# TRACE4D: GROUNDED SPATIOTEMPORAL REASONING VIA 4D STATE SEQUENCES

**Anonymous authors**
Paper under double-blind review

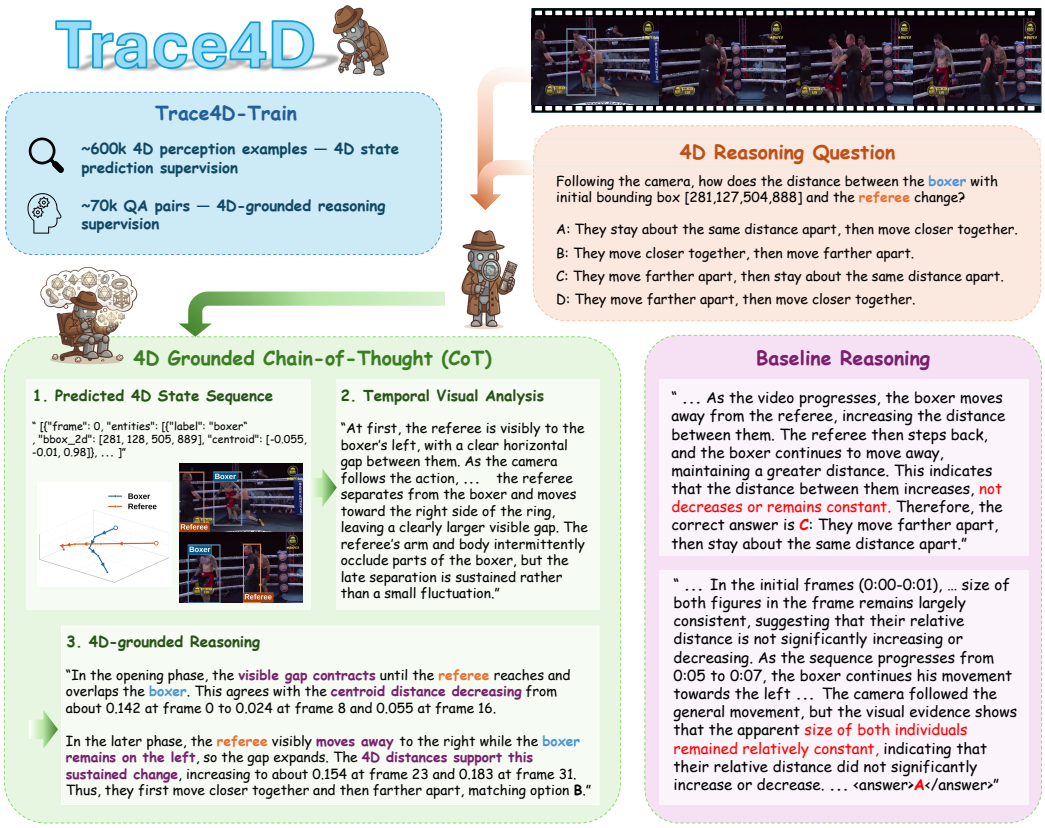



Figure 1: **Trace4D** connects structured 4D states with temporal visual observations for video reasoning. In the boxer–referee example, centroid-distance changes are matched to the visible approach and subsequent separation, supporting answer B. Baseline reasoning is from Qwen3-VL (top) and MLLM-4D (bottom).

## ABSTRACT

Understanding how objects and their spatial relationships evolve in 3D space over time remains challenging for vision-language models (VLMs), particularly under changing viewpoints. Recent approaches incorporate geometric information into reasoning, but explicit supervision for temporally ordered 4D state evolution and its use as intermediate evidence remains limited. We introduce **Trace4D**, which trains VLMs to predict 4D state sequences and reason over their inter-timestep changes alongside temporal visual observations. Its 4D-grounded chain-of-thought (CoT) explicitly represents 4D states, analyzes temporal visual context, and connects both as evidence for the answer. We construct Trace4D-Train from in-the-wild videos with 600K 4D perception examples and approximately 70K grounded QA examples, and train with joint supervised fine-tuning followed by geometry-aware GRPO. At inference, Trace4D predicts 4D states and reasoning



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.

directly from video without reference annotations or an external geometric model. Experiments on Trace4D-Bench and VLM4D show improved QA accuracy over the base VLMs and benefits from reasoning over 4D state changes and temporal visual observations.

## 1 INTRODUCTION

Understanding how objects and their spatial relationships evolve in 3D space over time is an important capability for video understanding, which we refer to as 4D understanding. Spatial relationships at a single moment provide only a partial account of an event; understanding its development requires relating observations across time. Videos capture only 2D projections of the 3D world, making changes in 3D position and spatial relationships difficult to infer directly. This makes 4D reasoning challenging for vision-language models (VLMs) (Yang et al., 2025; Zhou et al., 2025).

