SPARSE TRAJECTORY TOKENS FOR METRIC MOTION QUERIES IN MULTIMODAL LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video-language models can recognize motion yet struggle to answer questions about metric displacement and motion rate when scale and camera motion must be inferred from images. We study an interface that supplies an MLLM with externally reconstructed 3D object and camera trajectories alongside video. Our representation associates sparse, part-aware keypoints with metric world-frame tracks, splits tracks at occlusions and motion events, and encodes adaptively fitted spline samples as trajectory tokens. A lightweight projector injects these tokens into Qwen3-VL-8B without modifying its vision encoder. We also construct TrajQA-B from Stereo4D tracks, with token-side answer checks and controls for text and option shortcuts. On its 7,500 four-choice numeric questions, the trajectory model obtains 82.2% accuracy, versus 48.3% for a separately fine-tuned frames-only reference without timestamps or guaranteed query frames. Replacing trajectories with those of another scene lowers accuracy to 29.6%; the model's generated metric evidence also tracks the reference positions and motion quantities. These results show dependence on supplied trajectories for these QA tasks. Caption and QA training exclude the evaluation scenes; source-video separation for caption training has not been audited. The benchmark uses reconstructed rather than independently measured motion references.

1 INTRODUCTION

Understanding a dynamic scene requires reasoning about where objects are in three dimensions and how their positions change over physical time. A question such as *How far did the person move between these frames?* needs metric scale, object identity, temporal correspondence, and separation of object motion from camera motion. These quantities matter for embodied agents, video search, and assistive scene understanding, but can be ambiguous in a short, sparsely sampled video. Contemporary multimodal language models (MLLMs) remain weak on spatial and spatiotemporal questions (Yang et al., 2025; Jia et al., 2026; Li et al., 2026a). Figure 1 shows the particular difficulty of recovering a numerical motion quantity from frames alone.

Most spatial MLLMs acquire geometric knowledge through images, depth, or video instruction tuning (Chen et al., 2024; Cheng et al., 2024; Cai et al., 2025). Recent dynamic-scene models also incorporate reconstructed features or 4D questions (Fan et al., 2026; Yin et al., 2026). Yet the model must still infer and retain the queried world-frame trajectory from a limited visual context. Conversely, 3D reconstruction can supply explicit point tracks, but naively listing hundreds of points across tens of frames overwhelms the language context and obscures which motion belongs to which object. This leaves an interface problem: how can an MLLM read compact, metric, object-grounded motion without having to recover it again from pixels?

We propose *Sparse Trajectory Tokens*, an interface that turns externally reconstructed 4D motion into an ordered sequence of physical-time trajectory records. The central idea is to preserve the information needed for a motion query, including object and part identity, visible world-frame positions, time, and motion discontinuities, while spending tokens only where the track changes. This is a conditional setting: reconstruction is supplied at inference, and its accuracy bounds the answers the model can give.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

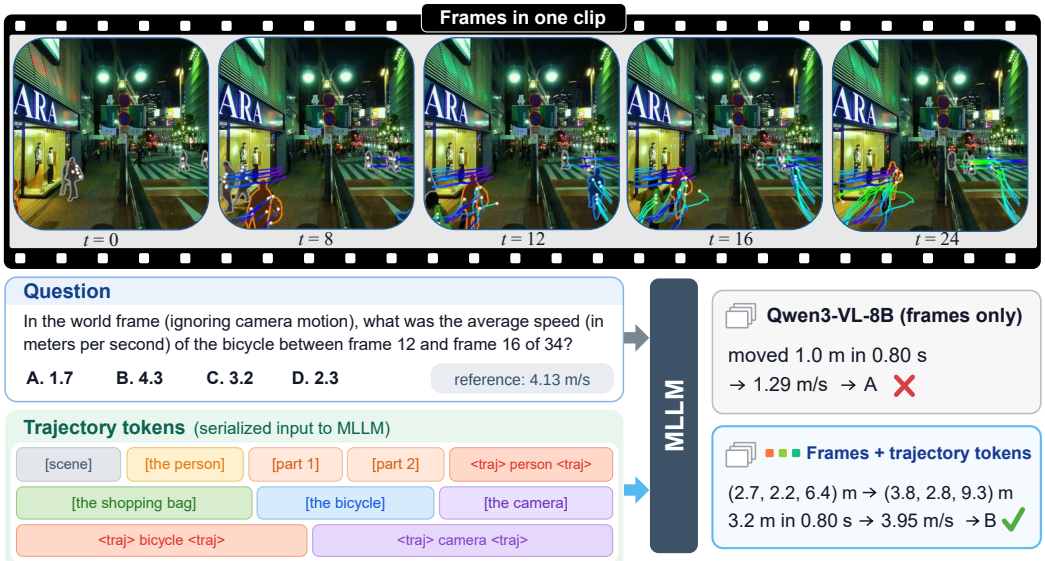


Figure 1: **Metric motion is hard to read from sampled frames.** Five frames and reconstructed 3D tracks accompany a question whose legacy *average speed* wording is scored as endpoint displacement per elapsed second. Separately fine-tuned Qwen3-VL-8B arms predict 1.29 m/s from frames and 3.95 m/s from frames plus trajectory tokens; the reconstruction-derived reference is 4.13 m/s. The example illustrates the input setting, not an isolated causal effect of any one tokenization component.

The method has three connected modules (Fig. 2). The *Object-Centric Track Constructor* assigns reconstructed 3D tracks to segmented objects and chooses sparse, part-aware keypoints, separating object and camera motion. The *Event-Aware Trajectory Tokenizer* normalizes the scene, preserves visibility gaps, splits salient changes, and adaptively fits piecewise curves whose on-curve samples become time-stamped tokens. The *Trajectory-Language Adapter* embeds the records directly into an existing MLLM and trains it to describe motion and answer questions using the supplied values. Together they connect dense reconstruction to metric language output under a bounded token budget.

We evaluate this interface with TrajQA-B, a benchmark constructed from Stereo4D reconstructions (Jin et al., 2025). The questions ask for object or camera displacement and endpoint displacement per elapsed second, with answers checked against the serialized tokens. On 7,500 four-choice numeric questions, our trajectory-input model reaches 82.2% accuracy; a separately trained frames-only arm reaches 48.3%. Replacing the correct trajectories with another scene's records reduces accuracy to 29.6%, and the generated numerical evidence becomes substantially less accurate. These results support effective reading of supplied trajectories. They do not establish the individual benefit of the spline, event, or part choices, for which controlled component ablations are still needed.

Contributions. (1) We formulate an object-centred, part-aware interface from dense reconstructed 4D tracks to sparse world-frame motion records. (2) We introduce event-aware adaptive trajectory tokens and a direct token-to-language embedding adapter that retain metric scale and physical time within a compact sequence. (3) We construct TrajQA-B with token-side answer checks and report a separately trained frames-only reference, same-model input interventions, and metric evidence diagnostics that delimit what the model has learned.

2 Related Work

2.1 Spatial and spatiotemporal reasoning in MLLMs

SpatialVLM studies how to equip image-language models with quantitative spatial knowledge, while SpatialRGPT grounds spatial answers in regions and SpatialBot extends the input toward RGB-D geometry (Chen et al., 2024; Cheng et al., 2024; Cai et al., 2025). Scene-LLM projects camera- and world-coordinate 3D scene features into an LLM using staged alignment and instruction tuning (Fu et al., 2025). These approaches make depth, position, and object relationships more accessible to language generation. Video adds another difficulty: an object can change position while the camera moves, and its apparent displacement may not equal its motion in the world. VSI-Bench, OmniSpatial, and ViSTR-Bench document persistent shortcomings on spatial and spatiotemporal queries (Yang et al., 2025; Jia et al., 2026; Li et al., 2026a). Dynamic-scene methods attempt to narrow this gap. VG-LLM uses geometry-oriented video supervision; VLM-3R links reconstruction and language learning; MLLM-4D uses instruction tuning for time-varying spatial relationships (Zheng et al., 2025b; Fan et al., 2026; Yin et al., 2026). Their representation, supervision, and test questions differ, so their results do not directly answer whether a language model can perform a numerical motion query when world-frame tracks are already available. Our setting isolates that interface: it gives the MLLM reconstructed object and camera trajectories and asks it to read metric positions and elapsed times from them. This is complementary to methods that infer geometry from video. It also makes the dependence on the reconstruction explicit; strong performance here cannot by itself establish accurate motion recovery from raw pixels. The frames-only arm in our experiments tests the practical value of this extra input on the same questions, while donor-token and question-only controls ask whether the answer actually depends on the supplied trajectory values.

2.2 Dynamic 3D reconstruction and object tracks

A broad line of work estimates correspondences and geometry across a moving camera and moving objects. Stereo4D mines world-consistent, pseudo-metric point tracks from internet stereo video and provides the reconstructed trajectories on which our data and reference answers are based (Jin et al., 2025). Methods such as TAPIP3D and SpatialTrackerV2 pursue persistent 3D point tracking under their respective input assumptions (Zhang et al., 2025a; Xiao et al., 2025). Track4World predicts world-centric dense 3D tracks from monocular video (Lu et al., 2026); our broader pipeline can use it when released tracks are absent, but every downstream TrajQA-B result in this paper uses Stereo4D's provided tracks. This distinction matters because a better tracker could change both the input and the reconstruction-derived target. Dense tracks are also not an object-level language interface: they can contain many redundant surface points, change visibility across time, and mix rigid camera motion with independently moving objects. We therefore use masks to attribute tracks to instances, select representative points for their geometry or semantic parts, and retain a separate camera trajectory. The object inventory and part assignment may themselves be imperfect, which is why the benchmark gates retain only cases whose answers agree when recomputed from the serialized input. Our contribution begins after motion reconstruction: it organizes already estimated tracks into a compact, queryable sequence. We do not claim that this organization improves the underlying 3D estimates, and our evaluation inherits the reconstruction's scale, occlusion, and identity errors. This places our method between 4D reconstruction and language reasoning rather than presenting it as a replacement for either.

2.3 Trajectory representations and language interfaces

Continuous trajectories pose a representation choice: uniform frame sampling preserves a simple temporal grid but spends tokens on nearly redundant positions and may skip a brief change; quantization and learned codes can reduce sequence length but may obscure metric values. One Trajectory, One Token organizes video features around panoptic sub-object tracks and connects the encoder to VideoLLM QA (Zheng et al., 2025a). TrajTok adapts spatial tokenization of GPS traces (Xiong et al., 2026), while LLaMo represents human motion with continuous autoregressive latents (Li et al., 2026b). MotionLM and AutoTraces use language-model mechanisms for motion forecasting (Seff et al., 2023; Wang et al., 2026). These works mostly encode or generate motion, whereas ours supplies reconstructed metric motion to answer queries from recoverable numeric records. We

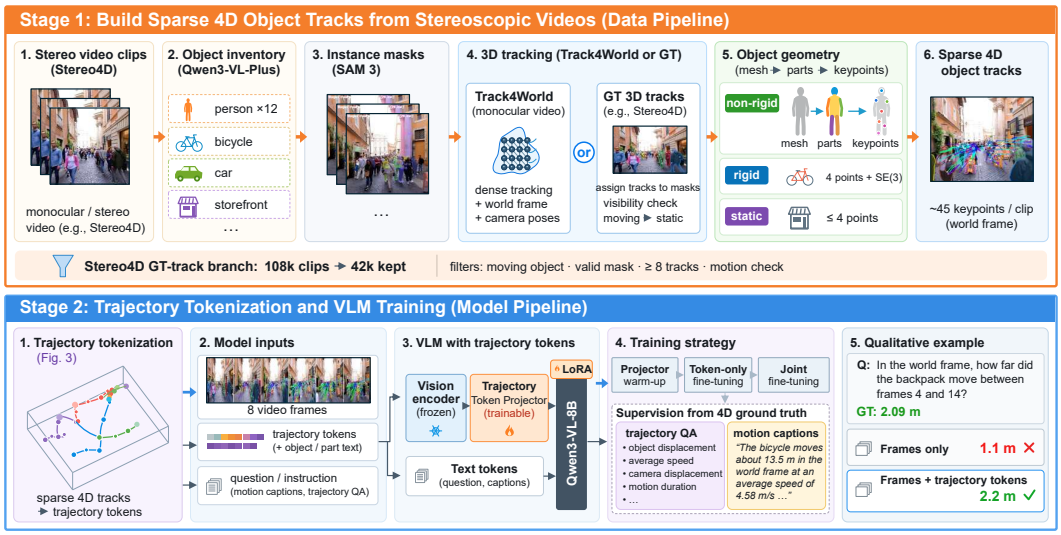


Figure 2: **Pipeline with real video frames and reconstructed tracks.** The upper row implements the Object-Centric Track Constructor; the lower row links the Event-Aware Trajectory Tokenizer to the Trajectory-Language Adapter and its three training stages. The reported QA uses released Stereo4D tracks. In the supplied artwork, "GT" means released reconstructed tracks, "ground truth" means reconstruction-derived references, and "average speed" means endpoint displacement rate. Track4World is an optional branch not used in the reported QA.

