion semantics, without an explicit spatio-temporal state that can be reinterpreted as the target, reference, or temporal range changes.

Spatial representations for VLMs. DepthLM (Cai et al., 2025) and DepthVLM (Yu et al., 2026) improve spatial representations through depth supervision. Cambrian-P (Yang et al., 2026) and VLM-3R (Fan et al., 2025) introduce camera pose, camera tokens, or richer spatial states, while SpaceMind (Zhao et al., 2025) uses camera representations to guide modality fusion and Q-GeoMem (Gao et al., 2026) further learns question-guided geometric memory. These methods mainly model depth, pose, or scene structure, with limited explicit modeling of cross-frame motion correspondence and continuous spatio-temporal states required for camera-motion understanding.

Embodied and world models. RynnBrain (Dang et al., 2026; Li et al., 2026) and Hy-Embodied-VLM (Wang et al., 2026c) learn states for egocentric perception, action, and interaction, while Orca (Wang et al., 2026b) and Cosmos (NVIDIA et al., 2025; 2026) model broader physical states and state transitions. These models capture richer dynamics and physical information but do not explicitly resolve reference ambiguity in camera motion. The motion of the camera, subject, and background still changes with the reference entity and frame, so a richer dynamic state alone does not determine camera-motion semantics.

Camera and motion state estimation. One line of feed-forward methods predicts camera and geometric states (Wang et al., 2023; Leroy et al., 2024; Wang et al., 2025a), while tracking meth-

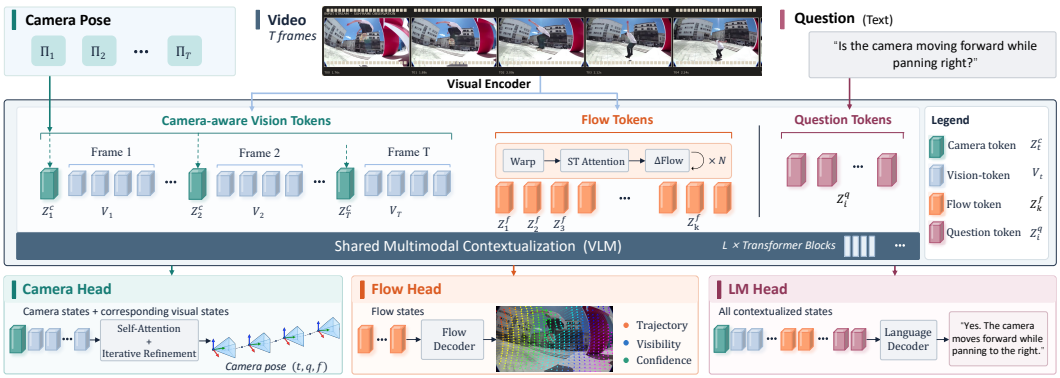

Figure 3: Architecture of CameraVLM. Camera tokens provide continuous camera states, while flow tokens provide cross-frame motion states. Camera, vision, flow, and question tokens are jointly contextualized by the shared VLM backbone. The Camera Head predicts continuous camera parameters, the flow pathway predicts trajectories, visibility, and confidence, and the LM Head interprets the joint states under the query to answer camera-motion questions.

ods estimate cross-frame trajectories (Doersch et al., 2023; Karaev et al., 2023; 2024; Harley et al., 2025; Lai et al., 2026). Other work uses trajectories for cross-frame alignment or video representations (Lai & Vedaldi, 2025; Zheng et al., 2026). These methods provide camera and motion states but typically separate state estimation from language interpretation. Our camera and flow pathways retain both states within the VLM and align them with camera-motion semantics through state–language alignment.

3 METHOD

3.1 PROBLEM FORMULATION

Answering a camera-motion question requires two things: the motion itself, and the frame in which the question asks to describe it. Neither is given directly. The frame comes from the question rather than the video, and the motion must be inferred from image changes that mix off-screen camera and on-screen scene motion. We therefore formulate camera-motion understanding as reading a dynamic state under a query, and derive from the projection equation what this state must contain.

Camera state and motion semantics. To make the camera component explicit, we represent its state at time t as

$$C(t) = \big(\mathbf{c}(t), R_{WC}(t), K(t)\big), \tag{1}$$

where $\mathbf{c}(t)$ is the camera center, $R_{WC}(t)$ its orientation, and $K(t)$ its intrinsics. Their changes over time describe camera translation, rotation, and intrinsic changes such as zoom.

The resulting dynamic state does not by itself determine its language description; the interpretation also depends on the relation specified by the question. We write a query as

$$Q = (a, b, F, \tau), \tag{2}$$

where a and b denote the target and reference entities, F the reference frame, and τ the temporal interval. The same dynamic state can therefore support different motion descriptions under different queries. In a follow shot, for example, the camera moves relative to the scene while remaining nearly stationary relative to the subject. We formulate camera-motion understanding as

$$r_Q = \Phi(S_\tau, Q), \tag{3}$$

where S_τ denotes the continuous dynamic state over the queried interval.

Image-motion ambiguity. Image motion does not uniquely identify its physical source. For a scene or subject point $\mathbf{P}(t)$, its camera-frame position is

$$\mathbf{X}^C(t) = R_{WC}^\top(t)\,[\mathbf{P}(t) - \mathbf{c}(t)]\,, \tag{4}$$

with image position $\mathbf{u}(t) = \pi(K(t), \mathbf{X}^C(t))$. Differentiating, as in the image Jacobian of visual servoing (Chaumette & Hutchinson, 2006), gives

$$\dot{\mathbf{u}} = J_X R_{WC}^{\top}(\dot{\mathbf{P}} - \dot{\mathbf{c}}) - J_X(\boldsymbol{\omega}_C \times \mathbf{X}^C) + J_K \dot{K}, \tag{5}$$

where the three terms capture scene/camera translation, camera rotation, and intrinsic change. Similar image motion can therefore arise from different camera and scene dynamics. Pan and lateral translation can produce similar first-order motion at nearly constant depth, while parallax separates them across depths. Likewise, zoom and axial translation can produce similar instantaneous scale changes but differ in their depth dependence. Detailed derivations are provided in Appendix A.

State representation. These ambiguities motivate our two complementary state components. Camera states represent camera motion, while flow states represent cross-frame motion and correspondence across subjects and background. We therefore represent the video state as

$$S = (S_{\text{camera}}, S_{\text{flow}}), \tag{6}$$

where S_{camera} is supervised by camera pose and S_{flow} by cross-frame motion. Together, they retain the information needed to separate different sources of image motion, while Eq. 3 connects the resulting state to the semantics specified by the query.

3.2 Architecture

Fig. 3 gives an overview of CameraVLM. Following Eq. 6, we represent the video with two complementary dynamic states: pose-grounded camera states and trajectory-grounded flow states. Both are integrated into a shared VLM together with the native video and question tokens.

VLM backbone. We build CameraVLM on Qwen3.5-VL (Qwen Team, 2026). Its native vision encoder extracts the video features, which are then contextualized by the multimodal backbone together with camera, flow, and question tokens. The same shared representation is used for both state prediction and language reasoning, so pose and flow supervision directly shape the information available to the language model. No separate visual backbone is introduced for camera or flow state learning.

The multimodal input is

$$X_{\text{VLM}} = [Z_{\text{video+camera}}; Z_{\text{flow}}; Z_Q], \tag{7}$$

where camera tokens are interleaved with the video tokens of their source frames, Z_{flow} contains pooled cross-frame motion states, and Z_Q denotes the question tokens. The shared backbone jointly contextualizes all token types. The Camera Head predicts camera parameters from the contextualized camera states, while the LM Head reads the joint representation to answer the query.

3.3 Pose-Grounded Camera States

The camera state should describe how the observer itself moves: its translation, rotation, and intrinsic change at each sampled frame. These quantities are not determined by any single frame; they must be inferred from how the view changes across frames. We therefore form camera states inside the backbone, where each position can attend to the whole video.

We insert one learnable camera token per source frame next to that frame's video tokens. After the backbone, the hidden states at these positions form the camera states $H^c = \{h_t^c\}_{t=1}^T$. The Camera Head applies self-attention across the camera states and iteratively refines per-frame translation, rotation, and field of view, $\hat{Y}^c = \text{CameraHead}(H^c)$, which are supervised with camera pose.

Because this supervision acts on H^c rather than on a separate encoder, the states that the language model reads when answering are the same states that must predict camera geometry. Camera motion thus remains continuous in the representation instead of being reduced to discrete labels such as pan, truck, or zoom, and how these states map to language is determined by the query (Sec. 3.5).

3.4 Trajectory-Grounded Flow States

The flow stream complements the camera states with how visible content moves across frames. As indicated by Eq. 5, relative motion and parallax across subjects, foreground, and background provide important cues for separating camera motion from scene motion.