To address this challenge, recent approaches have equipped VLMs with geometric knowledge in different ways. Some incorporate pretrained geometric features or outputs (Liu et al., 2026; Zhou et al., 2026; Gao et al., 2026), providing spatial references that complement the VLM's visual representations. Others transfer 4D knowledge through QA supervision (Zhou et al., 2026; Cho et al., 2026), sometimes complemented by perception and tracking tasks. Recent approaches also extend chain-of-thought (CoT) reasoning (Wei et al., 2022) with geometric states, relations, and temporal visual evidence (Yin et al., 2026; Wang et al., 2025c). However, explicit supervision for predicting a temporally ordered 4D state sequence remains limited. Such a sequence makes inter-timestep changes explicit, allowing the model to compare states across time and use these changes as intermediate evidence for reasoning.

Grounded reasoning methods link intermediate reasoning steps to identifiable visual evidence. Image-based approaches localize relevant object regions and connect them to reasoning steps (Wu et al., 2026; Wang et al., 2026a), while video methods extend this principle to specific objects and timestamps (Han et al., 2025; Meng et al., 2025). For 4D understanding, the relevant evidence includes both how 3D states change over time and the visual context in which those changes occur. A 4D state sequence provides a basis for comparing states across time, while temporal visual observations describe actions, interactions, and occlusions that geometric states alone do not fully capture. This motivates grounded temporal reasoning that connects state changes to their visual context and uses both sources of evidence to support the final answer.

We introduce **Trace4D**, which learns to predict a 4D state sequence as an explicit intermediate representation for video reasoning. The sequence describes object and, when required, camera states across annotated timesteps. Our 4D-grounded CoT uses this sequence to interpret state changes alongside temporal visual observations and derive an answer (Figure 1). We construct Trace4D-Train with 600K 4D perception examples and approximately 70K grounded QA examples from in-the-wild videos to supervise state prediction and its use in reasoning. Joint supervised fine-tuning (SFT) learns both tasks together. Geometry-aware GRPO further rewards agreement with reference positions and inter-timestep displacements. At inference, the model generates its own state sequence and reasoning from the video and QA prompt without an external geometric model. Trace4D achieves the highest overall accuracy among evaluated models on our held-out **Trace4D-Bench** and improves overall accuracy over its base VLMs on VLM4D.

Our contributions are summarized as follows:

- We introduce **Trace4D**, which predicts temporally ordered 4D state sequences as explicit intermediate representations and grounds reasoning over their changes in temporal visual observations.
- We construct **Trace4D-Train** with 600K 4D perception and approximately 70K grounded QA examples from in-the-wild videos, supervising state prediction and its use in reasoning.
- Ablations show that reasoning over 4D state changes and geometry-aware GRPO improve performance, while temporal visual analysis provides complementary evidence for spatiotemporal reasoning.

## 2 Related Works

**VLMs for 3D and 4D Understanding.** General-purpose VLMs support video QA (Zhang et al., 2024; Bai et al., 2025b;a), but evaluations of spatial and dynamic scene understanding reveal persistent challenges (Yang et al., 2025; Zhou et al., 2025; 2026). To strengthen these capabilities, geometry-derived QA provides supervision for spatial relations in images (Chen et al., 2024a; Cheng et al., 2024) and kinematic changes along object trajectories (Ko et al., 2025). Multitask supervision further covers depth, correspondence, and camera/object motion (Xu et al., 2026), as well as 4D QA, tracking, and camera-pose tasks (Zhou et al., 2026; Cho et al., 2026; Yang et al., 2026). Geometric knowledge can also be supplied through pretrained features (Liu et al., 2026; Miao et al., 2026) or constructed as explicit evidence at inference time (Gao et al., 2026). Several works also use geometry as an explicit intermediate representation for reasoning: 3D locations and orientations enable spatial computation (Ma et al., 2026), while generated scene-layout maps guide reasoning through rewards for spatial accuracy (Ouyang et al., 2025). For video reasoning, relation graphs make inter-object geometry explicit (Wang et al., 2025c). MLLM-4D (Yin et al., 2026) predicts explicit 3D states at query-relevant start and end frames, analyzes the intervening visual progression, and rewards coordinate accuracy at those anchors. In contrast, Trace4D predicts a multi-timestep 4D state sequence as an explicit intermediate representation and uses its inter-timestep changes as reasoning evidence alongside temporal visual observations.