fit each visible keypoint run with a bounded-error piecewise curve, allocate more nodes where the path changes, and serialize on-curve samples with object, part, physical-time, and event fields. The representation also carries the normalization parameters, so a model can in principle reconstruct metre-scale positions rather than relying on relative motion codes. A learned projector maps each record to the language model's embedding space without changing its visual encoder; relations among records are learned inside the language model. This direct interface is related to trajectory tokenization in its budget goal, but differs in its emphasis on object identity, visibility gaps, camera motion, and verifiable metric answers. Our current intervention experiments establish dependence on correct token content as a whole. They do not yet isolate the necessity of adaptive fitting, part labels, or event flags, which remain specific component-level hypotheses for further ablation.

3 METHOD

Problem definition. Let a clip have T frames $I_1, \dots, I_T$ at physical times $t_1 < \cdots < t_T$, externally reconstructed world-frame tracks $\{\boldsymbol{p}_k(t_i), v_k(t_i)\}_{k,i}$, and camera poses $\boldsymbol{C}_i \in SE(3)$. Here $\boldsymbol{p}_k(t_i) \in \mathbb{R}^3$ is point k's position in metres and $v_k(t_i) \in \{0, 1\}$ denotes visibility. Given a question q about an object or camera over frames $a < b$, the model predicts an answer y from the frames, tracks, and q. Object displacement is $D_{o,a:b} = \|\boldsymbol{p}_o(t_b) - \boldsymbol{p}_o(t_a)\|_2$, where $\boldsymbol{p}_o(t)$ is the per-frame median of its visible selected keypoints; the benchmark's *speed* is $D_{o,a:b}/(t_b - t_a)$ (the magnitude of average velocity). The targets inherit reconstruction error because the tracks are their reference.

Pipeline overview. The *Object-Centric Track Constructor* converts dense tracks and masks to sparse object- and part-indexed paths. The *Event-Aware Trajectory Tokenizer* compresses visible paths into records that preserve position, physical time, visibility, and local motion changes. The *Trajectory-Language Adapter* maps those records into Qwen3-VL-8B's input embeddings (Bai et al., 2025) and trains motion description and QA. Figure 2 follows these three named modules and Fig. 3 expands the middle one. Benchmark question generation is described in Appendix D; the three-stage training schedule appears in Appendix E.

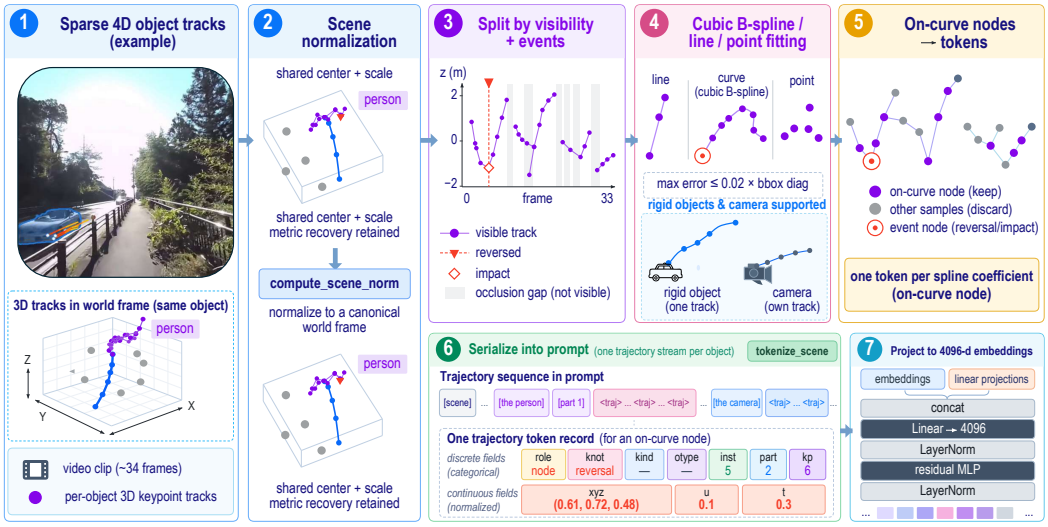


Figure 3: **From a keypoint track to trajectory tokens** (one pedestrian keypoint). The scene is normalized; visible runs are split at motion events; spline pieces are fitted with a target error of 0.02 times the trajectory bounding-box diagonal. Fitted on-curve nodes are serialized by object and part with text labels and projected into the LLM embedding space. A small fraction of pieces cannot meet the target under the node budget (Appendix C).

3.1 Object-Centric Track Constructor

Figure 2 (top) summarizes how a video clip is converted into *sparse 4D object tracks*. We first obtain per-object masks and dense 3D tracks with visibility and camera poses. All reported TrajQA-B results use Stereo4D's released tracks; Track4World (Lu et al., 2026) is an optional path for other monocular videos that is not evaluated in the downstream QA results. For non-rigid objects, SAM 3D Objects (SAM 3D Team et al., 2025) reconstructs an object mesh that is partitioned into semantic parts by PartSAM (Zhu et al., 2026); representative surface keypoints are then selected and associated with the corresponding dense tracks. The complete extraction pipeline, including object discovery with Qwen3-VL-Plus, mask propagation with SAM 3 (Carion et al., 2026), mesh reconstruction, and keypoint sampling and association, is detailed in Appendix B.

The resulting representation is object-centric, part-aware, and metric. A non-rigid object uses sparse keypoints assigned to semantic parts; a rigid object uses four reference keypoints and a per-frame pose $(\boldsymbol{R}_i, \boldsymbol{t}_i) \in SE(3)$ estimated from its tracks; and a static object uses up to four points. For object o with K_o keypoints, the module returns positions $\mathbf{X}_o \in \mathbb{R}^{K_o \times T \times 3}$, visibility $\boldsymbol{V}_o \in \{0,1\}^{K_o \times T}$, a part index for each keypoint, and a text label. Here T is the number of frames, $\boldsymbol{R}_i \in SO(3)$ is a rigid rotation, and $\boldsymbol{t}_i \in \mathbb{R}^3$ is its translation. Camera poses and timestamps are stored alongside the object tracks. All coordinates are in the reconstructed world frame, so object and camera motion can be distinguished. A scene contains 44.8 selected keypoints on average (20.7 on moving objects), rather than hundreds of dense tracks per object.

3.2 Event-Aware Trajectory Tokenizer

Normalization and segmentation. All visible keypoints of a scene are shifted by their centroid $\boldsymbol{\mu} \in \mathbb{R}^3$ and scaled by the maximum radius $s > 0$ into the unit sphere, $\tilde{\boldsymbol{p}} = (\boldsymbol{p} - \boldsymbol{\mu})/s$; $(\boldsymbol{\mu}, s)$ are emitted as the scene header, so a metric position is recoverable as $\boldsymbol{p} = s\tilde{\boldsymbol{p}} + \boldsymbol{\mu}$. A track is fitted only on its maximal visible runs: an occlusion gap is recorded explicitly rather than silently interpolated. Within a run we detect motion *events* on the inter-frame displacements: a direction change exceeding 90° when both steps exceed a noise floor (called a reversal in the token vocabulary), or a peak in discrete acceleration (called an impact). We split the run at these event frames. Consecutive pieces include the same observed event frame, but separate smoothing fits need not interpolate it exactly.

Splitting permits a local change of direction or acceleration that a single global smooth fit might attenuate; its benefit for QA remains to be isolated experimentally.

Target-error fit. For observed frames $f_0 < \cdots < f_{m-1}$, let $u_i = (f_i - f_0)/(f_{m-1} - f_0)$ and fit a cubic B-spline to normalized positions $\tilde{\boldsymbol{p}}_i$:

$$\boldsymbol{B}(u) = \sum_{h=1}^{n} N_{h,3}(u; \boldsymbol{\xi})\, \boldsymbol{\beta}_h, \qquad \max_i \|\boldsymbol{B}(u_i) - \tilde{\boldsymbol{p}}_i\|_2 \le \varepsilon = \alpha d, \quad \alpha = 0.02. \tag{1}$$

Here $n \le 12$ is the coefficient count, $N_{h,3}$ is the cubic basis on knot vector $\boldsymbol{\xi}$, $\boldsymbol{\beta}_h \in \mathbb{R}^3$ is a control point, and d is the bounding-box diagonal of the object's normalized visible path. Adaptive knots allocate more nodes to harder motion. The sampled bound holds for 99.82% of pieces; the rest retain their best fit and are counted as failures (Appendix C).

On-curve nodes as tokens. For knot values ξ_l, the Greville parameter $\tau_h = (\xi_{h+1} + \xi_{h+2} + \xi_{h+3})/3$ yields the on-curve sample $\boldsymbol{B}(\tau_h)$, unlike an off-curve spline control point. We serialize one node per coefficient with its mapped physical timestamp and a smooth, event, or endpoint flag. The question generator linearly interpolates these records to obtain a decoded path $\hat{\boldsymbol{p}}(t)$ and checks its answers against the references (Gate 0). The LLM does not receive the knot vector, so the fit bound in Eq. 1 is not an error bound on $\hat{\boldsymbol{p}}(t)$; separate event-side fits may also disagree at their shared time. Appendix C details this distinction. Rigid objects fit translation and rotation jointly on shared knots; their keypoints follow $\boldsymbol{p}_j(t) = \boldsymbol{R}_t \tilde{\boldsymbol{r}}_j + \boldsymbol{c}(t)$, where $\tilde{\boldsymbol{r}}_j \in \mathbb{R}^3$ is a reference point, $\boldsymbol{R}_t \in SO(3)$ is rotation, and $\boldsymbol{c}(t) \in \mathbb{R}^3$ is translation. The camera is appended as a rigid pseudo-object.

Serialization. Each record has seven discrete fields (role, knot type, kind, VLM type, instance, part, keypoint) and three continuous fields (3D value, normalized clip time, absolute seconds). A scene header stores $(\boldsymbol{\mu}, s)$; objects follow with text category and part labels adjacent to their trajectory placeholders, and the camera comes last. A clip uses a median of 165 trajectory tokens against 1,152 visual tokens from eight frames. Training caps the stream at 512 records while keeping structural and event nodes.