Our flow pathway follows CoWTracker (Lai et al., 2026) and operates on features from the VLM's vision encoder. Point motion is iteratively refined from multi-level visual features through feature warping and spatio-temporal attention. The resulting point states are decoded into trajectories, visibility, and confidence and supervised by $\mathcal{L}_{\text{flow}}$.

We initialize points on a uniform image grid so that motion across the full scene is retained without predefining semantic regions. Their motion states $\{s_i\}$ are compressed into a fixed set of flow tokens through attention pooling,

$$Z_{\text{flow}} = \text{AttnPool}(\{s_i\}), \tag{8}$$

which are then passed to the shared VLM backbone. Individual trajectories are not assigned to camera, subject, or background in advance; their roles are interpreted jointly with the camera states, visual context, and question.

3.5 Dynamic-State Learning and Language Alignment

We jointly train the camera and flow states with language supervision so that they preserve continuous motion information and remain accessible to language reasoning. The training objective is

$$\mathcal{L} = \mathcal{L}_{\text{QA}} + \lambda_c \mathcal{L}_{\text{camera}} + \lambda_f \mathcal{L}_{\text{flow}}, \tag{9}$$

where $\mathcal{L}_{\text{camera}}$ supervises per-frame translation, rotation, and field of view, and $\mathcal{L}_{\text{flow}}$ supervises point trajectories, visibility, and confidence. The geometric terms keep the states distinct at a finer granularity than motion labels, while $\mathcal{L}_{\text{QA}}$ teaches the language model to read them under the entity, reference, and interval specified by each question.

Training proceeds in two stages. In state-interface adaptation, the language backbone is frozen, the camera and flow pathways are trained, and the vision encoder is updated with a small learning rate. The QA loss still backpropagates through the frozen language model to the new states, so they learn to enter the representation space the pretrained model already uses, rather than only fitting pose and flow targets in isolation. In joint state–language adaptation, the language backbone is also unfrozen and all components are optimized under the full objective.

Training samples also differ in what the camera tokens receive. In video-only samples, the tokens are learned queries and the camera state must be inferred from the frames. In pose-only samples, ground-truth pose is embedded into the camera tokens without video, so the language model learns how camera geometry maps to camera-motion language. Video+pose samples provide both, tying the inferred state to the explicit geometry. Mixing the three ties camera states from different sources to the same camera-motion language, so that this correspondence enters the shared representation rather than residing only in the camera tokens. At inference, only the video and question are given.

3.6 Training Data

State–language alignment requires the same clip to carry both kinds of supervision: pose and flow labels specify what the states should encode, and QA or captions specify how that content is described. Using disjoint supervision would weaken the correspondence between learned motion states and their language descriptions. We therefore pair camera-motion language with pose and flow labels on the same clips.

Camera-motion language comes from ACaM (Tan et al., 2026) (27K QA) and CameraBench (Lin et al., 2025) (68K QA and captions). Since these videos lack camera annotations, we estimate camera parameters offline with VGGT-Ω (Wang et al., 2026a) and generate trajectory supervision with CoWTracker (Lai et al., 2026). Pairing these signals on the same clips directly links continuous motion states to their language descriptions.

We further use the video-understanding training data from Cambrian-P (Yang et al., 2026), including VSI-590K spatial examples and 510K general video QA examples from Cambrian-S (Yang et al., 2025b), together with the ground-truth and ViPE-based (Huang et al., 2025) pseudo camera-pose annotations used by Cambrian-P. We additionally generate offline CoWTracker trajectory supervision for a large portion of these videos. Kubric (Greff et al., 2022) provides additional ground-truth camera pose and trajectories, and we include 10K spatial CoT examples from 3DThinker (Chen et al., 2025). Training and evaluation videos are strictly disjoint. Detailed sampling ratios are given in Table 5 (Appendix B).

Table 1: **Camera-motion understanding accuracy (%).** Best result per column in **bold**, second best underlined. CameraVLM (9B) surpasses all open-source and camera-motion-specific baselines by a large margin, exceeds human average accuracy, and is comparable to GPT-6.

Model	Params	CameraBench	CineTechBench	ShotBench	Avg
General open-source models					
Qwen2.5-VL (Bai et al., 2025b)	7B	60.00	48.67	38.36	49.01
Qwen3-VL (Bai et al., 2025a)	8B	62.30	37.17	40.73	46.73
Qwen3.5 (backbone) (Qwen Team, 2026)	9B	63.64	54.87	46.98	55.16
Qwen3.5 + CoT (Qwen Team, 2026)	9B	69.20	53.98	45.47	56.22
Orca (Wang et al., 2026b)	4B	62.20	52.21	36.85	50.42
DepthVLM (Yu et al., 2026)	4B	59.30	56.64	40.09	52.01
Cosmos3-Edge (NVIDIA et al., 2026)	4B	57.60	56.64	47.41	53.88
Cosmos3-Nano (NVIDIA et al., 2026)	16B	70.30	67.26	57.33	64.96
Hy-Embodied-VLM (Wang et al., 2026c)	30B-A3B	65.70	53.10	47.63	55.48
RynnBrain1.1 (Li et al., 2026)	9B	65.80	56.64	43.32	55.25
Geometry-aware models					
Cambrian-P (Yang et al., 2026)	7B	64.30	53.09	48.06	55.15
CamDistill (Du et al., 2026)	8B	40.60	42.48	43.97	42.35
CamInject (Du et al., 2026)	8B	45.50	46.90	41.81	44.74
Spatial-MLLM (Wu et al., 2025a)	–	57.70	37.17	32.11	42.33
VLM-3R-7B (Fan et al., 2025)	7B	57.00	33.63	39.66	43.43
Camera-motion-specific models					
CameraModel-7B (Lin et al., 2025)	7B	70.90	52.21	44.83	55.98
ACaM (Tan et al., 2026)	8B	63.00	57.52	56.03	58.85
CamReasoner-7B (Wu et al., 2026)	7B	68.00	48.67	44.83	53.83
ShotVL-7B (Liu et al., 2025)	7B	66.10	42.48	61.21	56.60
Closed-source models					
Claude Opus 4.8	–	63.49	54.87	49.03	55.80
Gemini 3.5 Flash	–	66.70	61.95	58.84	62.50
GPT-6 Astra (high)	–	82.50	**83.19**	79.09	81.59
Human	–	76.20	77.76	87.30	80.37
CameraVLM (Ours)	9B	**88.20**	77.88	**81.47**	**82.52**

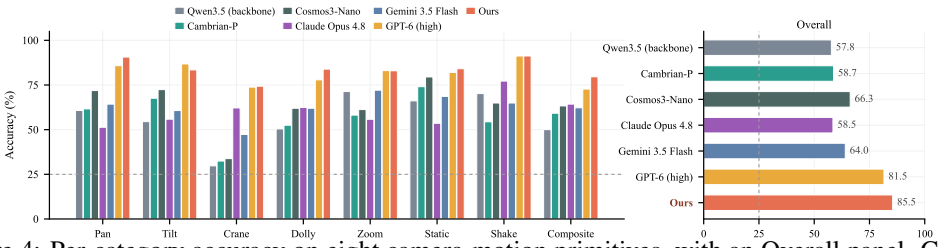

Figure 4: Per-category accuracy on eight camera-motion primitives, with an Overall panel. CameraVLM leads in most categories and is second only to GPT-6 Astra in the others; full numbers for all models are in Table 8.

4 Experiments

Implementation details. We build the model on native Qwen3.5-VL-9B. The visual encoder has 27 Transformer blocks, the language model has 32 layers, and the hidden dimension is 4096. Videos are uniformly sampled to at most 128 frames. Training uses AdamW with a cosine learning-rate schedule: 2×10^{-6} for the language model and the visual tower, 1×10^{-4} for the camera module, and 1.5×10^{-4} for the flow module. The first and second stages run for 20K and 30K iterations, respectively, on 32 H100 GPUs, taking about 7.5 days in total. Further details are given in Appendix B.

Benchmarks. We evaluate camera-motion understanding on CameraBench (Lin et al., 2025), CineTechBench (Wang et al., 2025c), and ShotBench (Liu et al., 2025). For CameraBench, we uniformly sample 1,000 Yes/No questions across camera-motion categories; for CineTechBench and ShotBench, we use all camera-motion samples. All main evaluations use only the raw video and question, without external geometric information. We also evaluate camera pose on TUM RGB-D (Sturm et al., 2012), ScanNet (Dai et al., 2017), and Sintel (Butler et al., 2012), and flow on TAP-Vid DAVIS (Doersch et al., 2022; Pont-Tuset et al., 2017).

Human evaluation. We further evaluate human performance with 48 participants, including professional filmmakers, computer vision students, and camera-motion researchers. Participants answer randomly assigned Yes/No and four-way multiple-choice questions.