**Grounded reasoning for VLMs.** Chain-of-thought (CoT) prompting elicits intermediate reasoning before an answer (Wei et al., 2022). Early studies extend rationale supervision to multimodal QA (Lu et al., 2022; Zhang et al., 2023), while LLaVA-CoT (Xu et al., 2025) separates visual interpretation and reasoning into stages. Grounded reasoning links these steps to identifiable visual evidence. At the image level, VisCoT (Shao et al., 2024a) selects informative regions, while VoCoT (Li et al., 2025) incorporates localized object information into the reasoning sequence. This connection can be learned through stepwise grounding supervision (Wu et al., 2026) or joint localization and reasoning rewards (Wang et al., 2026a), with further visual feedback obtained by revisiting selected or zoomed regions (Wang et al., 2026b; Sarch et al., 2026). For video, evidence must also be associated with temporal locations: VideoEspresso (Han et al., 2025) annotates frames, objects, and textual evidence, while Open-o3-Video (Meng et al., 2025) links reasoning to timestamps, objects, and bounding boxes. Related settings ground motion reasoning through segmentation outputs (Deng et al., 2025) or connect 3D scene entities to reasoning steps and subproblems (Chen et al., 2026a; Linghu et al., 2026). Existing video-grounding approaches localize supporting evidence in space and time. Trace4D additionally supervises reasoning over predicted 4D state changes alongside the corresponding temporal visual observations.

## 3 Methodology

We present Trace4D, a pipeline for learning to predict 4D state sequences and use their changes together with temporal visual observations for video reasoning. We first construct reference state sequences and QA pairs from in-the-wild videos (Section 3.1). A multimodal teacher then produces 4D-grounded CoT that relates these state changes to temporal visual observations and uses both as evidence for the answer (Section 3.2). Finally, we jointly train the VLM on state prediction and grounded QA, followed by geometry-aware GRPO using answer correctness and geometric rewards (Section 3.3).

### 3.1 Dataset Generation Pipeline

Our data pipeline uses the geometric perception components and selected answer-derivation rules of DSR-Suite (Zhou et al., 2026). We further filter the resulting tracks and adapt the question and option templates to our clip-level setting.

**Video and entity curation.** We curate source videos from Koala-36M (Wang et al., 2025a) using its motion and video-training-suitability metadata to identify videos with observable scene or object dynamics. We identify candidate agents and foreground objects from the captions and use the Grounded SAM pipeline (Ren et al., 2024), with Grounding DINO (Liu et al., 2024) for localization and SAM 2 (Ravi et al., 2025) for mask tracking across frames.

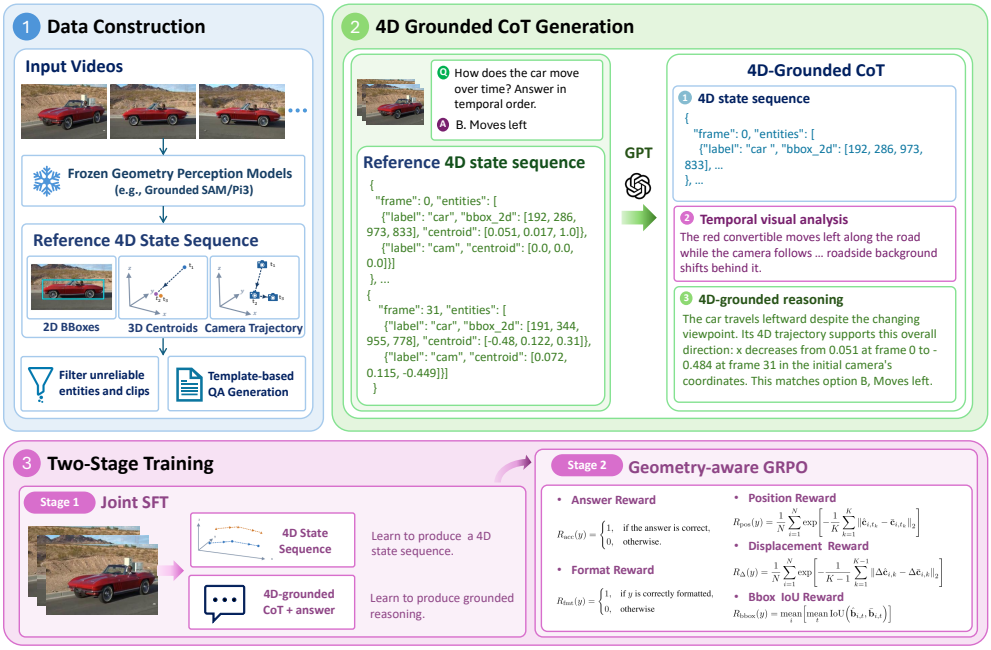



Figure 2: Overview of **Trace4D**. Reference 4D states support template-based QA construction and teacher-generated CoT that connects state changes with temporal visual observations. Joint SFT learns state prediction and grounded QA together, followed by geometry-aware GRPO using answer, format, position, displacement, and bounding-box rewards.

**Reference 4D state construction.** Pi3 (Wang et al., 2026c) estimates 3D point maps and camera poses. We obtain object centroids by averaging reconstructed points within each tracked mask and derive camera positions from the estimated poses. These estimates form our *reference 4D states* (Figure 3), distinct from student predictions.