3.3 Trajectory-Language Adapter

Trajectory Token Projector. The projector (Fig. 3, right) maps every token independently. Let $d_1, \ldots, d_7$ denote its seven discrete field values; $\boldsymbol{x} \in \mathbb{R}^3$ is its position, rotation, reference point or gap value, $u \in [0, 1]$ is normalized clip time, and t is physical time in seconds. Learned tables $\boldsymbol{E}_k$ embed discrete fields, learned matrices $\boldsymbol{W}_{\boldsymbol{x}}, \boldsymbol{W}_u, \boldsymbol{W}_t$ project continuous fields, and binary gates $g_{\boldsymbol{x}}, g_u, g_t$ mark fields present in this token. A learned fusion matrix $\boldsymbol{W}_{\mathrm{f}}$ and bias $\boldsymbol{b}_{\mathrm{f}}$ precede layer normalization (LN) and a residual multilayer perceptron (MLP):

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{W}_{\mathrm{f}}\big[\boldsymbol{E}_1(d_1); \ldots; \boldsymbol{E}_7(d_7);\ g_{\boldsymbol{x}}\boldsymbol{W}_{\boldsymbol{x}}\boldsymbol{x};\ g_u\boldsymbol{W}_u u;\ g_t\boldsymbol{W}_t t\big] + \boldsymbol{b}_{\mathrm{f}}, \\ \boldsymbol{z} &= \mathrm{LN}\big(\boldsymbol{h}' + \mathrm{MLP}(\boldsymbol{h}')\big), \qquad \boldsymbol{h}' = \mathrm{LN}(\boldsymbol{h}), \end{aligned} \tag{2}$$

with 85.93M parameters in total. Here $\boldsymbol{h}$ is the fused field representation, $\boldsymbol{h}'$ its normalized value, and $\boldsymbol{z}$ the output token embedding. It contains no attention, convolution or pooling, so every relation between trajectories is formed inside the LLM. Its outputs replace the placeholder embeddings by a masked scatter into the LLM's input embeddings; trajectory tokens are otherwise treated as text (ordinary 1-D positions on all rotary axes) and never pass through the vision encoder.

The backbone is Qwen3-VL-8B-Instruct with frozen vision and base weights. We train the projector and language-model LoRA adapters (Hu et al., 2021) by next-token cross-entropy on generated evidence and answer text. Stage 1 warms up the projector on rule-generated motion captions with the base model frozen; Stage 2 learns token-only QA with the projector and LoRA; Stage 3 restores frames and drops them independently with probability 0.2 while keeping tokens. This sequence first teaches the new interface to express motion and then adapts it to metric questions. All three training stages exclude QA evaluation scenes; stages 2 and 3 additionally use source-video-disjoint QA splits. A stage-1 source-video audit remains to be documented (Appendix E).

4 EXPERIMENTS

Dataset and evaluation. TrajQA-B draws on 42,151 curated Stereo4D scenes and uses source-video-disjoint QA splits: 56,120 training, 1,318 validation, 4,492 test, and 3,430 additional held-out questions (Table 4). Seven question families cover camera and object displacement, endpoint displacement per elapsed second, interval comparisons, motion source, and duration (Appendix D). The three four-choice numeric families contribute 7,500 evaluation questions and have 25.0% uniform-guessing accuracy. In every table *speed* means endpoint displacement divided by elapsed time, rather than path length per time. Gate 0 retains questions whose reference answer a deterministic reader can reproduce from the capped records; this tests record reading and calculation, not reconstruction accuracy. Option ranks and binary answers are balanced, then audited against question-only heuristics. The separately selected stage-1 caption training scenes also exclude QA evaluation scenes; its source-video overlap has not been audited (Appendix E). Appendix F gives the full protocols and denominators.

Implementation details. Our backbone is Qwen3-VL-8B-Instruct with frozen vision and base language weights. The trainable trajectory projector has 85.93M parameters; language-model LoRA adapters have rank 128, scale 256, dropout 0.05, and 349.18M parameters. Each sample contains eight uniformly sampled video frames, resized to a longest side of 384 pixels (144 visual tokens per frame), the object-labelled trajectory stream capped at 512 records, and a question or instruction. We use three stages: 26,000-step projector warm-up on 64,523 rule-generated motion captions from a separately selected caption corpus; 8,400-step projector-plus-LoRA QA training with trajectory tokens but no frames; and 3,507-step joint QA training with frames restored and independently dropped with probability 0.2. All stages use AdamW with $\beta = (0.9, 0.95)$, zero weight decay, gradient clipping at 1.0, global batch size 16, and seed 0. The peak projector learning rate is 2×10^{-4} throughout; the adapters use 10^{-4} in stages 2 and 3. We train on the assistant's evidence and answer text with next-token cross-entropy and decode greedily. The frames-only Qwen3-VL-8B control was fine-tuned on the same QA training split, although its exact schedule and checkpoint selection are unavailable in the manuscript. Most external baselines are zero-shot and some use their official video pipelines or a smaller proprietary-model subset; they contextualize performance rather than isolate the effect of trajectory input. Stage-1 captions use metadata frame rate while QA targets use frame timestamps, a source of noisy time labels described in Appendix A.

4.1 COMPARISONS ON TRAJQA-B

Table 1 places our model next to 15 external baselines. Open-source and specialised 3D models score between 25.0% and 29.4% on the pooled numeric questions; the two proprietary models reach 39.2% and 35.2% on their smaller subset. Comparisons from 7B to 38B do not show a clear gain from parameter count in this collection of models, although architectures and video protocols also vary. MLLM-4D-SFT scores 25.0% after training on a separate 4D QA corpus drawn from the same source videos; 85% of evaluation clips appear in its SFT data. Its strong option-position preference provides a plausible explanation for the poor transfer to our balanced options (Appendix F). Our fine-tuned model reaches 82.2% overall, with 81.7% on held-out and 82.5% on test. A frames-only Qwen3-VL-8B arm trained on the same question split reaches 48.3% (Table 2). This task-adapted visual reference lacks timestamps and may omit the queried endpoint frames; its undocumented training schedule also prevents a controlled modality comparison.

4.2 INPUT ABLATION STUDIES

Trajectory content changes the answers. The same final model scores 26.8% with the question alone and 29.6% with frames plus another scene's tokens, versus 82.2% with the correct tokens (Table 2). Both controls remain above uniform guessing, so they do not establish the complete absence of shortcuts; their large gap from the correct-token condition shows that trajectory content matters. The separately trained frames-only arm reaches 48.3%, roughly 34 points below the full model. On held-out questions, the donor-token intervention lowers accuracy from 81.7% to 30.6%. The final model without frames also scores 82.2% (Appendix F), indicating that this benchmark mainly probes use of supplied metric tracks rather than added value from fusing frames with them.

Table 1: TrajQA-B numeric accuracy (%). Numeric-3 pools the three four-choice families ($n = 7{,}500$; guessing: 25.0); split and family scores are also shown. External models are zero-shot except MLLM-4D-SFT, while ours is fine-tuned on the TrajQA-B training split. Proprietary models use a 1,000-question subset. Bold marks the largest displayed score in each column; protocols and subsets differ. Full results appear in Appendix F.

Model	Numeric-3↑	Held-out↑	Test↑	Cam. disp.↑	Obj. disp.↑	Rate↑
Proprietary models (1,000-question subset)						
GPT-6 Astra	39.2	37.9	40.5	46.0	32.7	33.9
Claude Fable 5.1	35.2	35.8	34.5	39.6	35.2	26.9
Open-source base models, 7B and 8B						
Qwen2.5-VL-7B	27.3	26.6	27.9	27.1	27.7	27.3
Qwen3-VL-8B	28.1	28.5	27.7	28.9	27.9	26.9
LLaVA-NeXT-Video-7B	25.0	24.5	25.4	25.7	24.2	24.7
InternVideo2.5-8B	25.5	25.5	25.5	25.5	24.9	26.1
InternVL2.5-8B	27.0	27.3	26.8	27.7	27.4	25.5
InternVL3.5-8B	26.8	25.8	27.5	26.3	26.4	27.9
Open-source base models, 32B						
Qwen2.5-VL-32B	28.4	28.3	28.5	29.7	27.3	27.5
Qwen3-VL-32B	27.8	27.8	27.7	27.9	25.9	29.5
Open-source base models, 38B						
InternVL2.5-38B	29.4	29.3	29.5	30.3	29.9	27.5
InternVL3.5-38B	28.5	28.9	28.3	29.4	27.5	28.2
3D spatial-reasoning models						
VG-LLM (Qwen2.5-VL-7B)	25.2	24.6	25.7	24.8	25.4	25.6
VLM-3R (LLaVA-Video-7B)	26.9	26.9	27.0	27.1	27.0	26.7
4D instruction-tuned model (Stereo4D source)						
MLLM-4D-SFT (Qwen3-VL-8B)	25.0	24.7	25.3	25.3	24.9	24.8
TrajQA-B fine-tuned model						
Ours (frames + trajectory tokens)	**82.2**	**81.7**	**82.5**	**89.4**	**77.6**	**75.2**

Table 2: Input interventions on the full evaluation set. All rows except frames only use the same final model; that row uses a separately fine-tuned visual model. Bold marks the best value in each column, including ties. Donor tokens replace trajectory content with that of another scene while preserving format, count, and timestamps. Cam./Obj.: camera/object displacement; rate: endpoint displacement per elapsed second. All results use one training seed.

Inputs at inference	Frames	Tokens	Numeric-3↑	Held-out↑	Test↑	Cam.↑	Obj.↑	Rate↑
Question only	No	No	26.8	27.5	26.2	27.2	27.0	25.9
Frames only (separate FT)	✓	No	48.3	47.8	48.8	53.7	44.5	43.9
Frames + donor tokens	✓	donor	29.6	30.6	28.8	30.7	28.8	28.6
Trajectory tokens only	No	✓	**82.2**	81.5	**82.8**	**89.4**	**77.7**	**75.5**
Frames + traj. tokens (full)	✓	✓	**82.2**	**81.7**	82.5	**89.4**	77.6	75.2

Generated evidence tracks metric quantities. On held-out questions, the positions stated in the model's evidence differ from the reference by a median of 0.33 m for objects and 0.14 m for the camera (Table 3). Its stated displacement or speed lies within a factor of 1.2 of the reference on 80% to 91% of questions. With donor tokens, median position errors rise to metres and these rates fall to 16% to 18%. The evidence analysis supports use of the trajectory values beyond selecting an option, subject to the reference-track quality and parsing protocol in Appendix F. Qualitative examples appear in Appendix G.

Module-facing diagnostics. Selected scenes average 44.8 keypoints; the tokenizer uses a median of 165 records and its intermediate spline fits meet the target on 99.82% of pieces. Donor substitution probes use of correct records. All three main numeric families can be answered from endpoint

Table 3: Metric grounding of generated evidence on held-out numeric questions ($n = 3{,}312$). Acc.: accuracy; E_{obj} / E_{cam}: median position error. Bold marks the best value for each metric, including ties. Within $1.2\times$: fraction whose stated camera displacement / object displacement / endpoint rate differs from the reference by less than a multiplicative factor of 1.2.

Inputs at inference	**Acc.↑**	$\boldsymbol{E_{\mathrm{obj}}}$ **(m)↓**	$\boldsymbol{E_{\mathrm{cam}}}$ **(m)↓**	**Within $1.2\times$ ↑: cam / obj / rate**
Question only	27.5	3.56	0.83	21.0 / 15.5 / 14.9
Frames + donor tokens	30.6	3.44	1.35	18.2 / 17.7 / 16.0
Trajectory tokens only	81.5	**0.32**	**0.14**	**91.4 / 80.0 / 80.3**
Frames + trajectory tokens (full)	**81.7**	0.33	**0.14**	91.1 / **80.0** / 79.9

positions and times, so these diagnostics do not establish a benefit from part labels, event splitting, adaptive fitting, or projector warm-up.

Where the improvement applies. Camera displacement is the easiest numeric family for the full model (89.4%), followed by object displacement (77.6%) and endpoint rate (75.2%). The final model has the same pooled score with tokens alone as with tokens and frames (82.2%), so these results demonstrate use of reconstructed motion, not added value from visual fusion. The 81.7% held-out and 82.5% test scores show similar performance across the two QA splits, but stage-1 source-video overlap has not been audited. Because donor input remains above 25% chance and the visual arm lacks queried timestamps, neither control rules out all shortcuts or supplies a fully matched visual comparison.