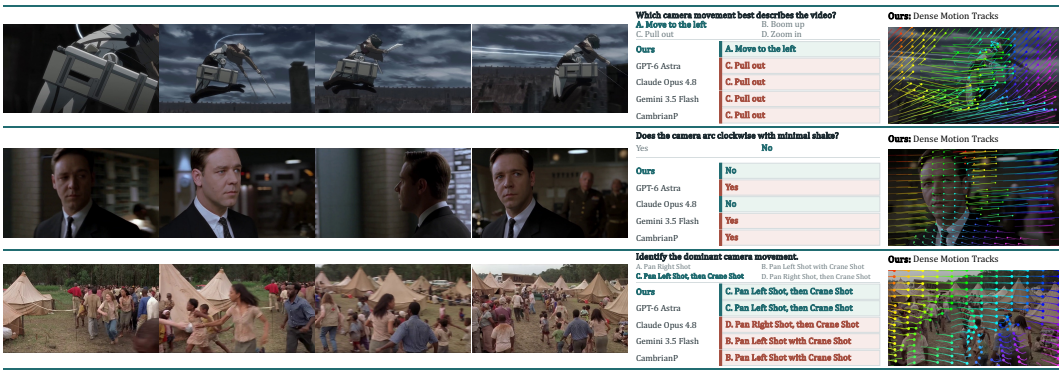

Figure 5: **Qualitative comparison.** Each row compares a model prediction with the bolded ground truth; green denotes correct predictions and red denotes errors. The rightmost column shows the dense flow predicted by CameraVLM. In the first clip, large subject motion causes most models to mistake a truck-left shot for a pull-out, while CameraVLM correctly identifies the lateral camera motion from background parallax.

Table 2: **Camera pose estimation.** ATE, RPE translation, and RPE rotation are reported on ScanNet, TUM RGB-D, and Sintel; lower is better. The learned camera states retain stable pose information across datasets and remain competitive with dedicated streaming pose models.

	ScanNet			TUM RGB-D			Sintel		
Model	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r	ATE	RPE_t	RPE_r
Offline									
DUSt3R-GA (Wang et al., 2023)	0.081	0.028	0.784	0.083	0.017	3.567	0.417	0.250	5.796
MASt3R-GA (Leroy et al., 2024)	0.078	0.020	0.475	0.038	0.012	0.448	0.185	0.060	1.496
MonST3R-GA (Zhang et al., 2024)	0.077	0.018	0.529	0.098	0.019	0.935	0.111	0.044	0.869
Fast3R (Yang et al., 2025a)	0.155	0.123	3.491	0.090	0.101	1.425	0.371	0.298	13.750
FLARE (Zhang et al., 2025)	0.064	0.023	0.971	0.026	0.013	0.475	0.207	0.090	3.015
VGGT (Wang et al., 2025a)	0.035	0.015	0.380	0.009	0.008	0.350	0.172	0.061	0.470
π^3 (Wang et al., 2025e)	0.031	0.013	0.347	0.014	0.009	0.312	0.074	0.040	0.282
MapAnything (Keetha et al., 2025)	0.052	0.025	0.720	0.029	0.023	0.370	0.226	0.077	0.640
Streaming									
StreamVGGT (Zhuo et al., 2025)	0.127	0.041	1.880	0.062	0.030	0.690	0.273	0.109	0.850
CUT3R (Wang et al., 2025b)	0.096	0.022	0.590	0.045	0.015	0.440	0.215	0.070	0.630
Point3R (Wu et al., 2025b)	0.097	0.035	2.791	0.058	0.031	0.758	0.442	0.154	1.897
Spann3R (Wang & Agapito, 2024)	0.096	0.023	0.661	0.056	0.021	0.591	0.329	0.110	4.471
G2VLM (Hu et al., 2025)	0.148	0.048	1.220	0.129	0.044	0.700	0.301	0.135	1.450
Cambrian-P (Yang et al., 2026)	0.078	0.023	0.880	0.046	0.020	0.580	0.239	0.081	2.440
CameraVLM	0.124	0.044	2.177	0.049	0.022	1.440	0.148	0.064	1.420

4.1 QA EVALUATION

Table 1 reports the main comparison. Although CameraModel (Lin et al., 2025) and ACaM (Tan et al., 2026) are trained on camera-motion-specific data, both remain clearly below CameraVLM. CameraVLM consistently outperforms its backbone and all open-source baselines across the three benchmarks, exceeds average human performance, and remains competitive with GPT-6. The gain is consistent across benchmarks with different motion distributions, a property that single-benchmark fine-tuning does not exhibit.

Figure 4 reports performance across eight camera-motion categories. CameraVLM performs consistently well across all categories and achieves the best overall accuracy, showing that the gains are broadly distributed across different motion types. Figure 5 presents qualitative comparisons on representative clips. In the first example, substantial subject motion causes most models to mistake a truck-left shot for a pull-out, while CameraVLM correctly recovers the camera motion using the background motion cues. The predicted dense flow further visualizes these cross-frame dynamics, helping disambiguate camera and subject motion. Additional examples are provided in Appendix D.2, with full videos in the supplementary material.

4.2 STATE EVALUATION

We further evaluate the learned camera and flow states to verify that they retain the continuous motion information introduced by their corresponding supervision. This provides a direct check that the model preserves explicit motion states beyond the final language predictions.

Table 3: **Point tracking on TAP-Vid DAVIS.** AJ, δ_{avg}, and OA are reported; higher is better. The learned flow states retain cross-frame tracking information after integration into the VLM.

Model	Train data	AJ	δ_{avg}	OA
TAPIR (Doersch et al., 2023)	Kub	56.2	70.0	86.5
CoTracker (Karaev et al., 2023)	Kub	61.8	76.1	88.3
TAPTR (Li et al., 2024)	Kub	63.0	76.1	91.1
LocoTrack (Cho et al., 2024)	Kub	62.9	75.3	87.2
BootsTAPIR (Doersch et al., 2024)	Kub+15M	61.4	73.6	88.7
CoTracker3 (Karaev et al., 2024) (online)	Kub+15k	63.8	76.3	90.2
CoTracker3 (Karaev et al., 2024) (offline)	Kub+15k	64.4	76.9	91.2
DOT (Moing et al., 2023)	Kub	60.1	74.5	89.0
DELTA (Ngo et al., 2024)	Kub	60.8	75.2	87.6
AllTracker (Harley et al., 2025)	Kub	61.9	75.4	87.8
AllTracker (Harley et al., 2025)	Kub+Mix	63.3	76.3	90.1
CoWTracker (Lai et al., 2026)	Kub	65.5	78.0	92.1
Ours	Kub+CoW distill	50.0	63.2	88.2

Table 4: Ablation study of the main components and training strategy. Two-stage variants are trained for 10K iterations per stage; one-stage training runs only the 10K-iteration state-interface stage. Numbers are therefore lower than in Table 1.

Method	CameraBench	CineTech	ShotBench	Avg.
Backbone	63.64	54.87	46.98	55.2
w/o Flow	64.50	55.75	72.20	64.1
w/o Camera & Flow	67.20	57.10	42.10	55.5
One-stage Training	64.50	64.00	53.88	60.8
Full Model	**68.20**	**67.88**	**70.79**	**69.0**

Camera state. We evaluate pose estimation from the camera states (Table 2) to verify that they retain meaningful camera-motion information. Baseline results follow Yang et al. (2026). Although several baselines are trained on substantially broader geometric data, our model still achieves competitive pose estimation. On Sintel, it obtains the best ATE and RPE_t among all streaming baselines, showing that the camera states learn and preserve transferable camera-motion information.

Flow state. We evaluate point tracking from the flow states on TAP-Vid DAVIS (Table 3). Baseline results follow Lai et al. (2026). Although the flow pathway is integrated into the VLM for camera-motion understanding rather than optimized as a standalone tracker, it retains meaningful cross-frame tracking information, including competitive occlusion accuracy. This confirms that the learned flow states preserve correspondence and motion information after joint training.

4.3 Ablation Study

Table 4 evaluates the contributions of the state representation and supervision. For a controlled comparison, two-stage variants are trained for 10K iterations per stage, and one-stage training runs only the 10K-iteration state-interface stage without joint state–language adaptation; the full model here therefore differs from the main result in Table 1. Removing flow states degrades performance, showing that cross-frame motion complements the camera states. Without camera and flow supervision, performance drops close to the backbone, confirming that explicit state supervision is important for learning useful dynamic representations. The full training strategy further improves performance over one-stage training, indicating that state learning and state–language alignment are complementary.

The Appendix further shows that inference-time masking causes only limited degradation (Table 6), suggesting that dynamic information is distributed across the shared representation, and that the model is robust to most flow configurations (Table 7).