For a clip represented by $T$ sampled frames, $V = (I_t)_{t=0}^{T-1}$, let $\mathcal{K} = \{t_1, \ldots, t_K\} \subseteq \{0, \ldots, T-1\}$ denote its $K$ annotated frame indices, ordered as $t_1 < \cdots < t_K$. Each state $(t, \mathcal{E}_t)$ describes the target entities at timestep $t$, and the 4D state sequence is

$$\mathcal{P} = \big[(t, \mathcal{E}_t)\big]_{t \in \mathcal{K}}, \qquad \mathcal{E}_t = \big\{(\ell_i, \mathbf{b}_{i,t}, \mathbf{c}_{i,t})\big\}_{i \in \mathcal{O}_t}. \tag{1}$$

Here, $\mathcal{O}_t$ indexes the annotated entities and $\ell_i$ is the label of entity $i$. The index $i$ refers to the same entity across timesteps. For an object, $\mathbf{b}_{i,t} \in \mathbb{R}^4$ is its normalized 2D box and $\mathbf{c}_{i,t} \in \mathbb{R}^3$ its normalized 3D centroid. Camera entries use $\ell_i = \texttt{cam}$, $\mathbf{c}_{i,t}$ for position, and $\mathbf{b}_{i,t} = \emptyset$ to denote an omitted box.

Appendix A.1 defines the serialized fields, frame indexing, and coordinate normalization.

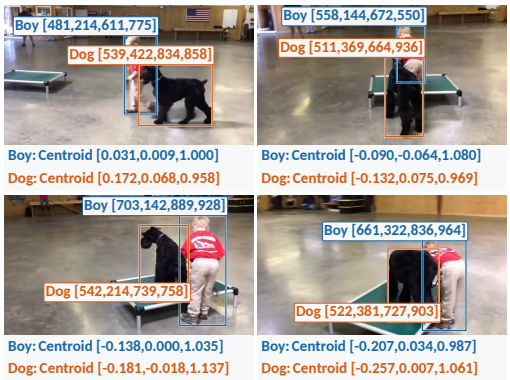



Figure 3: Example of 4D reference annotations. Colored boxes identify the same two entities across four frames; labels report their bounding-box coordinates and 3D centroids.

**Quality filtering and clip selection.** We filter the resulting object tracks using bounding-box size checks and a multimodal verifier that rejects empty, background, or incorrectly labeled target regions. We then select temporal windows with identifiable targets and valid required bounding boxes and 3D centroids, ensuring that their annotations are indexed to the sampled clip timeline. Unreadable clips that cannot support the required

frame sampling are excluded. The retained states serve as references for perception and QA supervision.

**Template-based QA construction.** We construct QA pairs under two coordinate-frame conventions. *Relative questions* express spatial relations and their changes in the camera frame at each timestep, while *Absolute questions* express them in a fixed anchor-camera frame for comparison over time. Here, *absolute* refers to a fixed reference frame rather than metric world coordinates.

For each question, we transform the reference 4D states into the required coordinate frame, yielding the task-specific reference sequence $\bar{\mathcal{P}}$. An overbar denotes a training target. We then derive the answer from the resulting temporal state changes using selected geometric rules adopted from DSR-Suite (Zhou et al., 2026). We revise the question and option templates for our clip-level setting and additionally construct motion questions within the Absolute group for object and camera translation over time. For task-specific distractors, we give priority to candidates with the same number of temporal phases as the reference answer to reduce answer-length cues. Appendix A.2 gives question templates, answer criteria, and distractor rules. Each resulting QA pair is retained with its source clip and aligned reference 4D states for subsequent CoT generation.

### 3.2 4D-Grounded CoT Generation

We next augment the QA pairs with 4D-grounded CoT annotations, keeping the questions, answer options, and reference answers fixed. Let $q$ denote the complete QA prompt, including the question, answer options, and task instruction. A multimodal teacher receives $V$, $q$, the reference 4D state sequence $\bar{\mathcal{P}}$, and the reference option label $a_{\mathrm{ref}} \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$ obtained during QA construction. The teacher constructs the following three-part CoT to connect visual observations with state changes in support of the reference answer.

*Step 1: Structured 4D state sequence.* The teacher copies the task-specific reference states unchanged into the `<4d>` block at every annotated timestep in $\mathcal{K}$, providing the student's state-prediction target.

*Step 2: Temporal visual analysis.* The `<4d>` block represents geometry but does not fully capture visual cues such as actions, interactions, and occlusion. The `<visual>` block therefore records question-relevant observations across temporal phases.