Evaluation composition. Table 4 shows the family denominators and the number of scenes and source videos in each split. The held-out and test splits contain 3,292 camera-displacement, 2,172 object-displacement, and 2,036 endpoint-rate questions; these are the 7,500 questions underlying the pooled numeric score. The other four families contribute 422 evaluation questions in total. Their reported aggregate in Appendix F is therefore a smaller, heterogeneous diagnostic, not evidence that the proposed event and part representations improve interior-motion reasoning.

Table 4: TrajQA-B split counts (descriptive, not performance metrics). Splits are disjoint in source videos. Cam./Obj./Rate count the three 4-way families (chance 25%); Other pools interval_disp, interval_speed, motion_source, and binary motion_duration. Rate denotes endpoint displacement divided by elapsed time.

Split	**Questions**	**Cam.**	**Obj.**	**Rate**	**Other**	**Scenes**	**Videos**
train	56,120	21,240	16,492	14,980	3,408	12,409	2,198
validation	1,318	524	404	340	50	668	162
dev test	4,492	1,692	1,284	1,212	304	2,241	578
held-out test	3,430	1,600	888	824	118	862	166

Residual data risks. Because 98.4% of clips share a source video, the unaudited stage-1 source-video intersection remains consequential. Caption frame rates differ from QA timestamps by more than 10% in 12.85% of scenes, potentially adding rate-label noise.

5 Conclusion

Sparse Trajectory Tokens turn reconstructed tracks into compact metric records for an MLLM. TrajQA-B accuracy is 82.2% with correct records and 29.6% with donor records; a separate frames-only reference scores 48.3%. Token-only accuracy is also 82.2%, so frames add no measured benefit here. Endpoint questions cannot isolate tokenization components. Generalization to unseen source videos, component effects, and physical validity need a stage-1 source-video audit, matched visual inputs, ablations, and independent ground truth.

AI USE STATEMENT

Generative AI tools were used solely for grammar correction and minor language polishing of text written by the authors. No generative AI tools were used for any research tasks requiring disclosure under the ICLR 2027 AI Policy. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The appendix describes the data source, QA split rule, curation gates, token fields, spline fitting, question generator, and trajectory-model schedules. Exact reproduction additionally requires the Stereo4D reconstructions, training and evaluation code, model checkpoints, the frames-only training recipe, and the donor-token mapping; the latter two are not documented in the available manuscript. Reported trained-model results use one seed.

ETHICS STATEMENT

The source videos may include identifiable people. Our analysis uses the Stereo4D reconstructions and selected illustrative frames; any redistribution of videos or derived visual assets should follow the source permissions and applicable privacy requirements. Automatic curation favors objects that can be segmented and tracked reliably, so the retained scenes should not be treated as a representative sample of all activities or environments.

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, November 2025.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. SpatialBot: Precise Spatial Understanding with Vision Language Models, March 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/pdf/c01d0dd8d4e7179b0952062eff08eb52c8d8c322.pdf.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14455–14465, Seattle, WA, USA, June 2024. IEEE. ISBN 979-8-3503-5300-6. doi: 10.1109/CVPR52733.2024.01370.

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, October 2024.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31054–31065, 2026.

Weilun Feng, Mingqiang Wu, Zhiliang Chen, Chuanguang Yang, Haotong Qin, Yuqi Li, Xiaokun Liu, Guoxin Fan, Libo Huang, Yulun Zhang, Michele Magno, Yongjun Xu, and Zhulin An. Fast-SAM3D: 3Dfy Anything in Images but Faster, June 2026.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual reasoning. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 2195–2206, 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Fu_Scene-LLM_Extending_Language_Model_for_3D_Visual_Reasoning_WACV_2025_paper.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021.

Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, XinQiang Yu, Jiawei He, He Wang, and Li Yi. OmniSpatial: Towards Comprehensive Spatial Reasoning Benchmark for Vision Language Models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=6nZKT2rL0H.

Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4D: Learning How Things Move in 3D from Internet Stereo Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10497–10509, 2025.

Han Li, Si Liu, Zehao Huang, Dongxin Lyu, Longfei Xu, Jiahui Fu, Daxin Tian, Yuliang Xiu, and Naiyan Wang. ViSTR-Bench: Can MLLMs Reason from Continuous Visual Cues in Dynamic Scenes?, July 2026a.

Zekun Li, Sizhe An, Chengcheng Tang, Chuan Guo, Ivan Shugurov, Linguang Zhang, Amy Zhao, Srinath Sridhar, Lingling Tao, and Abhay Mittal. Llamo: Scaling pretrained language models for unified motion understanding and generation with continuous autoregressive tokens, 2026b. URL https://arxiv.org/abs/2602.12370.

Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4World: Feedforward World-centric Dense 3D Tracking of All Pixels, March 2026.

Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4RC: 4D Reconstruction via Conditional Querying Anytime and Anywhere, May 2026.

SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam 3d: 3dfy anything in images, 2025. URL https://arxiv.org/abs/2511.16624.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. MotionLM: Multi-Agent Motion Forecasting as Language Modeling. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8545–8556, Paris, France, October 2023. IEEE. ISBN 979-8-3503-0718-4. doi: 10.1109/ICCV51070.2023.00788.

Teng Wang, Yanting Lu, and Ruize Wang. AutoTraces: Autoregressive Trajectory Forecasting via Multimodal Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4054–4064, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_AutoTraces_Autoregressive_Trajectory_Forecasting_via_Multimodal_Large_Language_Models_CVPR_2026_paper.html.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. InternVL3.5: Advancing Open-Source Multimodal Models in Versatility, Reasoning, and Efficiency, August 2025a.

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, July 2025b.

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, July 2025.

Zhen Xiong, Shang-Ling Hsu, and Cyrus Shahabi. TrajTok: Adaptive Spatial Tokenization for Trajectory Representation Learning, May 2026.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10632–10643, 2025.

Xingyilang Yin, Chengzhengxu Li, Jiahao Chang, Chi-Man Pun, and Xiaodong Cun. MLLM-4D: Towards Visual-based Spatial-Temporal Intelligence, February 2026.

Bowei Zhang, Lei Ke, Adam Harley, and Katerina Fragkiadaki. TAPIP3D: Tracking Any Point in Persistent 3D Geometry. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 135284–135303. Curran Associates, Inc., 2025a. doi: 10.52202/085713-4514.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently Reconstructing Dynamic Scenes One D4RT at a Time, December 2025b.

Chenhao Zheng, Jieyu Zhang, Mohammadreza Salehi, Ziqi Gao, Vishnu Iyengar, Norimasa Kobori, Quan Kong, and Ranjay Krishna. One trajectory, one token: Grounded video tokenization via panoptic sub-object trajectory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025a. URL https://openaccess.thecvf.com/content/ICCV2025/papers/Zheng_One_Trajectory_One_Token_Grounded_Video_Tokenization_via_Panoptic_Sub-object_ICCV_2025_paper.pdf.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from Videos for 3D World: Enhancing MLLMs with 3D Vision Geometry Priors. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 20560–20586. Curran Associates, Inc., 2025b. doi: 10.52202/085713-0693.

Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. PartSAM: A Scalable Promptable Part Segmentation Model Trained on Native 3D Data, February 2026.

A Data Curation Details

Source data and frame sampling. The official Stereo4D index lists 108,015 clips; 3D tracks are released for 89,454 of them, which form our candidate pool. Every clip is decoded from the rectified left view at 512×512 pixels and temporally subsampled from the native 30 fps with a per-clip stride $\sigma = \min\big(\max(1, \lfloor (N-1)/19 \rfloor),\, 6\big)$, where N is the number of source frames, so that a clip keeps 20–34 frames (up to 38 are accepted) at an effective rate of $30/\sigma$ fps. Over the delivered scenes, 76.9% use stride 6 (5 fps); the frame count ranges from 20 to 38 with a median of 34, and 67.6% of the scenes have exactly 34 frames. The per-frame timestamps of the release are kept with every scene. Trajectory tokens and QA targets use these timestamps; stage-1 motion captions instead use the metadata frame rate. The two time sources disagree by more than 10% in 12.85% of scenes, which introduces noise into time-dependent caption targets.

Curation steps. A clip enters the training set only if at least one object moves in the world frame and this motion is observable both in the released tracks and in the image. The steps, and the number of clips each removes, are (the funnel in Fig. 2 summarizes them as 108k $\rightarrow$ 42k):

- *Object inventory.* A VLM (Qwen3-VL-Plus) inspects 8 uniformly spaced frames and returns, for every object, a category label, a 2D box on one frame, a rigid/non-rigid attribute and a static/dynamic attribute; the prompt asks it to discount camera motion. Clips in which every object is static are discarded before any further processing (29,915).
- *Instance masks.* SAM 3 is prompted with each label and its box on the VLM's frame (text-only retry on failure) and propagates the mask bidirectionally through the clip. Masklets whose largest single-frame area is below 0.3% of the image are dropped, and masklets of different concepts with temporal IoU above 0.5 are merged. Clips in which no dynamic object obtains a mask are discarded (9,207, of which 1,292 receive no mask at all).
- *Track attribution.* Every released 3D track is projected into the frames and attributed to the instance whose mask (dilated by 4 px) contains at least 50% of its visible projections and beats the runner-up by a margin of 0.15; only tracks visible in at least 3 frames are eligible, and at most 800 tracks per instance are kept by farthest-point sampling. A dynamic instance is kept only if it owns at least 8 tracks (4 for a static instance), one of which falls on its undilated mask. Clips whose dynamic objects all fail this test are discarded (5,725).
- *Motion check.* A one-way gate demotes a dynamic instance to static when the median world-frame speed of its tracks is below 0.05 m/s and a static 3D point explains its tracks up to a median image residual of 1 px; it never promotes a static instance. Clips whose last dynamic object is demoted are discarded (2,444).
- *Failures.* 11 clips failed (7 rejected by the VLM API's content filter, 4 pipeline give-ups).

Outcome and splits. The remaining 42,152 clips come from 4,079 source videos; one clip with a corrupt reconstruction file is excluded from QA generation, leaving 42,151 QA scenes with, on average, 9.9 objects and 44.8 keypoints per scene (20.7 on moving objects). Since a source video contributes several clips (median 5, up to 160; 98.4% of the clips share their video with at least one other clip), the QA splits are made by hashing source-video id. The separate stage-1 caption training selection excludes QA evaluation scenes; its source-video-level relation to the QA splits is not documented here. The pipeline's numerical gates check consistency among attributed tracks, projected motion, and serialized answers. They do not provide an independent measurement of metric reconstruction fidelity or a human audit of object identity and mask correctness. Errors in either upstream stage can therefore survive a token-side answer check.

B Trajectory Representation Pipeline Details

Tracking module. Two interchangeable implementations produce the same output, namely per-object dense 3D tracks with per-frame visibility, together with camera poses. (i) For ordinary monocular video we use Track4World: a dense grid of query points is seeded inside every instance mask (about 200 seeds per object, at most 2,000 tracks), tracked in a few shared passes with the first camera as world frame, and re-tracked once more at the reference frame t^* after the mesh is available.

Table 5: Tracker bake-off (five trackers, three datasets). EPE-2D is measured in pixels and EPE-3D in each dataset's 3D coordinate units (lower is better); δ_8 is the fraction of points within 8 px, OA is occlusion accuracy, and on-mask is the fraction of tracked points remaining on the object mask (higher is better). Bold marks the best available value for each dataset and metric.