4.4 Limitations

Some training videos rely on offline supervision generated by VGGT-Ω and CoWTracker, so state learning may inherit errors from upstream estimation. Rare compound motions and unusual imaging conditions are also less represented in the training data. Finally, our current setting focuses on shot-level camera motion and does not yet model camera-motion relations across cuts or longer cinematographic sequences.

5 Conclusion

We represent camera motion through continuous camera and flow states aligned with camera-motion semantics. Keeping both off-screen camera motion and on-screen image motion in the representation yields consistent gains across camera-motion benchmarks, particularly on spatial direction and compound camera motion. This provides a basis for interpreting shot composition and subject–camera relations, and for more controllable video generation and editing.

LLM USAGE STATEMENT

Large language models were used to aid and polish writing, and for retrieval and discovery, such as finding related work.

REPRODUCIBILITY STATEMENT

We will release the code, model weights, training data, and evaluation scripts. Implementation details are given in Appendix B.

REFERENCES

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

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

Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A Naturalistic Open Source Movie for Optical Flow Evaluation. In *Computer Vision – ECCV 2012*, pp. 611–625. Springer Berlin Heidelberg, 2012. ISBN 9783642337833. doi: 10.1007/978-3-642-33783-3_44. URL http://dx.doi.org/10.1007/978-3-642-33783-3_44.

Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. DepthLM: Metric Depth From Vision Language Models, 2025. URL https://arxiv.org/abs/2509.25413.

François Chaumette and Seth Hutchinson. Visual servo control. I. Basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006. doi: 10.1109/MRA.2006.250573.

Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Xiang An, Yan Feng, Peng Pei, Xunliang Cai, and Ruqi Huang. Think with 3D: Geometric Imagination Grounded Spatial Reasoning from Limited Views, 2025. URL https://arxiv.org/abs/2510.18632.

Seokju Cho, Jiahui Huang, Jisu Nam, Honggyu An, Seungryong Kim, and Joon-Young Lee. Local All-Pair Correspondence for Point Tracking, 2024. URL https://arxiv.org/abs/2407.15420.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes, 2017. URL https://arxiv.org/abs/1702.04405.

Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangpin Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, Minghao Zhu, Xiao Lin, Yang Bai, Qian Jiang, Yaxi Zhao, Minghua Zeng, Junlong Gao, Yuming Jiang, Jun Cen, Siteng Huang, Liuyi Wang, Wenqiao Zhang, Chengju Liu, Jianfei Yang, Shijian Lu, and Deli Zhao. RynnBrain: Open Embodied Foundation Models, 2026. URL https://arxiv.org/abs/2602.14979.

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, 2022. URL https://arxiv.org/abs/2211.03726.

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, 2023. URL https://arxiv.org/abs/2306.08637.

Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, and Andrew Zisserman. BootsTAP: Bootstrapped Training for Tracking-Any-Point, 2024. URL https://arxiv.org/abs/2402.00847.

Dazhao Du, Shiyan Du, Jian Liu, Yongjian Yu, Bohai Gu, Tao Han, Hualuo Liu, Eric Liu, Yujia Zhang, Xi Chen, and Song Guo. Temporally Grounded Compositional Camera Motion Understanding via Geometric Knowledge Distillation, 2026. URL https://arxiv.org/abs/2608.10932.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. VLM-3R: Vision-Language Models Augmented with Instruction-Aligned 3D Reconstruction, 2025. URL https://arxiv.org/abs/2505.20279.

Haoan Feng, Sri Harsha Musunuri, and Guan-Ming Su. Geometry-Guided Camera Motion Understanding in VideoLLMs, 2026. URL https://arxiv.org/abs/2603.13119.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis, 2024. URL https://arxiv.org/abs/2405.21075.

Xianqiang Gao, Qizhi Chen, Delin Qu, Haoming Song, Zhigang Wang, Bin Zhao, Dong Wang, and Xuelong Li. Q-GeoMem: Question-Guided Geometric Memory for Video Spatial Reasoning, 2026. URL https://arxiv.org/abs/2605.27318.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti Derek Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator, 2022. URL https://arxiv.org/abs/2203.03570.

Adam W. Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Pavel Tokmakov, Suya You, Rares Ambrus, Katerina Fragkiadaki, and Leonidas J. Guibas. AllTracker: Efficient Dense Point Tracking at High Resolution, 2025. URL https://arxiv.org/abs/2506.07310.

Wenbo Hu, Jingli Lin, Yilin Long, Yunlong Ran, Lihan Jiang, Yifan Wang, Chenming Zhu, Runsen Xu, Tai Wang, and Jiangmiao Pang. G^2VLM: Geometry Grounded Vision Language Model with Unified 3D Reconstruction and Spatial Reasoning, 2025. URL https://arxiv.org/abs/2511.21688.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. ViPE: Video Pose Engine for 3D Geometric Perception, 2025. URL https://arxiv.org/abs/2508.10934.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is Better to Track Together, 2023. URL https://arxiv.org/abs/2307.07635.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and Better Point Tracking by Pseudo-Labelling Real Videos, 2024. URL https://arxiv.org/abs/2410.11831.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal Feed-Forward Metric 3D Reconstruction, 2025. URL https://arxiv.org/abs/2509.13414.

Fei Kong, Jinhao Duan, Kaidi Xu, Zhenhua Guo, Xiaofeng Zhu, and Xiaoshuang Shi. LRR-Bench: Left, Right or Rotate? Vision-Language models Still Struggle With Spatial Understanding Tasks, 2025. URL https://arxiv.org/abs/2507.20174.

Zihang Lai and Andrea Vedaldi. Tracktention: Leveraging Point Tracking to Attend Videos Faster and Better, 2025. URL https://arxiv.org/abs/2503.19904.

Zihang Lai, Eldar Insafutdinov, Edgar Sucar, and Andrea Vedaldi. CoWTracker: Tracking by Warping instead of Correlation, 2026. URL https://arxiv.org/abs/2602.04877.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding Image Matching in 3D with MASt3R, 2024. URL https://arxiv.org/abs/2406.09756.

Hongyang Li, Hao Zhang, Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, and Lei Zhang. TAPTR: Tracking Any Point with Transformers as Detection, 2024. URL https://arxiv.org/abs/2403.13042.

Kehan Li, Bohan Hou, Minghao Zhu, Tianyi Zhang, Zesen Cheng, Zhikai Wang, Sicong Leng, Xin Li, Xiao Lin, Biying Yao, Minghua Zeng, Jiangpin Liu, Ronghao Dang, Jiayan Guo, Siteng Huang, Haoyu Zhao, Heng Ping, Yaxi Zhao, Tong Zhao, Kexiang Wang, Tong Lu, Shengke Xue, Jiahao Tang, Yulei Wang, Zejing Wang, Jianwei Gao, Shijian Lu, Chengju Liu, Jianfei Yang, Mingxiu Chen, and Deli Zhao. RynnBrain 1.1: Towards More Capable and Generalizable Embodied Foundation Model, 2026. URL https://arxiv.org/abs/2607.17977.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. VideoChat-Flash: Hierarchical Compression for Long-Context Video Modeling, 2025. URL https://arxiv.org/abs/2501.00574.

Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Tiffany Ling, Yuhan Huang, Sifan Liu, Mingyu Chen, Rushikesh Zawar, Xue Bai, Yilun Du, Chuang Gan, and Deva Ramanan. Towards Understanding Camera Motions in Any Video, 2025. URL https://arxiv.org/abs/2504.15376.

Hongbo Liu, Jingwen He, Yi Jin, Dian Zheng, Yuhao Dong, Fan Zhang, Ziqi Huang, Yinan He, Yangguang Li, Weichao Chen, Yu Qiao, Wanli Ouyang, Shengjie Zhao, and Ziwei Liu. ShotBench: Expert-Level Cinematic Understanding in Vision-Language Models, 2025. URL https://arxiv.org/abs/2506.21356.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts, 2023. URL https://arxiv.org/abs/2310.02255.

Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Dense Optical Tracking: Connecting the Dots, 2023. URL https://arxiv.org/abs/2312.00786.

Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. DELTA: Dense Efficient Long-range 3D Tracking for any video, 2024. URL https://arxiv.org/abs/2410.24211.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma,

Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos World Foundation Model Platform for Physical AI, 2025. URL https://arxiv.org/abs/2501.03575.