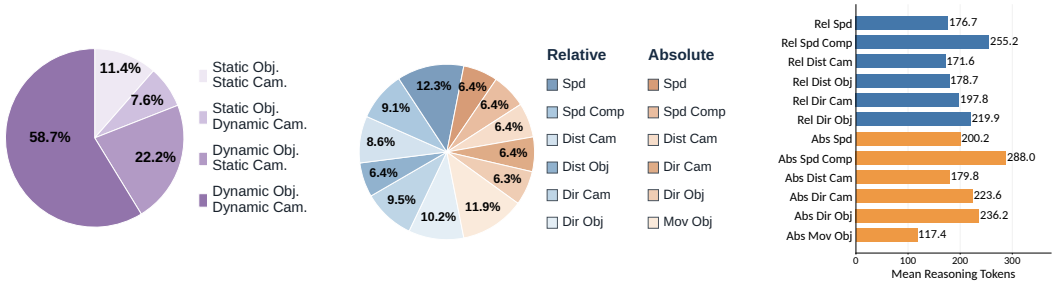
*Step 3: 4D-grounded reasoning.* Question-type-specific instructions require the teacher to connect a temporal visual observation with the relevant 4D state change for the same target, interval, and coordinate frame, then explain how this correspondence supports the answer. When the two sources differ, the teacher is instructed to re-examine relevant viewpoint or occlusion effects without forcing agreement.

Let $s$, $o$, $r$, and $a$ denote the token spans for the structured 4D state sequence, temporal visual analysis, 4D-grounded reasoning, and final answer. Each span includes its associated formatting tokens, so their concatenation forms the complete response $y = [s; o; r; a]$.

The teacher-generated target is $\bar{y} = [\bar{s}; \bar{o}; \bar{r}; \bar{a}]$, where $\bar{s}$ serializes $\bar{\mathcal{P}}$ and $\bar{a}$ encodes $a_{\mathrm{ref}}$. It is paired with $(V, q)$ for student training; the reference state sequence and correct-option label are not added as extra student inputs. We retain traces that preserve the reference states and answer and satisfy the required response structure. Appendix A.3 describes the teacher inputs and provides an example generation prompt.

**Training–evaluation splits and statistics.** Our data construction yields 600K *4D perception examples* from approximately 362K unique clips and 69,331 *grounded QA examples* (approximately 70K) from approximately 64K unique clips. A clip may contribute to either form of supervision or both. We call the training corpus **Trace4D-Train**. Splits are defined at the parent source-video level, with held-out sources excluded from all training, including GRPO and ablations. **Trace4D-Bench** is the held-out evaluation set of 1,200 QA pairs: 600 Relative and 600 Absolute questions, with 100 questions per type.

Figure 4 summarizes object–camera dynamics, question-type coverage, and reasoning length in the constructed dataset. Joint object and camera motion forms the largest dynamics category, while speed-comparison questions have the longest mean reasoning traces in both coordinate settings.




(a) Dynamics distribution (b) Question-type distribution (c) Mean reasoning length

Figure 4: Dataset statistics. (a) Joint distribution of object and camera dynamics. (b) Question-type proportions for relative and absolute QA. (c) Mean reasoning token counts by question type.

### 3.3 Two-Stage Training

We train Trace4D in two stages: joint supervised fine-tuning (SFT) on 4D perception and grounded QA, followed by geometry-aware GRPO.

**Stage 1: Joint SFT.** Starting from the base VLM, we jointly train on Trace4D-Train's 600K perception and approximately 70K grounded QA examples. A perception example provides a clip $V$ and an instruction $u$ specifying the targets and coordinate convention. The target serializes the reference 4D states $\bar{\mathcal{P}}$ within `<4d>`, without visual analysis or an answer. A grounded QA example provides $(V, q)$ and supervises the full response $\bar{y}$. Appendix B.4 shows both task instructions; Section 4.4 compares joint SFT with separate perception pretraining.

**Stage 2: Geometry-aware GRPO.** Starting from the joint-SFT checkpoint, we apply Group Relative Policy Optimization (GRPO) (Shao et al., 2024b) on approximately 7K QA examples to reward answer correctness and geometric consistency. Reference states and answer labels are used only for reward computation. The position reward $R_{\text{pos}}$ rewards position accuracy. The displacement reward $R_{\Delta}$ compares coordinate-change vectors between consecutive annotated timesteps, accounting for direction and magnitude. Both rewards are computed from predicted states in the generated `<4d>` block. For $N$ entities, including the camera when required, matched across $K \geq 2$ valid annotated timesteps, these rewards are

$$
\begin{aligned}
R_{\text{pos}}(y) &= \frac{1}{N}\sum_{i=1}^{N}\exp\left[-\frac{1}{\tau_{\text{pos}}K}\sum_{k=1}^{K}\left\|\hat{\mathbf{c}}_{i,t_k}-\bar{\mathbf{c}}_{i,t_k}\right\|_2\right], \\
R_{\Delta}(y) &= \frac{1}{N}\sum_{i=1}^{N}\exp\left[-\frac{1}{\tau_{\Delta}(K-1)}\sum_{k=1}^{K-1}\left\|\Delta\hat{\mathbf{c}}_{i,k}-\Delta\bar{\mathbf{c}}_{i,k}\right\|_2\right],
\end{aligned}
\tag{2}
$$