Tracker	Dataset	EPE-2D↓	δ_8 ↑	OA↑	EPE-3D↓	on-mask↑
Track4World	DAVIS	–	–	–	–	**0.9155**
	MOVi	**1.722**	**0.9257**	**0.8778**	0.3532	–
	PointOdyssey	**10.410**	**0.5597**	**0.8296**	0.3087	–
TAPIP3D (Zhang et al., 2025a)	DAVIS	–	–	–	–	0.9032
	MOVi	1.836	0.9124	0.8599	0.4259	–
	PointOdyssey	23.462	0.3647	0.7182	0.6936	–
SpaTrackerV2 (Xiao et al., 2025)	DAVIS	–	–	–	–	0.8970
	MOVi	1.918	0.9043	0.8576	0.4908	–
	PointOdyssey	16.937	0.4130	0.7795	**0.3029**	–
4RC (Luo et al., 2026)	DAVIS	–	–	–	–	0.7417
	MOVi	2.031	0.9066	0.8541	**0.2653**	–
	PointOdyssey	100.364	0.3320	0.3931	0.3546	–
D4RT (Zhang et al., 2025b)	DAVIS	–	–	–	–	0.5898
	MOVi	2.852	0.8583	0.8480	0.3695	–
	PointOdyssey	234.402	0.0031	0.7304	1.6874	–

(ii) For datasets that release 3D tracks we use the released tracks, attributed to instances as in Appendix A. All Stereo4D scenes in this paper use (ii); none of our training trajectories comes from a learned tracker.

Tracker selection. We selected Track4World after a bake-off of five 3D trackers on three datasets (Table 5): DAVIS (no 3D ground truth; only the fraction of tracked points that stay on the object mask is measured), MOVi and PointOdyssey (3D ground truth). Evaluation points were unified across trackers and queries were released at the same frames. Track4World has the highest DAVIS on-mask score and the best EPE-2D, δ_8, and OA values on MOVi and PointOdyssey in this table. Other methods obtain the lowest EPE-3D on those two datasets, so the choice reflects the tracking and visibility metrics used by the pipeline rather than a uniform win across metrics.

Object geometry and keypoints. Objects are routed by type. *Non-rigid dynamic* objects are reconstructed with SAM 3D Objects (Fast-SAM3D implementation (Feng et al., 2026)) from the best of at most three candidate reference frames (chosen by mask agreement), segmented into semantic parts with PartSAM (fragments are merged into neighbouring parts), and sampled on the mesh surface in two passes: the first draws farthest-point, curvature-weighted vertices per part until a coverage of 0.10 part diagonals is reached (at most 64 per part); the second keeps one medoid keypoint per semantic part among candidates that are visible in at least half of the frames, and discards points that are never observed. *Rigid dynamic* objects skip the mesh when the VLM calls them rigid, a conservative rigidity test on their raw tracks agrees, and they are present in a single contiguous segment; they receive 4 farthest-point keypoints and a per-frame pose $(\boldsymbol{R}_t, \boldsymbol{t}_t) \in SE(3)$ obtained by Kabsch alignment over all tracked keypoints (the pose is marked undefined when fewer than $\max(3, 0.1T)$ frames are solved or the reconstruction residual exceeds one bounding-box extent). *Static* objects keep 4 representative points (farthest-point sampling of per-track median positions) and no mesh.

Keypoint–pixel–track linking. At the reference frame t^* the track coordinate frame is aligned to the mesh by a Umeyama similarity; every keypoint is projected into the image and linked to the dense track that passes through that pixel; keypoints occluded at t^* are placed by the SE(3) motion of their part before linking. Each keypoint thereby inherits one metric 3D track over the whole clip together with its per-frame visibility. The pipeline output for static objects consists of their representative points only.

C TOKENIZATION AND PROJECTOR DETAILS

Throughout, T is the number of frames of the scene and d the diagonal of the trajectory AABB enclosing an object's visible normalized points across time; d includes motion extent and is not necessarily the object's physical size.

Scene normalization. All visible keypoint positions of a scene are shifted by their centroid $\boldsymbol{\mu}$ and scaled isotropically by the maximum radius s, $\tilde{\boldsymbol{p}} = (\boldsymbol{p} - \boldsymbol{\mu})/s$, so that the scene lies in the unit sphere; $(\boldsymbol{\mu}, s)$ are emitted in the scene header (for the clip in Fig. 3, $\boldsymbol{\mu} = (1.46, 2.12, 10.06)$ m and $s = 10.23$ m). A low-motion check re-labels a dynamic object as static when its world-frame motion is at most 5 cm and its reprojection shows no image motion (residual below 1 px).

Event detection. Inside a visible run we detect two kinds of events on the inter-frame displacements $\boldsymbol{\Delta}_t = \tilde{\boldsymbol{p}}_{t+1} - \tilde{\boldsymbol{p}}_t$: a *reversal* at frame t if $\langle \boldsymbol{\Delta}_{t-1}, \boldsymbol{\Delta}_t \rangle < 0$ and $\|\boldsymbol{\Delta}_{t-1}\|, \|\boldsymbol{\Delta}_t\| > \delta$ with the noise floor $\delta = \max\left(0.02\,d,\ 0.1\,\mathrm{median}_{t:\|\boldsymbol{\Delta}_t\|>0}\|\boldsymbol{\Delta}_t\|\right)$, and an *impact* at t if the second difference $a_t = \|\tilde{\boldsymbol{p}}_{t+1} - 2\tilde{\boldsymbol{p}}_t + \tilde{\boldsymbol{p}}_{t-1}\|$ is a local maximum exceeding $\max(\mu_a + 2\sigma_a,\ 0.05\,d)$. The run is split at every event frame; adjacent events are not merged.

Fitting procedure. We use FITPACK's smoothing spline (`splprep`) and bisect its smoothing factor (30 iterations) to the largest value that satisfies the target bound in Eq. 1; adaptive knots concentrate nodes on motion that is harder to fit. A piece may use at most $n = 12$ coefficients; if the bound cannot be met within this budget the best fit is kept and counted as a bound failure in preprocessing (3,088 of the 1,729,834 pieces in our data, 0.18%). No dedicated fit-status field is serialized to the model. Pieces with fewer than five frames use the lowest degree that meets the bound (a line with 2 coefficients, else a quadratic with 3, else interpolation of all points), and a piece with a single frame is one point. During preprocessing, the control points can be recovered numerically from the Greville nodes and the fitted knot vector to 10^{-12}. The LLM receives the node records, not that knot vector; Gate 0 therefore recomputes targets by interpolating the serialized nodes rather than relying on exact spline reconstruction. Knot-type flags are assigned as follows: *endpoint* for the first and last node of a run, *reversal* or *impact* for the two event-side boundary nodes, and *smooth* otherwise. Since fits are smoothed separately, event-side node positions are not guaranteed to coincide exactly.

What the fit bound measures. Let $\hat{\boldsymbol{p}}(t)$ denote the piecewise-linear path decoded from the serialized nodes after the record cap. Eq. 1 controls the fitted spline at observed frames, whereas the model and Gate 0 use $\hat{\boldsymbol{p}}(t)$. Because the knot vector is omitted, that bound does not imply an equal bound for the decoded path. Gate 0 only verifies that the decoded answer remains in the same option cell (or gives the same categorical answer), which is weaker than bounding trajectory error. Each of the three principal numeric questions is determined by two decoded endpoint positions and their timestamps; consequently, these questions do not by themselves test whether interior event nodes, part labels, or adaptive sampling are needed. At a split event, the two fitted sides can produce distinct samples at the same time; the available manuscript does not specify the implementation's duplicate-time interpolation rule or report decoded-path error statistics. These details require a code-level audit before asserting a stronger fidelity guarantee.

Rigid objects and the camera. A rigid object is described by its keypoints' reconstruction $\boldsymbol{p}_j(t) = \boldsymbol{R}_t\,\tilde{\boldsymbol{r}}_j + \boldsymbol{c}(t)$, where $\tilde{\boldsymbol{r}}_j$ are its reference points relative to their centroid and $\boldsymbol{c}(t)$ is the normalized centroid trajectory. Per run we fit the translation $\boldsymbol{c}(u)$ and the rotation $\boldsymbol{\phi}(u) = \log(\boldsymbol{R}_{f_0}^{\top}\boldsymbol{R}(u))$ (rotation vector relative to the first frame of the run, unwrapped modulo 2π) jointly as one 6-D curve $[\boldsymbol{c}(u),\ \rho\,\boldsymbol{\phi}(u)]$ with shared knots, where $\rho = \max_j \|\tilde{\boldsymbol{r}}_j\|$ converts radians into length, and the error bound of Eq. 1 is evaluated on the reconstructed keypoints, $\max_{i,j} \|\boldsymbol{R}_{f_0}\exp(\boldsymbol{\phi}(u_i))\,\tilde{\boldsymbol{r}}_j + \boldsymbol{c}(u_i) - \tilde{\boldsymbol{p}}_j(f_i)\| \le \varepsilon$. Rotation nodes are serialized as absolute rotation vectors. The camera is encoded as a rigid pseudo-object appended last, whose translation stream is the optical centre (error bound $\min(\varepsilon,\ 0.10\ \mathrm{m}/s_{\mathrm{norm}})$ in normalized coordinates, at most 48 coefficients). Here $s_{\mathrm{norm}} = s$ is the scene normalization radius in metres, not elapsed seconds.

Token record. Every token is a record with seven discrete and three continuous fields (Table 6); fields that do not apply to a role are masked by a presence flag. Besides the node tokens there are

Table 6: Fields of a trajectory token. Discrete fields are embedded and continuous fields are projected by the adapter in Sec. 3.3. Declared cardinalities include reserved and padding values beyond the semantic values listed; this table defines fields rather than performance metrics.

Field	Type	Values
role	discrete (12)	scene header, normalization centre, normalization scale, object header, static point, rigid reference point, rotation node, translation node, keypoint node, occlusion gap, visibility summary
knot type	discrete (5)	smooth, reversal, impact, endpoint
kind	discrete (4)	static, rigid, non-rigid
object type	discrete (4)	rigid, non-rigid (VLM attribute)
instance	discrete (65)	per-scene object index
part	discrete (17)	semantic part index within the object
keypoint	discrete (129)	keypoint index within the part
xyz	continuous (3)	normalized position, rotation vector, reference point, or gap interval
u	continuous (1)	normalized time in the clip, $u = f/(T-1)$
t	continuous (1)	absolute time in seconds from the frame timestamps

structural tokens: three scene-header tokens (frame count and clip duration; normalization centre; normalization scale), one object-header token per object (kind, VLM type, instance index, number of parts, and for rigid objects the SE(3) reconstruction residual), up to four point tokens per static object, the reference-point tokens of rigid objects, and, per non-rigid keypoint, one visibility-summary token and one token per occlusion gap.

Serialization and budget. The tokens of a scene are serialized as one sequence: the scene header, then every object in order, and the camera last. Each object is introduced by its category label as ordinary text, followed by its object-header token; a non-rigid object nests its parts as `[part 1]`, `[part 2]`, ..., each followed by the tokens of that part's keypoints (visibility summary, gaps, nodes); a rigid object lists its reference points, rotation nodes and translation nodes; a static object its points. Several instances of the same category share the same text label, and their identity is carried by the instance field. Over the 42,151 scenes a clip has a median of 165 trajectory tokens (mean 231, 90th percentile 507, maximum 2,227); for comparison, the 8 video frames occupy 1,152 visual tokens. At training time a sample is capped at 512 trajectory tokens by keeping all structural tokens and all reversal/impact nodes and thinning the remaining nodes uniformly.

Projector dimensions and injection. In Eq. 2 every embedding table $\boldsymbol{E}_k$ and every linear map $\boldsymbol{W}_{\boldsymbol{x}}, \boldsymbol{W}_u, \boldsymbol{W}_t$ has 456 output dimensions, so the concatenation has $10 \times 456 = 4{,}560$ dimensions; $\boldsymbol{W}_{\mathrm{f}} \in \mathbb{R}^{4096 \times 4560}$ maps it to the LLM dimension $D = 4096$, and the MLP is $4096 \to 8192 \to 4096$ with GELU. The output replaces the embedding of the corresponding `<|traj_pad|>` placeholder by a masked scatter into the LLM's input embeddings, before the visual tokens are inserted. The final LayerNorm sets the norm of every injected vector to $\sqrt{D} = 64$.