NVIDIA, Aditi, Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, Aarti Basant, Mukesh Beladiya, Mohammad Qazim Bhat, Zaid Pervaiz Bhat, Dan Blick, Vanni Brighella, Han Cai, Tiffany Cai, Eric Cameracci, Jiaxin Cao, Yulong Cao, Mark Carlson, Carlos Casanova, Ting-Yun Chang, Yan Chang, Yu-Wei Chao, Prithvijit Chattopadhyay, Roshan Chaudhari, Chieh-Yun Chen, Junyu Chen, Ke Chen, Qizhi Chen, Wenkai Chen, Xiaotong Chen, Yu Chen, An-Chieh Cheng, Click Cheng, Xiu Chia, Jeana Choi, Chaeyeon Chung, Wenyan Cong, Yin Cui, Magdalena Dadela, Nalin Dadhich, Wenliang Dai, Joyjit Daw, Alperen Degirmenci, Rodrigo Vieira Del Monte, Robert Denomme, Sameer Dharur, Marco Di Lucca, Ke Ding, Wenhao Ding, Yifan Ding, Yuzhu Dong, Nicole Drumheller, Yilun Du, Aigul Dzhumamuratova, Aleksandr Efitorov, Hamid Eghbalzadeh, Naomi Eigbe, Imad El Hanafi, Hassan Eslami, Benedikt Falk, Jiaojiao Fan, Jim Fan, Amol Fasale, Sergiy Fefilatyev, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Vikram Fugro, Prashant Gaikwad, TJ Galda, Katelyn Gao, Yihuai Gao, Wenhang Ge, Sreyan Ghosh, Arushi Goel, Vivek Goel, Akash Gokul, Rama Govindaraju, Jinwei Gu, Miguel Guerrero, Elfie Guo, Aryaman Gupta, Siddharth Gururani, Hugo Hadfield, Song Han, Ankur Handa, Zekun Hao, Mohammad Harrim, Ali Hassani, Nathan Hayes-Roth, Yufan He, Chris Helvig, Cyrus Hogg, Madison Huang, Michael Huang, Sophia Huang, Yufan Huang, Jacob Huffman, DeLesley Hutchins, Suneel Indupuru, Boris Ivanovic, Arihant Jain, Joel Jang, Ryan Ji, Yanan Jian, Dongfu Jiang, Jingyi Jin, Atharva Joshi, Nikhilesh Joshi, Pranjali Joshi, Andy Ju, Jaehun Jung, Weiwei Kang, Scott Kassekert, Jan Kautz, Ashna Khetan, Julia Kiczka, Slawek Kierat, Gwanghyun Kim, Kuno Kim, Sunny Kim, Kezhi Kong, Xin Kong, Zhifeng Kong, Tomasz Kornuta, Egor Krivov, Hui Kuang, Saurav Kumar, Chia-Wen Kuo, George Kurian, Wojciech Kutak, JF Lafleche, Himangshu Lahkar, Omar Laymoun, Jayjun Lee, Sanggil Lee, Gabriele Leone, Boyi Li, Freya Li, Jiajun Li, Jinfeng Li, Ling Li, Pengcheng Li, Shangru Li, Tingle Li, Xiaolong Li, Xuan Li, Zhaoshuo Li, Zhiqi Li, Hao Liang, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Ming-Yu Liu, Sifei Liu, Zihan Liu, Hai Loc Lu, Xiangyu Lu, Alice Luo, Ruipu Luo, Wenjie Luo, Jiangran Lyu, Martin Ding Ma, Nic Ma, Qianli Ma, Dawid Majchrowski, Louis Marcoux, Miguel Martin, Qing Miao, Ashkan Mirzaei, Shreyas Misra, Kaichun Mo, Durra Mohsin, Hyejin Moon, Pawel Morkisz, Saeid Motiian, Kirill Motkov, Seungjun Nah, Yashraj Narang, Deepak Narayanan, Thabang Ngazimbi, Julian Ouyang, Shubham Pachori, David Page, Yatian Pang, Sehwi Park, Mahesh Patekar, Mostofa Patwary, Marco Pavone, Trung Pham, Wei Ping, Soha Pouya, Shrimai Prabhumoye, Varun Praveen, Delin Qu, Hesam Rabeti, Morteza Ramezanali, Marilyn Reeb, Xuanchi Ren, Kristen Rumley, Wojciech Rymer, Jun Saito, Yeongho Seol, John Shao, Piyush Shekdar, Tianwei Shen, Humphrey Shi, Min Shi, Stella Shi, Kevin Shih, Mohammad Shoeybi, Mateusz Sieniawski, Shuran Song, Alexander Sotelo, Amir Sotoodeh, Sunil Srinivasa, Vignesh Srinivasakumar, Bartosz Stefaniak, Rahul Heinrich Steiger, Shangkun Sun, Jiaxiang Tang, Shitao Tang, Yangyang Tang, Yue Tang, Tolou Tavakkoli, Kayley Ting, Krzysztof Tomala, Wei-Cheng Tseng, Jibin Varghese, Sergei Vasilev, Thomas Volk, Raju Wagwani, Roger Waleffe, Andrew Z. Wang, Boxiang Wang, Haoxiang Wang, Qiao Wang, Shihao Wang, Shijie Wang, Ting-Chun Wang, Yan Wang, Yu Wang, Rohit Watve, David Wehr, Fangyin Wei, Xinshuo Weng, Jay Zhangjie Wu, Kedi Wu, Hongchi Xia, Summer Xiao, Tianjun Xiao, Kevin Xie, Daguang Xu, Jiashu Xu, Mengyao Xu, Ruqing Xu, Xingqian Xu, Yao Xu, Dinghao Yang, Dong Yang, Hans Yang, Xiaodong Yang, Xuning Yang, Yichu Yang, Yurong You, Zhiding Yu, Hao Yuan, Simon Yuen, Xiaohui Zeng, Pengcuo Zeren, Cindy Zha, Haotian Zhang, Jenny Zhang, Jing Zhang, Liangkai Zhang, Paris Zhang, Shun Zhang, Xuanmeng Zhang, Zhizheng Zhang, Ann Zhao, Yilin Zhao, Yuliya Zhautouskaya, Charles Zhou, Fengzhe Zhou, Shilin Zhu, Yuke Zhu, Dima Zhylko, and Artur Zolkowski. Cosmos 3: Omnimodal World Models for Physical AI, 2026. URL https://arxiv.org/abs/2606.02800.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS Challenge on Video Object Segmentation, 2017. URL https://arxiv.

org/abs/1704.00675.

Qwen Team. Qwen3.5: Towards Native Multimodal Agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Yueqi Song, Tianyue Ou, Yibo Kong, Zecheng Li, Graham Neubig, and Xiang Yue. VisualPuzzles: Decoupling Multimodal Reasoning Evaluation from Domain Knowledge, 2025. URL https://arxiv.org/abs/2504.10342.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580. IEEE, October 2012. doi: 10.1109/iros.2012.6385773. URL http://dx.doi.org/10.1109/IROS.2012.6385773.

Yuwen Tan, Joey Huang, Jin Huang, Haoxiang Li, and Boqing Gong. Natural Language Camera Movement Understanding, 2026. URL https://arxiv.org/abs/2607.03043.

Hengyi Wang and Lourdes Agapito. 3D Reconstruction with Spatial Memory, 2024. URL https://arxiv.org/abs/2408.16061.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer, 2025a. URL https://arxiv.org/abs/2503.11651.

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

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D Perception Model with Persistent State, 2025b. URL https://arxiv.org/abs/2501.12387.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy, 2023. URL https://arxiv.org/abs/2312.14132.

Xinran Wang, Songyu Xu, Xiangxuan Shan, Yuxuan Zhang, Muxi Diao, Xueyan Duan, Yanhua Huang, Kongming Liang, and Zhanyu Ma. CineTechBench: A Benchmark for Cinematographic Technique Understanding and Generation, 2025c. URL https://arxiv.org/abs/2505.15145.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, Min Dou, Kai Chen, Wenhai Wang, Yu Qiao, Yali Wang, and Limin Wang. InternVideo2.5: Empowering Video MLLMs with Long and Rich Context Modeling, 2025d. URL https://arxiv.org/abs/2501.12386.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-Equivariant Visual Geometry Learning, 2025e. URL https://arxiv.org/abs/2507.13347.

Yihao Wang, Yuheng Ji, Mingyu Cao, Yanqing Shen, Runze Xiao, Huaihai Lyu, Senwei Xie, Euan Liu, Klara Tian, Tianfeng Long, Yichi Zhang, Zhengliang Cai, Ruike Chen, Jifan Zhao, Ruochuan Shi, Zihan Tang, Jing Lyu, Wenxing Tan, Ningbo Zhang, Yangtao Hu, Yuming Gao, Xiansheng Chen, Junkai Zhao, Congsheng Xu, Boan Zhu, Ziqi Wang, Yupu Feng, Qiongqiong Zhang, Yingli Zhao, Yulong Ao, Shaoxuan Xie, You Liu, Guocai Yao, Leiduo Zhang, Xiaodan Liu, Yunyan Zhang, Yance Jiao, Xinyan Yang, Jiaxing Wei, Xu Liu, Tengfei Pan, Shaokai Nie, Chunlei Men, Sen Cui, Xiaojie Jin, Hongyang Li, Jianlan Luo, Yao Mu, Yunchao Wei, Jun Yan, Hang Zhao, Xiaolong Zheng, Jiaming Li, Yonghua Lin, Tiejun Huang, Zhongyuan Wang, and Pengwei Wang. Orca: The World is in Your Mind, 2026b. URL https://arxiv.org/abs/2606.30534.