where $\hat{\mathbf{c}}$ and $\bar{\mathbf{c}}$ are predicted and reference positions under the same coordinate and normalization conventions, $\Delta\mathbf{c}_{i,k} = \mathbf{c}_{i,t_{k+1}} - \mathbf{c}_{i,t_k}$, and $\tau_{\text{pos}} = \tau_{\Delta} = 1$. Absolute displacements use the fixed anchor-camera frame; relative displacements use camera-relative coordinates.

The binary rewards $R_{\text{acc}}$ and $R_{\text{fmt}}$ score answer correctness and response format; $R_{\text{bbox}}$ averages bounding-box IoU over matched objects. The combined reward is

$$
R(y) = w_{\text{acc}}R_{\text{acc}}(y) + w_{\text{fmt}}R_{\text{fmt}}(y) + w_{\text{pos}}R_{\text{pos}}(y) + w_{\Delta}R_{\Delta}(y) + w_{\text{bbox}}R_{\text{bbox}}(y), \tag{3}
$$

where the weights sum to one. Appendix B.3 specifies the reward weights, format criteria, and optimization settings.

**Inference.** Given $(V, q)$, Trace4D generates $y = [s; o; r; a]$ autoregressively. The predicted 4D state sequence remains in the context for subsequent temporal visual analysis and 4D-grounded reasoning. Reference 4D states and answer labels are not supplied at inference, and no external geometric model is invoked. Appendix D shows examples of the generated responses.