D Question Generation Details

Reference quantities and observability guards. The object position at frame t is the per-frame median of its visible keypoints (for rigid objects the pose applied to the reference points), i.e. the same quantity the tokens encode; the camera position is the optical centre from the released poses. Displacement is the Euclidean distance between the positions at frames i and j; the benchmark's "speed" target divides this endpoint displacement by elapsed time (intervals shorter than 0.5 s are excluded). It is the magnitude of average velocity, not path length divided by time. Motion duration is the longest contiguous span in which the per-frame speed exceeds 0.05 m/s and the object moves by at least 10 px in a frozen-camera projection. Windows are random pairs of visible frames with $j - i \geq 3$; the two intervals of the comparison families are adjacent, with equal length for interval_disp and a length ratio of at least 1.5 for interval_speed, and their displacements (speeds) must differ

Table 7: Question families and complete numeric grid parameters for TrajQA-B (descriptive table; no performance metric). Numeric families are 4-way with options taken from a fixed logarithmic grid $v_k = v_0\, r^k$ ($r = 1.35$); the remaining families are 2-way. The internal name speed_grid denotes endpoint displacement rate. Numeric questions specify frame indices without stating the elapsed time; the duration family instead specifies a time threshold.

Family	Asks for	Options	Needs time?
disp_grid	world-frame displacement of an object in $[i, j]$ (m)	grid, $v_0 = 0.5$, 12 cells	no
speed_grid	endpoint displacement rate of an object in $[i, j]$ (m/s)	grid, $v_0 = 0.1$, 14 cells	yes
cam_disp_grid	displacement of the camera in $[i, j]$ (m)	grid, $v_0 = 0.5$, 13 cells	no
interval_disp	which of two adjacent intervals covers the longer distance	2-way	no
interval_speed	which interval has greater endpoint displacement rate	2-way	yes
motion_source	whether the screen motion is due to the object or the camera	2-way	no
motion_duration	whether the object keeps moving for more than X seconds	yes/no	yes

by a factor of at least 1.4. A question is generated only if the asked quantity is observable: the displacement must exceed $\max(0.5\text{ m},\, 3\,\eta(z))$, where $\eta(z)$ is a depth-dependent noise level of the stereo reconstruction (0.05 m below 5 m, 0.25 m below 10 m, 0.53 m below 20 m, 0.90 m below 40 m; objects beyond 40 m are excluded); the object must move by at least 10 px under a frozen camera; the keypoint-based displacement must agree with the displacement of the object's dense tracks within $\max(0.3\text{ m}, 0.3\,d)$; radial motion must be corroborated by the change of apparent size; and speeds above 6 m/s for humans (30 m/s otherwise) are rejected. For motion_source, the "camera" side uses only static objects of movable categories (humans, vehicles, animals) under a camera that moves by at least 0.5 m or 5°, and the "object" side only dynamic objects under a camera that moves by at most 0.10 m and 1°; both require at least 40 px of screen motion. An object is referred to by its label when it is unique in the scene, or as "the left/right *label*" when exactly two instances are separated horizontally by at least 15% of the image width in every frame the question involves.

Options and balancing. Numeric options are the geometric centres of four *adjacent* cells of a fixed global logarithmic grid, $v_0 r^{w+k+0.5}$ for $k = 0, \ldots, 3$, chosen so that the cell of the true value is one of the four; the four letters are randomly permuted, and true values that fall within 6% of a cell edge are discarded. The grid is global rather than adapted to a scene. The true cell's rank within the window is balanced by a linear program (per stratum of object class $\times$ frame-distance bin) so that $P(\text{rank} \mid \text{window})$ is uniform. This limits common rank and class-median shortcuts, though the question-only controls show that residual bias is possible. The 2-way families are balanced in cells of question-visible context: interval_disp per object class over both answers and over whether the longer-displacement interval is the first one; interval_speed per class, duration-ratio bin and split-point position, keeping in equal numbers only the cases where the faster interval is the longer one with the larger displacement and where it is the shorter one with the smaller displacement, so that neither "the longer interval" nor "the larger displacement" predicts the answer; motion_source per class, label, clip length, frame distance, frame rate and screen position over both answers. Within every cell the answer letter alternates strictly. Per object and family, windows with an overlap ratio above 0.7 are not repeated; at most 80 questions per family and 300 questions in total are drawn from any source video.

Admission gates. *Gate 0 (token-side recomputation, hard).* For every candidate we rebuild the object (or camera) trajectory from exactly what the model sees—the serialized node positions and times after the 512-token cap, linearly interpolated between nodes, denormalized with the scene centre and scale—and recompute the answer; a question is dropped unless the recomputed value falls in the same grid cell (numeric families) or yields the same answer (2-way and yes/no families). *Gate 1 (text-shortcut audit).* On the finished splits we fit, on the training questions, a ridge regression

from bag-of-words, frame indices and window position to the nearest option, a logistic regression predicting the rank of the answer, and fixed rules (always the k-th smallest option, family or class medians; for 2-way families the longer or first interval, always camera, always yes), and require every predictor to score within 3 points of chance on the test questions; on the final splits the numeric families score 0.252–0.262 (chance 0.25). *Gates 2–3 (post-training).* After training we evaluate every model with the question only (no frames, no tokens) and, for token-reading models, with the tokens of a different scene substituted for the true ones (format, count and timestamps unchanged); these controls are reported in Sec. 4. Finally, a validation script checks every build: zero overlap of source videos, question ids and (video, object, window, family) tuples between splits; Gate 0 consistency of every retained question; rank-per-window and answer-side deviations below 10^{-9}; and no window pair of the same object and family with overlap above 0.8.

Splits and counts. QA splits are formed on the hash $u \in [0, 1)$ of the source-video id: $u < 0.2$ is the test split, $[0.2, 0.25)$ the validation split used for model selection, and $[0.25, 0.30)$ a held-out QA split; the training split takes the remaining videos. This separation applies to QA fine-tuning, not necessarily to the earlier caption-based projector warm-up described in Appendix E. From 42,151 scenes the generator produced 333,661 candidates, of which 65,360 questions survived the gates and balancing (Table 4); the training split was additionally resampled so that its family proportions match the test split (camera 38%, object displacement 29%, speed 27%). Every question is paired with a structured target: the evidence lists world-frame positions at the two frames in metres, the resulting displacement, and the elapsed time and speed where relevant, followed by the option letter.

E Training Details

Model and optimization. The backbone is Qwen3-VL-8B-Instruct. Its vision encoder and all base weights stay frozen; we train the Trajectory Token Projector (85.93M parameters) and LoRA adapters on the language model (rank 128, $\alpha = 256$, dropout 0.05, on the query, key, value, output, gate, up and down projections; 349.18M parameters). The projector embeddings are added to the input through a wrapper of the embedding layer, so gradients reach the projector even when the LLM is frozen. A sample consists of 8 frames sampled uniformly over the whole clip (longest side 384 px, 144 visual tokens per frame), the serialized trajectory block with its text labels (at most 512 trajectory tokens), and the instruction or question; there is no system prompt and no per-frame timestamp. For questions the prompt ends with an instruction to list the determining trajectory facts inside `<evidence>` tags before giving the letter inside `<answer>` tags, and the loss is the standard next-token cross-entropy on the whole assistant text (evidence and answer); prompt, image and trajectory positions are not supervised. All stages use AdamW ($\beta = (0.9, 0.95)$, no weight decay), gradient clipping at 1.0, a global batch of 16 samples, bf16 weights for the LLM, and seed 0.

Caption generator (stage 1). The captions are produced from the 4D tracks by a deterministic, camera-independent generator: one *scene inventory* item per scene ("Moving in the world frame: … Staying in place: …") and one *object motion* item per uniquely labelled moving object (net displacement in metres, mean speed in m/s, acceleration or deceleration when the mean speeds of the first and last thirds of the clip differ by a factor of at least 1.5, slowing to a stop, the frame of a direction reversal, and approaching, moving away from or moving faster than another object), 65,556 captions in total (42,030 inventories, 23,526 object descriptions), split by source video into 64,523 training and 1,033 validation captions. Stage 1 receives frames and trajectory tokens as input and generates the caption; only the projector is trained (no LoRA). The 42,030 caption scenes belong to a separately selected caption corpus, not a subset of the 42,151 QA scenes. Its 64,523 training captions exclude the 3,103 QA evaluation scenes. The exact source-video-level intersection, the selection rule relating the two corpora, and any overlap with caption validation are not documented in the available manuscript; these require a scene-ID manifest audit.

Schedules. *Stage 1.* The projector is kept in fp32 and trained for 26,000 steps (about 6.4 epochs): 500 warm-up steps, a constant learning rate of 2×10^{-4} until step 24,000, then a linear decay to zero over 2,000 steps. *Stage 2.* Starting from the stage-1 projector, LoRA adapters are added and trained together with the projector on the 56,120 training questions with the frames withheld. The learning rate is 1×10^{-4} for the adapters and 2×10^{-4} for the projector, with 180 warm-up steps, a constant

Table 8: Training hyperparameters (descriptive, not performance metrics). All stages use a global batch of 16, AdamW without weight decay and seed 0; peak LR gives the LoRA / projector learning rates.

Stage	Inputs	Trainable	Peak LR	Schedule	Steps
1 warm-up	frames + tokens	projector (fp32)	– / 2×10^{-4}	500 warm-up; constant; linear over steps 24,000–26,000	26,000
2 token-only	tokens	LoRA + projector	10^{-4} / 2×10^{-4}	180 warm-up; constant; linear over steps 7,000–8,400	8,400
3 joint	frames (p=0.8) + tokens	LoRA + projector	10^{-4} / 2×10^{-4}	105 warm-up (3%); cosine to zero over one epoch	3,507

phase until step 7,000 and a linear decay to zero over 1,400 steps (8,400 steps, about 2.4 epochs); the length of the constant phase and of the decay were chosen on the validation split. *Stage 3.* Adapters and projector continue from stage 2 with the frames restored; each training sample independently loses its 8 frames with probability 0.2 (trajectory tokens are never dropped). The schedule is one epoch (3,507 steps) with 105 warm-up steps (3%) and cosine decay to zero at the same peak learning rates; the checkpoint at the end of the epoch is used without any selection. Table 8 summarizes the three stages. At inference we decode greedily and read the letter inside the `<answer>` tags.

F Experimental Setup

Evaluation set and metrics. We evaluate on the held-out and test QA splits of TrajQA-B (Sec. 4; $3{,}430{+}4{,}492 = 7{,}922$ questions). Their scenes were excluded from stage-1 training and their source videos from stages 2–3 training and checkpoint selection. The stage-1 source-video and validation-set relation to QA evaluation still needs an ID-level audit. The main metric is pooled accuracy over the three numeric families (camera displacement, object displacement, endpoint displacement rate; $n = 7{,}500$; 4-way, uniform-guessing accuracy 25.0), also reported per split and per family; the mean accuracy over the four small 2-way and yes/no families ($n = 76$–146 each) is reported but not interpreted. All trained-model results come from a single seed. Because questions are clustered by source video, question-level binomial standard errors are not a valid measure of across-video uncertainty; source-video bootstrap intervals were not available for this version.

Baselines and protocol. The evaluated families include Qwen3-VL (Bai et al., 2025), InternVideo2.5 (Wang et al., 2025b), InternVL3.5 (Wang et al., 2025a), VG-LLM (Zheng et al., 2025b), VLM-3R (Fan et al., 2026), and MLLM-4D (Yin et al., 2026). Exact proprietary API version strings, query dates, and several released checkpoint identifiers are not specified here, limiting exact replication of the baseline scores. Baselines are evaluated zero-shot with frames sampled uniformly over the whole clip, greedy decoding and the first option letter in the output. The open-source 7–8B models use a controlled protocol with our visual budget (8 frames at 384 px, no timestamps); the larger and the specialised models use their official video configuration (all frames at 512 px or the official number of segments, official visual pathway, timestamps given). The two protocols agree: Qwen3-VL-32B scores 28.1 under the controlled and 27.8 under its official configuration. The two proprietary models are queried through their APIs with 8 frames at 512 px and timestamps on a random subset of 1,000 questions (500 per split; 998 answered by GPT-6 Astra). The available result record does not specify whether its two unanswered questions were counted as incorrect or excluded from the denominator. MLLM-4D-SFT is run with its official usage (32 frames through the official video processor with timestamps, and the official answer parser). Stages 2–3 of our model are fine-tuned on the TrajQA-B training split, whose source videos are disjoint from both QA evaluation splits; stage-1 training uses a separately selected caption corpus without QA evaluation scenes as noted in Appendix E. The comparison with zero-shot models is protocol-mismatched. The frames-only arm uses the same QA question split, but its full optimization schedule and checkpoint-selection rule are not specified here; it should therefore be treated as a task-adapted visual reference,

Table 9: Supplementary comparison columns. Checkpoint size (GB) is descriptive; non-numeric question accuracy (%) is higher-is-better (↑). Bold marks the maximum in this column. These small question families are reported for completeness and are not used for the main comparison. Numeric results for every model appear in Table 1.