Ziyi Wang, Xumin Yu, Yongming Rao, Yonggen Ling, Yunheng Li, Oran Wang, Mingqi Gao, Yuchen Zhou, Yves Liang, Zuyan Liu, Yani Zhang, Rui Huang, Xiaoran Xu, Bowen Yuan, Yifu Yuan, Xu Tan, He Zhang, Yufei Huang, Shenghao Zhang, Hongsheng Wu, Han Hu, and Zhengyou Zhang. Hy-Embodied-VLM-1.0: Efficient Physical-World Agents, 2026c. URL https://arxiv.org/abs/2607.12894.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-MLLM: Boosting MLLM Capabilities in Visual-based Spatial Intelligence, 2025a. URL https://arxiv.org/abs/2505.23747.

Hang Wu, Yujun Cai, Zehao Li, Haonan Ge, Bowen Sun, Junsong Yuan, and Yiwei Wang. CamReasoner: Reinforcing Camera Movement Understanding via Structured Spatial Reasoning, 2026. URL https://arxiv.org/abs/2602.00181.

Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3R: Streaming 3D Reconstruction with Explicit Spatial Pointer Memory, 2025b. URL https://arxiv.org/abs/2507.02863.

Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D Reconstruction of 1000+ Images in One Forward Pass, 2025a. URL https://arxiv.org/abs/2501.13928.

Jihan Yang, Zifan Zhao, Xichen Pan, Shusheng Yang, Junyi Zhang, Bingyi Kang, Hu Xu, Shang-Wen Li, and Saining Xie. Cambrian-P: Pose-Grounded Video Understanding, 2026. URL https://arxiv.org/abs/2605.22819.

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

Hanxun Yu, Xuan Qu, Yuxin Wang, Jianke Zhu, and Lei Ke. Unlocking Dense Metric Depth Estimation in VLMs, 2026. URL https://arxiv.org/abs/2605.15876.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for Expert AGI, 2023. URL https://arxiv.org/abs/2311.16502.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A Simple Approach for Estimating Geometry in the Presence of Motion, 2024. URL https://arxiv.org/abs/2410.03825.

Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. FLARE: Feed-forward Geometry, Appearance and Camera Estimation from Uncalibrated Sparse Views, 2025. URL https://arxiv.org/abs/2502.12138.

Ruosen Zhao, Zhikang Zhang, Jialei Xu, Jiahao Chang, Dong Chen, Lingyun Li, Weijian Sun, and Zizhuang Wei. SpaceMind: Camera-Guided Modality Fusion for Spatial Reasoning in Vision-Language Models, 2025. URL https://arxiv.org/abs/2511.23075.

Chenhao Zheng, Jieyu Zhang, Jianing Zhang, Weikai Huang, Ashutosh Kumar, Quan Kong, Oncel Tuzel, Chun-Liang Li, and Ranjay Krishna. TrajTok: Learning Trajectory Tokens enables better Video Understanding, 2026. URL https://arxiv.org/abs/2602.22779.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4D Visual Geometry Transformer, 2025. URL https://arxiv.org/abs/2507.11539.

A Derivations for Camera-Motion Formulation

A.1 Relative Motion under a Moving Reference Frame

Reference-conditioned relative motion. For a target entity a, reference entity b, and reference frame F, their relative displacement is

$$\mathbf{d}_{a|b}^{F}(t) = R_{WF}^{\top}(t)\left[\mathbf{x}_a^W(t) - \mathbf{x}_b^W(t)\right]. \tag{10}$$

Let

$$\mathbf{y} = \mathbf{x}_a^W - \mathbf{x}_b^W. \tag{11}$$

If $\boldsymbol{\omega}_F$ is the angular velocity of F relative to W, expressed in F, then

$$\frac{d}{dt}R_{WF}^{\top} = -[\boldsymbol{\omega}_F]_{\times}R_{WF}^{\top}. \tag{12}$$

Hence,

$$\frac{d}{dt}\mathbf{d}_{a|b}^{F} = \frac{dR_{WF}^{\top}}{dt}\mathbf{y} + R_{WF}^{\top}\dot{\mathbf{y}} \tag{13}$$

$$= R_{WF}^{\top}\left(\dot{\mathbf{x}}_a^W - \dot{\mathbf{x}}_b^W\right) - \boldsymbol{\omega}_F \times \mathbf{d}_{a|b}^{F}, \tag{14}$$

which is the relative velocity expressed in the moving frame F.

For a static relation, the world-frame positions remain fixed while the coordinate representation depends on F. If two reference frames have opposite horizontal axes, the same displacement has opposite horizontal signs.

For a follow shot, let a be the subject and b the camera. When

$$\dot{\mathbf{x}}_a^W \approx \dot{\mathbf{x}}_b^W \tag{15}$$

and reference-frame rotation is small over the interval, $\mathbf{v}_{a|b}^{C} \approx 0$, while both entities may have nonzero world-frame velocity.

A.2 Camera-Frame Motion Decomposition

For a scene point $\mathbf{P}(t)$ with camera-frame position $\mathbf{X}^C = R_{WC}^{\top}(\mathbf{P} - \mathbf{c})$, differentiating gives

$$\dot{\mathbf{X}}^C = R_{WC}^{\top}(\dot{\mathbf{P}} - \dot{\mathbf{c}}) - \boldsymbol{\omega}_C \times \mathbf{X}^C, \tag{16}$$

where the terms correspond to scene/subject motion minus camera translation, and camera rotation, respectively.

A.3 Pan–Truck Ambiguity

Consider a static scene point $\mathbf{X}^C = (X, Y, Z)^{\top}$ with normalized image coordinates

$$x = \frac{X}{Z}, \qquad y = \frac{Y}{Z}. \tag{17}$$

For pure yaw $\boldsymbol{\omega}_C = (0, \omega_y, 0)^{\top}$ with no translation, Eq. 16 gives

$$\dot{\mathbf{X}}^C = -\boldsymbol{\omega}_C \times \mathbf{X}^C. \tag{18}$$

Therefore,

$$\dot{X} = -\omega_y Z, \qquad \dot{Y} = 0, \qquad \dot{Z} = \omega_y X, \tag{19}$$

and

$$\dot{x}_{\text{pan}} = -\omega_y(1 + x^2), \qquad \dot{y}_{\text{pan}} = -\omega_y xy. \tag{20}$$

For pure lateral camera translation T_x,

$$\dot{X} = -T_x, \qquad \dot{Y} = 0, \qquad \dot{Z} = 0, \tag{21}$$

which gives

$$\dot{x}_{\text{truck}} = -\frac{T_x}{Z}, \qquad \dot{y}_{\text{truck}} = 0. \tag{22}$$

Let

$$Z = Z_0 + \delta Z, \qquad |x|, |y| \le \epsilon, \qquad \frac{T_x}{Z_0} = \omega_y. \tag{23}$$

Using

$$\frac{Z_0}{Z_0 + \delta Z} = 1 - \frac{\delta Z}{Z_0} + O\left(\frac{\delta Z^2}{Z_0^2}\right), \tag{24}$$

we obtain

$$\dot{x}_{\text{pan}} - \dot{x}_{\text{truck}} = O(\epsilon^2) + O\left(\frac{\delta Z}{Z_0}\right), \tag{25}$$

$$\dot{y}_{\text{pan}} - \dot{y}_{\text{truck}} = O(\epsilon^2). \tag{26}$$

The first-order ambiguity disappears through depth-dependent parallax, higher-order perspective terms, or direct camera-pose information.

A.4 Zoom–Dolly Ambiguity

Let $\mathbf{p}$ denote image coordinates relative to the principal point:

$$\mathbf{p} = f \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}. \tag{27}$$

For optical zoom with fixed camera pose,

$$\dot{\mathbf{p}}_{\text{zoom}} = \frac{\dot{f}}{f}\mathbf{p}. \tag{28}$$

For forward axial camera translation with velocity T_z,

$$\dot{Z} = -T_z, \tag{29}$$

which gives

$$\dot{\mathbf{p}}_{\text{dolly}} = \frac{T_z}{Z}\mathbf{p}. \tag{30}$$

If all visible points lie at depth Z_0 and

$$\frac{\dot{f}}{f} = \frac{T_z}{Z_0}, \tag{31}$$

the two instantaneous radial flows coincide.

For two visible points with $Z_i \neq Z_j$, dolly produces radial rates

$$\frac{T_z}{Z_i} \qquad \text{and} \qquad \frac{T_z}{Z_j}, \tag{32}$$

while zoom produces the same $\dot{f}/f$ for both. The spatial variation of the flow therefore separates axial translation from focal-length change.