| Models | Relative Spatial Relations | | | | | | | Absolute Spatial Relations | | | | | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Spd | Spd Comp | $\text{Dist}_{\text{Cam}}$ | $\text{Dist}_{\text{Obj}}$ | $\text{Dir}_{\text{Cam}}$ | $\text{Dir}_{\text{Obj}}$ | **Avg.** | Spd | Spd Comp | $\text{Dist}_{\text{Cam}}$ | $\text{Dir}_{\text{Cam}}$ | $\text{Dir}_{\text{Obj}}$ | Motion | **Avg.** | |
| *Proprietary Models* | | | | | | | | | | | | | | | |
| GPT-4o (Hurst et al., 2024) | 33.0 | 32.0 | 21.0 | 13.0 | 47.0 | 32.0 | 29.67 | 31.0 | 32.0 | 36.0 | 38.0 | 35.0 | 54.0 | 37.67 | 33.67 |
| GPT-5 (Singh et al., 2025) | 32.0 | 37.0 | 58.0 | 47.0 | 68.0 | 54.0 | 49.33 | 35.0 | 36.0 | 50.0 | 40.0 | 43.0 | 60.0 | 44.00 | 46.67 |
| *General VLMs* | | | | | | | | | | | | | | | |
| Qwen2.5-VL-3B (Bai et al., 2025b) | 40.0 | 47.0 | 25.0 | 35.0 | 21.0 | 22.0 | 31.67 | 42.0 | 49.0 | 31.0 | 24.0 | 29.0 | 33.0 | 34.67 | 33.17 |
| Qwen3-VL-4B (Bai et al., 2025a) | 26.0 | 38.0 | 30.0 | 41.0 | 41.0 | 53.0 | 38.17 | 29.0 | 33.0 | 41.0 | 38.0 | 47.0 | 50.0 | 39.67 | 38.92 |
| Qwen2.5-VL-7B | 39.0 | 48.0 | 32.0 | 26.0 | 40.0 | 32.0 | 36.17 | 33.0 | 40.0 | 35.0 | 49.0 | 26.0 | 55.0 | 39.67 | 37.92 |
| Qwen3-VL-8B | 32.0 | 43.0 | 36.0 | 31.0 | 55.0 | 44.0 | 40.17 | 37.0 | 41.0 | 31.0 | 49.0 | 39.0 | 55.0 | 42.00 | 41.08 |
| InternVL2.5-4B (Chen et al., 2024b) | 27.0 | 41.0 | 21.0 | 30.0 | 35.0 | 46.0 | 33.33 | 39.0 | 41.0 | 24.0 | 37.0 | 40.0 | 36.0 | 36.17 | 34.75 |
| InternVL3.5-4B (Wang et al., 2025b) | 41.0 | 43.0 | 30.0 | 43.0 | 40.0 | 39.0 | 39.33 | 42.0 | 36.0 | 37.0 | 41.0 | 42.0 | 46.0 | 40.67 | 40.00 |
| InternVL2.5-8B | 41.0 | 38.0 | 43.0 | 38.0 | 49.0 | 37.0 | 41.00 | 39.0 | 37.0 | 28.0 | 50.0 | 37.0 | 42.0 | 38.83 | 39.92 |
| InternVL3.5-8B | 34.0 | 45.0 | 33.0 | 32.0 | 33.0 | 48.0 | 37.50 | 37.0 | 39.0 | 28.0 | 50.0 | 37.0 | 55.0 | 41.00 | 39.25 |
| LLaVA-OneVision1.5-4B (An et al., 2025) | 30.0 | 43.0 | 26.0 | 29.0 | 34.0 | 42.0 | 34.00 | 22.0 | 35.0 | 29.0 | 38.0 | 54.0 | 43.0 | 36.83 | 35.42 |
| LLaVA-OneVision1.5-8B | 25.0 | 38.0 | 20.0 | 30.0 | 31.0 | 48.0 | 32.00 | 21.0 | 32.0 | 26.0 | 34.0 | 41.0 | 52.0 | 34.33 | 33.17 |
| *3D/4D Specialized Models* | | | | | | | | | | | | | | | |
| VG-LLM-8B (Zheng et al., 2026) | 39.0 | 45.0 | 36.0 | 43.0 | 87.0 | 18.0 | 44.67 | 44.0 | 47.0 | 38.0 | 91.0 | 25.0 | 44.0 | 48.17 | 46.42 |
| VLM-3R-7B (Fan et al., 2026) | 40.0 | 45.0 | 27.0 | 38.0 | 32.0 | 21.0 | 33.83 | 40.0 | 46.0 | 32.0 | 44.0 | 31.0 | 51.0 | 40.67 | 37.25 |
| MLLM-4D-8B (Yin et al., 2026) | 35.0 | 43.0 | 51.0 | 46.0 | 69.0 | 41.0 | 47.50 | 34.0 | 50.0 | 41.0 | 62.0 | 31.0 | 61.0 | 46.50 | 47.00 |
| GSM-7B (Zhou et al., 2026) | 43.0 | 45.0 | 33.0 | 34.0 | 74.0 | 28.0 | 42.83 | 42.0 | 46.0 | 33.0 | 77.0 | 30.0 | 47.0 | 45.83 | 44.33 |
| 4DThinker-3B (Chen et al., 2026b) | 43.0 | 42.0 | 61.0 | 52.0 | 32.0 | 34.0 | 44.00 | 50.0 | 37.0 | 56.0 | 41.0 | 29.0 | 26.0 | 39.83 | 41.92 |
| **Trace4D (Qwen3-VL-4B)** | **79.0** | **71.0** | 85.0 | 61.0 | **97.0** | **93.0** | 81.00 | **73.0** | 59.0 | 75.0 | 95.0 | **93.0** | **70.0** | **77.50** | **79.25** |
| **Trace4D (Qwen3-VL-8B)** | **79.0** | 69.0 | **89.0** | **68.0** | 96.0 | 86.0 | **81.17** | 68.0 | **63.0** | **77.0** | **96.0** | 92.0 | 65.0 | 76.83 | 79.00 |

Table 1: Comparison of vision-language models on Trace4D-Bench, covering relative and absolute spatial and motion understanding.

# 4 EXPERIMENTS

We evaluate whether Trace4D can use predicted 4D state changes and temporal visual observations to answer spatiotemporal questions. We evaluate the trained model on our held-out Trace4D-Bench and examine transfer to the external VLM4D benchmark. Our ablations study the effects of state-change reasoning, temporal visual evidence, joint SFT, and GRPO. We further analyze the relationship between 4D perception quality and QA accuracy. Detailed GRPO reward ablations are provided in Appendix C.

## 4.1 EXPERIMENTAL SETTINGS

We adapt Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct (Bai et al., 2025a) using LoRA (Hu et al., 2021) on the language model and multimodal projector, while keeping the vision backbone frozen. Trace4D results use joint SFT followed by geometry-aware GRPO. We evaluate multiple-choice QA accuracy on Trace4D-Bench and VLM4D. All ablations use the 4B model; CoT ablations use matched QA adaptation without GRPO, while the training-strategy ablation compares different SFT recipes and their GRPO continuation. Further details on data mixtures, optimization, task instructions, and decoding settings are provided in Appendix B.

## 4.2 RESULTS ON TRACE4D-BENCH

Trace4D-Bench evaluates distance, direction, speed, and translation under specified relative and absolute coordinate frames. Its questions use the same task formulation as Trace4D-Train, with source videos held out from training.