Model	Checkpoint (GB)	Non-numeric acc.↑
GPT-6 Astra	–	73.2
Claude Fable 5.1	–	69.6
Qwen2.5-VL-7B	15.4	62.3
Qwen3-VL-8B	16.3	64.2
LLaVA-NeXT-Video-7B	13.2	51.9
InternVideo2.5-8B	15.0	57.6
InternVL2.5-8B	15.0	63.0
InternVL3.5-8B	15.9	62.1
Qwen2.5-VL-32B	63.6	64.7
Qwen3-VL-32B	62.1	66.4
InternVL2.5-38B	71.5	59.2
InternVL3.5-38B	71.5	60.4
VG-LLM	17.2	57.8
VLM-3R	17.4	55.5
MLLM-4D-SFT	16.3	68.0
Ours (frames + trajectory tokens)	17.9	**90.3**

Table 10: Supplementary non-numeric ablation average (%, higher-is-better ↑). Bold marks the maximum. These small question families are reported for completeness; the main input comparison is Table 2. A token-only non-numeric score was not in the available results.

Inputs at inference	Non-numeric acc.↑
Question only	49.5
Frames only (separate FT)	80.8
Frames + donor tokens	69.0
Trajectory tokens only	–
Frames + trajectory tokens (full)	**90.3**

not a fully training-matched ablation. For the donor intervention, the recorded protocol preserves token format, count, and timestamps while replacing trajectory content with another scene's. Donor sampling and object/part mapping are not specified here, so this intervention is a coarse sensitivity test rather than a fully reproducible matched control.

MLLM-4D-SFT is not zero-shot. Its SFT set is built from Stereo4D like TrajQA-B: 2,635 of the 3,103 evaluation clips are in it. Its 25.0% pooled score is consistent with an option-position shortcut: in its 856k numeric training questions the correct option is the second- or third-smallest value 99.4% of the time; on TrajQA-B it picks the third-smallest option in 92.6% of numeric questions, whereas TrajQA-B balances the answer positions (25% each). Among the overlap categories available in the audit, it scores 24.9 on 6,332 numeric questions from clips in its SFT data and 26.1 on 904 questions from videos absent from that data. These categories cover 7,236 of the 7,500 numeric questions; the remaining 264 were not classified in the available breakdown.

Evidence metrics. From the generated evidence we parse the stated positions and displacement or benchmark "speed"; we report median L2 position error (objects and camera separately) and the fraction whose stated quantity is within a multiplicative factor of 1.2 of the reference, i.e. $|\log(\text{stated}/\text{reference})| < \log 1.2$. Evidence is directly supervised during training, so this diagnostic is not an independent proof of the model's internal reasoning process.

Full result tables. Tables 9 and 10 report checkpoint size and the mean accuracy over the four small non-numeric families; all numeric-family results are in Tables 1 and 2.

Table 11: Endpoint-rate errors from Fig. 1 and three examples below. Errors use rounded displayed predictions (m/s, ↓); bold is best within each row. These are illustrative cases.

Scene	Reference (m/s)	Frames-only error↓	Trajectory-input error↓
Bicycle	4.13	2.84	**0.18**
Stroller	0.64	0.31	**0.16**
Multiple people	0.26	0.06	**0.00**
Position swap	0.43	0.14	**0.03**

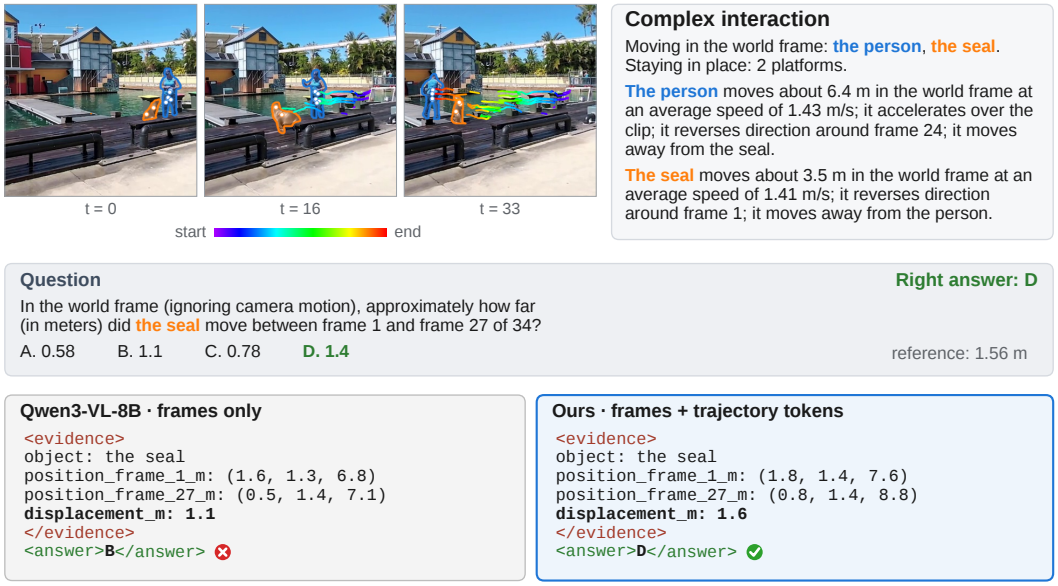


Figure 4: **Complex interaction.** *Top left:* three frames with estimated keypoint trajectories (colour encodes time). *Top right:* the rule-generated motion caption. *Bottom:* seal displacement between frames 1 and 27 (reconstruction-derived reference 1.56 m): frames only 1.1 m (wrong), frames and trajectory tokens 1.6 m (correct).

Secondary diagnostics. *Trajectory tokens only.* The final model evaluated without frames scores 82.2 on the numeric families (81.5 on held-out, 82.8 on test; camera displacement / object displacement / endpoint rate 89.4 / 77.7 / 75.5). Its full-input score is the same to one decimal place; the separately trained frames-only arm scores 33.9 points lower than the full-input model. The latter difference is descriptive because training and visual inputs are not matched. *Floors.* On held-out the full model is $+54.2$ points above its own question-only floor (81.7 versus 27.5) and $+51.1$ above the donor control (30.6); on the test split the corresponding numbers are 82.5 versus 26.2 ($+56.3$) and 28.8 ($+53.7$). The frames-only arm's own question-only floor is 25.7 (24.3 on held-out, 26.9 on test). *Evidence metrics on the test split.* Median position error (objects / camera) and within-20% rates (camera displacement / object displacement / speed): full model 0.32 / 0.10 m, 91.8 / 83.3 / 84.1%; trajectory tokens only 0.32 / 0.10 m, 92.0 / 83.6 / 83.8%; frames + donor tokens 3.05 / 1.09 m, 15.1 / 15.6 / 15.9%; question only 3.16 / 0.64 m, 23.0 / 17.1 / 14.5%.

G Qualitative Results

Figures 4–7 show four clips whose source videos are outside the stages 2–3 QA training split and whose scene IDs were excluded from stage-1 caption training. Each shows the moving objects' keypoint trajectories, a rule-generated motion caption, and one TrajQA-B question with outputs from the separately trained frames-only and trajectory-input models.



Complex video

Moving in the world frame: 16 persons, the stroller. Staying in place: 5 signs, 2 persons.

The stroller moves about 3.8 m in the world frame at an average speed of 1.04 m/s; it reverses direction around frame 4.

Question **Right answer: A**

In the world frame (ignoring camera motion), what was the average speed (in meters per second) of the stroller between frame 5 and frame 16 of 34?

A. 0.70 B. 1.7 C. 0.95 D. 1.3 reference: 0.64 m/s

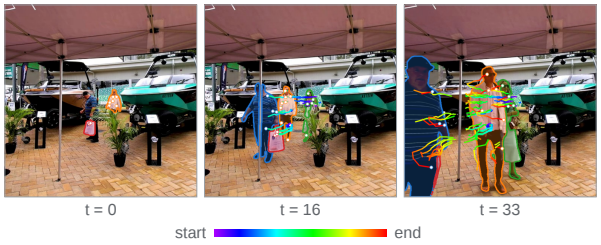
Qwen3-VL-8B · frames only

```
<evidence>
object: the stroller
position_frame_5_m: (-0.7, 1.3, 8.3)
position_frame_16_m: (-0.6, 1.1, 6.2)
displacement_m: 2.1
time_s: 2.20
speed_mps: 0.95
</evidence>
<answer>C</answer>
```

Ours · frames + trajectory tokens

```
<evidence>
object: the stroller
position_frame_5_m: (-0.6, 1.5, 8.7)
position_frame_16_m: (-0.6, 1.3, 6.9)
displacement_m: 1.8
time_s: 2.20
speed_mps: 0.80
</evidence>
<answer>A</answer>
```

Figure 5: **Complex video.** Same layout as Fig. 4. The stroller's endpoint displacement rate between frames 5 and 16 (reference 0.64 m/s): frames only 0.95 m/s (wrong), frames and trajectory tokens 0.80 m/s (correct).



Multiple people

Moving in the world frame: **the man in striped shirt**, **the man in white shirt**, **the woman in white dress**, **the shopping bag**. Staying in place: the tent canopy, 2 potted plants, the boat on trailer (green), the boat on trailer (brown), 2 sign stands.

The man in striped shirt moves about 1.5 m in the world frame at an average speed of 0.45 m/s; it reverses direction around frame 4; it moves away from the woman in white dress; it moves faster than the woman in white dress.

The man in white shirt moves about 1.8 m in the world frame at an average speed of 0.42 m/s; it reverses direction around frame 12; it moves faster than the woman in white dress.

The woman in white dress moves about 0.9 m in the world frame at an average speed of 0.26 m/s; it accelerates over the clip; it reverses direction around frame 16; it moves away from the man in striped shirt.

Question — **Right answer: A**

In the world frame (ignoring camera motion), what was the average speed (in meters per second) of **the man in white shirt** between frame 4 and frame 18 of 34?

A. 0.29 B. 0.39 C. 0.52 D. 0.21

reference: 0.26 m/s

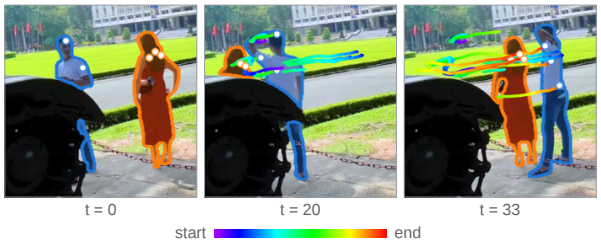
Qwen3-VL-8B · frames only

```
<evidence>
object: the man in white shirt
position_frame_4_m: (0.1, 0.1, 4.1)
position_frame_18_m: (-0.4, 0.1, 3.3)
displacement_m: 0.9
time_s: 2.80
speed_mps: 0.32
</evidence>
<answer>A</answer>
```

Ours · frames + trajectory tokens

```
<evidence>
object: the man in white shirt
position_frame_4_m: (0.1, -0.2, 3.5)
position_frame_18_m: (-0.4, 0.1, 3.0)
displacement_m: 0.7
time_s: 2.80
speed_mps: 0.26
</evidence>
<answer>A</answer>
```

Figure 6: **Multiple people.** Same layout as Fig. 4. The man's endpoint displacement rate between frames 4 and 18 (reference 0.26 m/s): both models answer correctly, frames only with 0.32 m/s and frames and trajectory tokens with 0.26 m/s.