A.5 Reflection

For a planar mirror with unit normal $\mathbf{n}$, the reflection matrix is

$$M = I - 2\mathbf{n}\mathbf{n}^\top, \qquad \det(M) = -1. \tag{33}$$

A motion vector transforms as

$$\mathbf{v}_{\text{refl}} = M\mathbf{v}. \tag{34}$$

Its normal component changes sign,

$$\mathbf{n}^\top \mathbf{v}_{\text{refl}} = -\mathbf{n}^\top \mathbf{v}, \tag{35}$$

while tangential components are unchanged. The resulting image direction depends on the mirror orientation and the queried reference frame. A left–right reversal occurs when this sign change projects onto the horizontal axis of that frame.

A.6 Query-Induced State Equivalence

Let $\mathcal{Q}_{\mathrm{tr}}$ denote the training query set. Define

$$S_1 \sim_{\mathcal{Q}_{\mathrm{tr}}} S_2 \quad \Longleftrightarrow \quad \Phi(S_1, Q) = \Phi(S_2, Q), \qquad \forall Q \in \mathcal{Q}_{\mathrm{tr}}. \tag{36}$$

For

$$\mathcal{L}_{\mathrm{QA}} = -\log p(A \mid V, Q), \tag{37}$$

states in the same equivalence class produce the same target answers for all observed queries. Mapping them to the same internal representation incurs no additional QA loss.

There may still exist $Q^\star \notin \mathcal{Q}_{\mathrm{tr}}$ such that

$$\Phi(S_1, Q^\star) \neq \Phi(S_2, Q^\star). \tag{38}$$

Answer equivalence under the training queries therefore does not imply state equivalence.

Camera-pose supervision constrains camera translation, rotation, and viewpoint change. Scene-wide flow supervision constrains cross-frame motion and correspondence across image regions. Without metric depth supervision, world-frame object translation remains ambiguous in scale.

B Implementation Details

CameraVLM builds on Qwen3.5-VL-9B. Videos are uniformly sampled to at most 128 frames (64 temporal groups). The vision tower (27 blocks) keeps its native spatial merger and multimodal RoPE; the language backbone has 32 layers with $d = 4096$.

Camera pathway. Each source frame carries one camera token. Two learnable query embeddings q_{first} and q_{shared} parameterize the token positions; frame-specific states emerge from multimodal contextualization. Camera hidden states are projected to 2048-D and consumed by a VGGT-style camera head (Wang et al., 2025a; Yang et al., 2026) (four self-attention layers, four refinement iterations) that predicts translation, quaternion rotation and field of view per frame.

Flow pathway. Unmerged visual features from four intermediate layers are fused through a DPT decoder (128 channels) and fed to the CoWTracker tracking head (three refinement iterations) to predict trajectories, visibility and confidence on a fixed 16×16 grid (256 anchor points). Learned-query attention pooling (64 queries) compresses these anchor states into 64 flow tokens in the 4096-D language space. Each pooled token is residually anchored to the mean video embedding.

Sequence layout. Camera tokens are interleaved with their source frames; flow tokens are inserted before the question: $X_{\mathrm{LM}} = [Z_{\mathrm{video+camera}};\, Z_{\mathrm{flow}};\, Z_Q]$. All state tokens are computed in prefill and cached; they occupy multimodal input positions and are excluded from vocabulary generation.

Supervision. Flow: pixel-space Huber loss on trajectories plus binary cross-entropy on visibility and confidence. Camera: multi-stage regression on translation, rotation and field of view. CoWTracker produces offline pseudo labels and does not participate in inference.

Optimization. AdamW with cosine schedule. Learning rates: 2×10^{-6} (LM and vision tower), 1×10^{-5} (merger and general modules), 1×10^{-4} (camera module), 1.5×10^{-4} (flow modules). The two stages run 20K and 30K iterations at batch size 256; total training takes about 7.5 days.

Training data. Table 5 lists the training sources together with their language, camera, and flow supervision and their sampling ratios.

Table 5: **Training data composition.** Sampling ratios are normalized across all training sources. Camera supervision is provided by ground-truth poses or offline pseudo poses, while flow supervision is provided by ground-truth trajectories or CoWTracker teacher labels.

Data source	Language	Camera supervision	Flow supervision	Ratio
ACaM (Tan et al., 2026)	QA	VGGT-Ω	CoWTracker	15.05%
CameraBench VQA (Lin et al., 2025)	QA	VGGT-Ω	CoWTracker	15.05%
CameraBench Caption (Lin et al., 2025)	Caption	VGGT-Ω	CoWTracker	3.23%
Cambrian-P / Cambrian-S (Yang et al., 2026; 2025b)	Spatial + Video QA	GT / ViPE pseudo pose	CoWTracker*	48.39%
Kubric (Greff et al., 2022)	–	GT	GT	10.75%
3DThinker (Chen et al., 2025)	CoT	–	CoWTracker	7.53%

C Additional Experiments

C.1 Inference-time State Token Masking

Table 6: Inference-time masking of state tokens. All variants use the fully trained model and differ only in the state tokens masked during inference.

Inference Setting	CameraBench	CineTech	ShotBench	Avg.
Full	88.20	77.88	81.47	82.5
Mask Flow Tokens	88.51	75.30	81.04	81.6
Mask Camera Tokens	88.30	76.19	81.92	82.1
Mask Both	88.81	77.07	81.71	82.5

Masking camera, flow, or both state-token groups at inference produces only limited performance changes, indicating that the learned dynamic information is distributed across the shared multimodal representation after state–language alignment.

C.2 Hyperparameter Sensitivity

Table 7 shows that the model is largely insensitive to the configuration of the flow pathway. Varying the number of flow tokens, spatial grid, temporal window, or refinement iterations changes CineTech accuracy by at most 1.69 points. In contrast, reducing the input to 64 and 32 frames decreases accuracy by 4.35 and 7.01 points, respectively, indicating that sufficient temporal coverage is important for camera-motion understanding.

Table 7: Hyperparameter sensitivity analysis. Only the specified configuration is changed while all other settings remain fixed.

Setting	Value	CineTech
Default	–	**77.88**
Flow Tokens	16	77.07
	32	77.07
	128	77.07
Grid	8	77.07
	32	77.07
Window	8	76.19
	32	77.07
Iterations	1	76.19
	5	77.07
Frames	64	73.53
	32	70.87

D Additional Results

D.1 Full Per-Category Results

Table 8 reports accuracy on every camera-motion primitive category for representative models from Table 1. Overall is the accuracy pooled over every evaluated item across the three benchmarks,

so it need not equal the unweighted Avg. column of Table 1. CameraVLM achieves the highest overall accuracy (85.48%) and leads in most individual categories, with particularly strong gains on geometrically challenging primitives such as Crane, Dolly and Truck, where many baselines fall below 55%.

Table 8: **Per-category camera-motion accuracy (%).** Overall pools every evaluated item across CameraBench, CineTechBench and ShotBench.

Model	Overall	Arc	Composite	Crane	Dolly	Other	Pan	Pedestal	Roll	Shake	Static	Tilt	Tracking	Truck	Zoom
Open-source models															
ACaM	60.56	50.00	56.12	55.41	51.65	61.65	53.12	90.00	69.09	78.95	70.71	52.41	100.00	56.74	74.42
Cambrian-P	58.72	40.00	59.18	32.43	52.48	63.91	61.61	60.00	65.45	54.39	74.06	67.59	60.00	40.43	58.14
CamDistill	41.72	40.00	36.73	33.78	37.60	35.34	46.43	40.00	45.45	45.61	49.79	50.34	60.00	41.13	27.91
CamInject	44.51	55.00	39.80	43.24	38.43	47.37	42.41	40.00	56.36	47.37	55.65	42.07	60.00	47.52	31.01
Cosmos3-Edge	54.53	20.00	56.12	25.68	40.08	56.39	60.71	80.00	61.82	52.63	63.18	63.45	60.00	48.94	65.12
Cosmos3-Nano	66.27	75.00	63.27	33.78	61.98	75.94	71.88	70.00	61.82	64.91	79.50	72.41	70.00	51.06	61.24
DepthVLM	53.46	40.00	42.86	29.73	38.84	66.17	57.59	90.00	49.09	61.40	68.62	53.10	70.00	43.97	61.24
Hy-Embodied-VLM-1.0	59.48	60.00	54.08	31.08	53.72	69.92	59.38	80.00	70.91	64.91	75.31	53.79	60.00	46.81	62.02
Orca	54.03	40.00	55.10	27.03	45.04	62.41	50.00	70.00	61.82	54.39	71.97	56.55	90.00	36.88	61.24
Qwen3.5 (our backbone)	57.77	30.00	50.00	29.73	50.41	64.66	60.71	80.00	63.64	70.18	66.11	54.48	70.00	50.35	71.32
Qwen3.5 + CoT	61.13	45.00	45.92	43.24	54.55	70.68	64.29	60.00	52.73	84.21	71.55	62.07	60.00	49.65	68.22
RynnBrain1.1	58.53	50.00	46.94	48.65	61.57	69.92	61.61	40.00	67.27	59.65	55.65	64.14	60.00	41.84	65.89
Closed-source models															
Claude Opus 4.8	58.47	50.00	64.29	62.16	62.40	62.41	51.34	90.00	61.82	77.19	53.56	55.86	70.00	56.03	55.81
Gemini 3.5 Flash	64.05	50.00	62.24	47.30	61.98	73.68	64.29	60.00	60.00	64.91	68.62	60.69	70.00	59.57	72.09
GPT-6 (high)	81.55	75.00	72.73	73.81	77.88	79.33	85.80	90.00	83.64	91.23	82.02	86.79	100.00	80.49	83.05
Human	73.16	50.00	61.70	40.00	71.60	55.56	70.42	80.00	89.47	66.67	87.32	80.95	90.91	80.00	82.14
Ours	85.48	100.00	79.59	74.32	83.88	90.23	90.63	80.00	87.27	91.23	84.10	83.45	80.00	87.94	82.95