Objects swapping positions

Moving in the world frame: **the man**, **the woman**. Staying in place: the tank, the tree, the fountain, 2 chain barriers, the lamp post.

The man moves about 0.5 m in the world frame at an average speed of 0.23 m/s; it reverses direction around frame 15; it approaches the woman.

The woman moves about 0.3 m in the world frame at an average speed of 0.26 m/s; it accelerates over the clip; it reverses direction around frame 21; it approaches the man.

Question — **Right answer: B**

In the world frame (ignoring camera motion), what was the average speed (in meters per second) of **the woman** between frame 25 and frame 32 of 34?

A. 0.21 **B. 0.39** C. 0.52 D. 0.29

reference: 0.43 m/s

Qwen3-VL-8B · frames only

```
<evidence>
object: the woman
position_frame_25_m: (-0.8, 0.4, 3.0)
position_frame_32_m: (-0.4, 0.5, 3.1)
displacement_m: 0.4
time_s: 1.40
speed_mps: 0.29
</evidence>
<answer>D</answer>
```

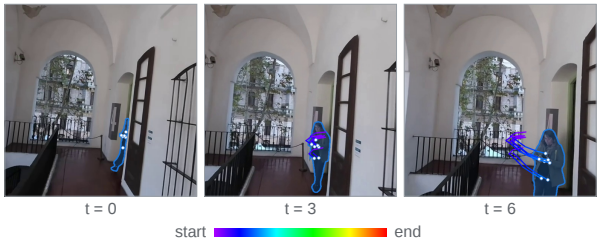
Ours · frames + trajectory tokens

```
<evidence>
object: the woman
position_frame_25_m: (-0.9, 0.2, 2.4)
position_frame_32_m: (-0.4, 0.3, 2.7)
displacement_m: 0.6
time_s: 1.40
speed_mps: 0.40
</evidence>
<answer>B</answer>
```

Figure 7: **Objects swapping positions.** Same layout as Fig. 4. The woman's endpoint displacement rate between frames 25 and 32 (reference 0.43 m/s): frames only 0.29 m/s (wrong), frames and trajectory tokens 0.40 m/s (correct).

G.1 ADDITIONAL QUALITATIVE EXAMPLES

Five further selected clips are shown in Figures 8–12. Their embedded "ground truth" labels mean reconstruction-derived references, and any "average speed" question is scored as endpoint displacement divided by elapsed time. The cases illustrate answer and evidence differences; they are not a representative sample or component ablations.



Approaching person

Moving in the world frame: **the person**. Staying in place: the railing, the tree, the sign, 2 doors.

The person moves about 1.1 m in the world frame at an average speed of 0.68 m/s.

Question — **Right answer: A**

In the world frame (ignoring camera motion), what was the average speed (in meters per second) of **the person** between frame 0 and frame 6 of 26?

A. 0.52 B. 0.95 C. 0.70 D. 1.3

ground truth: 0.59 m/s

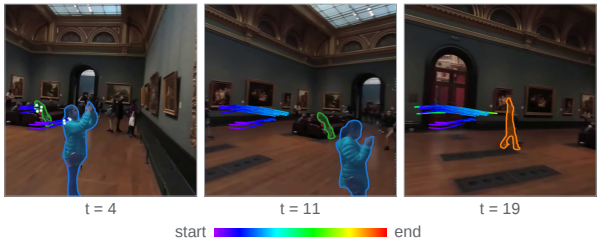
Qwen3-VL-8B · frames only

```
<evidence>
object: the person
position_frame_0_m: (0.8, 1.6, 6.5)
position_frame_6_m: (0.4, 1.2, 4.8)
displacement_m: 1.8
time_s: 1.20
speed_mps: 1.53
</evidence>
<answer>D</answer>
```

Ours · frames + trajectory tokens

```
<evidence>
object: the person
position_frame_0_m: (0.8, 0.5, 4.8)
position_frame_6_m: (0.3, 0.6, 4.3)
displacement_m: 0.7
time_s: 1.20
speed_mps: 0.58
</evidence>
<answer>A</answer>
```

Figure 8: **Approaching person.** Endpoint displacement rate from frames 0 to 6: reconstruction-derived reference 0.59 m/s, frames-only evidence 1.53 m/s (wrong option), and trajectory-input evidence 0.58 m/s (correct option).



Camera motion

Moving in the world frame: 4 persons. Staying in place: the door, 12 paintings, 2 persons.

Question — **Right answer: C**

Approximately how far (in meters) did the camera move between frame 4 and frame 19 of 28?

A. 1.1 B. 0.78 **C. 0.58** D. 1.4

ground truth: 0.64 m

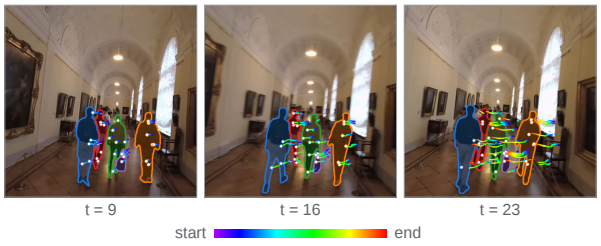
Qwen3-VL-8B · frames only

```
<evidence>
camera_frame_4_m: (-0.1, -0.0, 0.4)
camera_frame_19_m: (-0.7, -0.0, 1.5)
camera_displacement_m: 1.3
</evidence>
<answer>D</answer>
```

Ours · frames + trajectory tokens

```
<evidence>
camera_frame_4_m: (-0.1, 0.0, 0.1)
camera_frame_19_m: (-0.6, 0.0, 0.5)
camera_displacement_m: 0.6
</evidence>
<answer>C</answer>
```

Figure 9: **Camera motion.** Camera displacement from frames 4 to 19: reconstruction-derived reference 0.64 m, frames-only evidence 1.3 m (wrong option), and trajectory-input evidence 0.6 m (correct option).



Carried object

Moving in the world frame: 4 persons, the handbag. Staying in place: 6 paintings, the window, the chair.

The handbag moves about 0.9 m in the world frame at an average speed of 0.59 m/s.

Question — **Right answer: A**

In the world frame (ignoring camera motion), approximately how far (in meters) did the handbag move between frame 9 and frame 23 of 24?

A. 0.58 B. 1.4 C. 1.1 D. 0.78

ground truth: 0.62 m

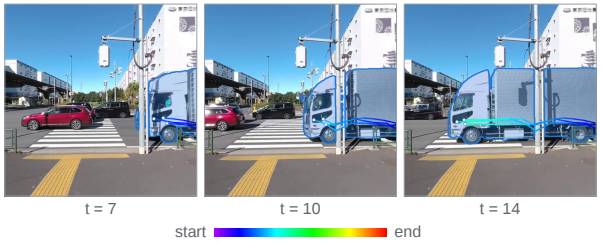
Qwen3-VL-8B · frames only

```
<evidence>
object: the handbag
position_frame_9_m: (0.8, 1.5, 5.6)
position_frame_23_m: (1.0, 1.6, 6.9)
displacement_m: 1.3
</evidence>
<answer>B</answer>
```

Ours · frames + trajectory tokens

```
<evidence>
object: the handbag
position_frame_9_m: (1.0, 1.5, 6.5)
position_frame_23_m: (1.2, 1.5, 7.1)
displacement_m: 0.6
</evidence>
<answer>A</answer>
```

Figure 10: **Carried object.** Handbag displacement from frames 9 to 23: reconstruction-derived reference 0.62 m, frames-only evidence 1.3 m (wrong option), and trajectory-input evidence 0.6 m (correct option).



Crossing truck

Moving in the world frame: the white box truck. Staying in place: 2 yellow tactile pavings, the traffic light pole, 2 green railings.

The white box truck moves about 16.4 m in the world frame at an average speed of 6.31 m/s; it accelerates over the clip; it reverses direction around frame 19.

Question — **Right answer: D**

In the world frame (ignoring camera motion), approximately how far (in meters) did the white box truck move between frame 7 and frame 14 of 34?

A. 12 B. 6.4 C. 4.7 **D. 8.7**

ground truth: 8.14 m

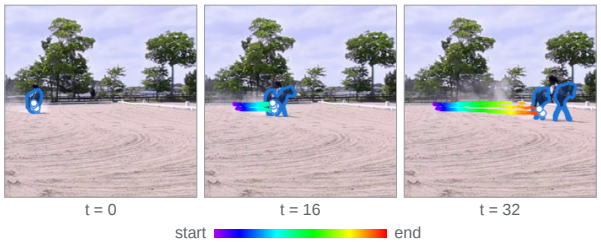
Qwen3-VL-8B · frames only

```
<evidence>
object: the white box truck
position_frame_7_m: (4.1, 1.0, 9.3)
position_frame_14_m: (-0.2, -0.2, 10.0)
displacement_m: 4.5
</evidence>
<answer>C</answer>
```

Ours · frames + trajectory tokens

```
<evidence>
object: the white box truck
position_frame_7_m: (5.6, 2.6, 13.6)
position_frame_14_m: (-2.7, 2.3, 14.8)
displacement_m: 8.4
</evidence>
<answer>D</answer>
```

Figure 11: **Crossing truck.** Truck displacement from frames 7 to 14: reconstruction-derived reference 8.14 m, frames-only evidence 4.5 m (wrong option), and trajectory-input evidence 8.4 m (correct option).



Distant animal

Moving in the world frame: **the horse**. Staying in place: 9 trees.

The horse moves about 1.4 m in the world frame at an average speed of 0.23 m/s; it accelerates over the clip.

Question **Right answer: C**

In the world frame (ignoring camera motion), what was the average speed (in meters per second) of **the horse** between frame 0 and frame 32 of 34?

A. 0.29 B. 0.39 **C. 0.21** D. 0.16 ground truth: 0.22 m/s

Qwen3-VL-8B · frames only

```
<evidence>
object: the horse
position_frame_0_m: (0.1, -0.0, 13.0)
position_frame_32_m: (2.6, 0.0, 10.4)
displacement_m: 3.6
time_s: 6.41
speed_mps: 0.56
</evidence>
<answer>B</answer>
```

Ours · frames + trajectory tokens

```
<evidence>
object: the horse
position_frame_0_m: (0.1, 0.2, 5.1)
position_frame_32_m: (1.4, 0.1, 4.7)
displacement_m: 1.4
time_s: 6.41
speed_mps: 0.22
</evidence>
<answer>C</answer>
```

Figure 12: **Distant animal.** Horse endpoint displacement rate from frames 0 to 32: reconstruction-derived reference 0.22 m/s, frames-only evidence 0.56 m/s (wrong option), and trajectory-input evidence 0.22 m/s (correct option).

H LIMITATIONS AND SCOPE

TrajQA-B is built from Stereo4D's reconstructed tracks. Its reference answers inherit errors in those tracks, masks, object assignments, and camera poses. Questions are deliberately defined to be answerable from the serialized records, so accuracy measures trajectory reading under this construction rather than recovery of 4D geometry from raw video or transfer to arbitrary motion tasks. Some original prompts use the phrase *average speed* for endpoint displacement rate; this wording should be corrected when regenerating the benchmark. External baseline comparisons differ in supervision, visual processing, and sometimes evaluation subset. The separately trained frames-only arm is the closest visual comparison, but its full optimization schedule is unavailable and its eight sampled frames lack timestamps; queried endpoints may be absent. Stage-1 training excludes QA evaluation scenes, but its source-video and validation-set relation to QA evaluation has not been independently audited in the available records. Results use one training seed, and no experiment isolates the contribution of part labels, event splitting, adaptive spline fitting, or projector warm-up. Future evaluation should use independent motion ground truth, documented stage-1 source-video separation, matched frame and timestamp inputs, and component ablations.