D.2 Qualitative Examples

Figures 6–9 present eight additional benchmark clips in the same format as Fig. 5: each row shows a video's sampled frames, the multiple-choice question and ground-truth answer, followed by every model's prediction colour-coded by correctness. The examples are drawn from all three main benchmarks and span a range of camera-motion primitives including pull-out, compound pan-push, tilt, roll, boom-up and dolly-zoom. Across these cases, most baselines struggle with motions that require distinguishing translation from zoom or resolving rotation direction, whereas CameraVLM consistently produces the correct answer. The full video for the first clip of Fig. 5 (the truck-left shot) is included in the supplementary material.

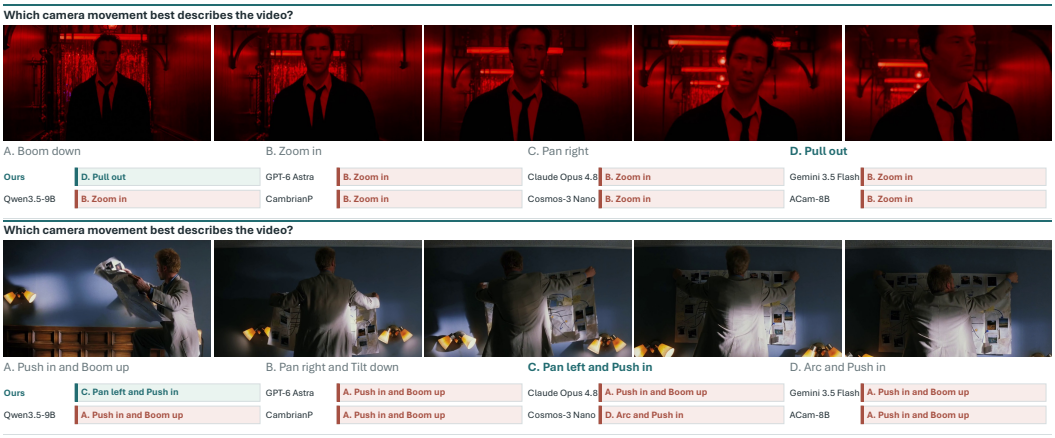

Figure 6: Qualitative examples on pull-out and compound pan-left-push-in motions. Most baselines confuse pull-out with zoom-in; only CameraVLM correctly identifies both clips.

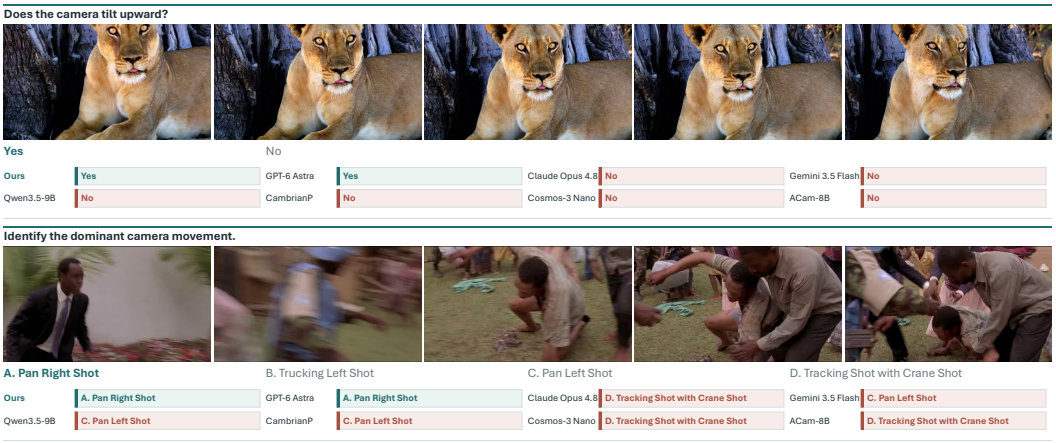

Figure 7: Qualitative examples on tilt-up and pan-right recognition. Most baselines fail on the binary tilt question; several confuse pan direction or hallucinate a crane motion.

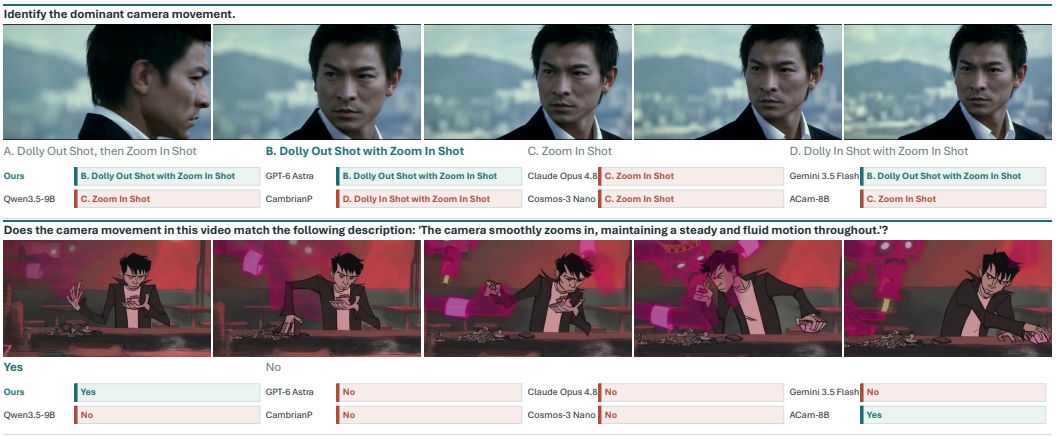

Figure 8: Qualitative examples on compound dolly-zoom and zoom-in description matching. The dolly-out-with-zoom-in clip tests fine-grained compound-motion parsing; the description-matching clip requires verifying motion smoothness.

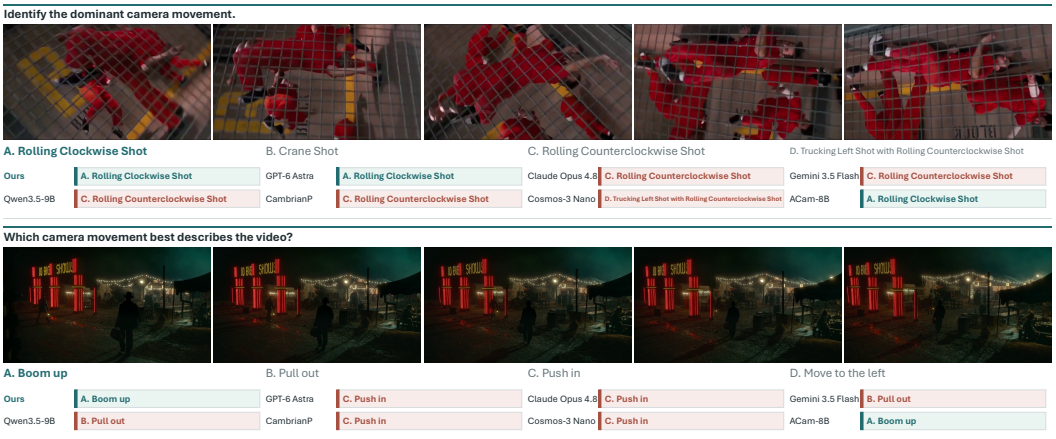

Figure 9: Qualitative examples on rolling and boom-up motions. Rolling-clockwise requires distinguishing rotation direction; boom-up is commonly misread as push-in or pull-out by baselines that lack vertical-translation awareness.