We compare three baseline groups: proprietary models GPT-4o (Hurst et al., 2024) and GPT-5 (Singh et al., 2025); general-purpose VLMs Qwen2.5-VL (Bai et al., 2025b), Qwen3-VL (Bai et al., 2025a), InternVL2.5/3.5 (Chen et al., 2024b; Wang et al., 2025b), and LLaVA-OneVision1.5 (An et al., 2025); and 3D/4D-specialized models VG-LLM (Zheng et al., 2026), VLM-3R (Fan et al., 2026), MLLM-4D (Yin et al., 2026), GSM (Zhou et al., 2026), and 4DThinker (Chen et al., 2026b). Table 1 reports task-wise QA accuracy (%) across the spatial and motion reasoning categories.

Trace4D achieves overall accuracies of $79.25\%$ and $79.00\%$ with the 4B and 8B models, respectively, outperforming the evaluated proprietary, general-purpose, and 3D/4D-specialized baselines. Both models improve over their corresponding base VLMs by more than 37 percentage points. Each model also outperforms the evaluated baselines across all 12 question types, covering both relative and absolute reasoning on unseen videos. The ablations in Section 4.4 further examine the roles of reasoning over 4D state changes and incorporating temporal visual observations.

### 4.3 RESULTS ON VLM4D

VLM4D (Zhou et al., 2025) evaluates spatiotemporal reasoning in real and synthetic videos, including first- and third-person views. Its multiple-choice questions target motion, perspective, and temporal continuity in the videos, while Trace4D-Bench constructs questions from tracked 4D states under specified coordinate conventions. Their distinct video collections and question formulations provide complementary evaluations of 4D understanding.

| Model | Syn. | Real. | Avg. |
|---|---|---|---|
| *Proprietary Models* | | | |
| GPT-4o (Hurst et al., 2024) | 54.38 | 56.02 | 55.62 |
| GPT-5 (Singh et al., 2025) | 62.70 | 74.69 | 71.75 |
| *General VLMs* | | | |
| Qwen2.5-VL-3B (Bai et al., 2025b) | 35.96 | 47.34 | 44.55 |
| Qwen3-VL-4B (Bai et al., 2025a) | 48.76 | 57.55 | 55.40 |
| Qwen2.5-VL-7B | 48.31 | 51.79 | 50.94 |
| Qwen3-VL-8B | 55.73 | <u>60.25</u> | 59.14 |
| InternVL2.5-4B (Chen et al., 2024b) | 32.81 | 50.84 | 46.42 |
| InternVL3.5-4B (Wang et al., 2025b) | 42.24 | 56.38 | 52.92 |
| InternVL2.5-8B | 40.45 | 55.80 | 52.04 |
| InternVL3.5-8B | 46.07 | 55.87 | 53.47 |
| LLaVA-OneVision1.5-4B (An et al., 2025) | 40.00 | 44.42 | 43.34 |
| LLaVA-OneVision1.5-8B | 40.45 | 45.22 | 44.05 |
| *3D/4D VLMs* | | | |
| VG-LLM-8B (Zheng et al., 2026) | 34.83 | 48.87 | 45.43 |
| VLM-3R-7B (Fan et al., 2026) | 35.28 | 52.88 | 48.57 |
| MLLM-4D-8B (Yin et al., 2026) | 55.05 | **63.09** | 61.12 |
| GSM-7B (Zhou et al., 2026) | 50.11 | 47.56 | 48.18 |
| 4DThinker-3B (Chen et al., 2026b) | 24.72 | 38.95 | 35.46 |
| **Trace4D (Qwen3-VL-4B)** | **77.30** | 60.10 | **64.32** |
| **Trace4D (Qwen3-VL-8B)** | <u>68.09</u> | 59.08 | <u>61.29</u> |

Table 2: Performance on VLM4D. **Bold** and <u>underline</u> denote the best and second-best scores among non-proprietary models.

As shown in Table 2, Trace4D reaches $64.32\%$ with the 4B model and $61.29\%$ with the 8B model, exceeding their respective base VLMs by $8.92$ and $2.15$ percentage points. Both models outperform the evaluated general-purpose VLMs in overall accuracy. The gains over the base VLMs are larger on synthetic videos, while real-video accuracy improves for 4B but decreases for 8B. These results indicate transfer to a different video collection and question formulation, with gains that vary across model sizes and video subsets. Notably, the 4B model outperforms the 8B model on VLM4D and achieves a substantially larger improvement over its base VLM. One possible explanation is a capacity–data mismatch: the fixed amount of task-specific supervision may not fully exploit the larger model's capacity, with this limitation becoming more apparent under the distribution shift in VLM4D.

### 4.4 ABLATION AND ANALYSIS

**CoT ablation.** We compare four response formulations to examine the roles of reasoning over 4D state changes and incorporating temporal visual observations. All variants start from the same perception-pretrained 4B checkpoint and undergo matched QA adaptation without GRPO. Only the QA response supervision and task instructions differ; QA pairs, perception replay, and optimization settings are fixed. This controlled setting is distinct from the final method's joint SFT. *State-only* predicts the structured state sequence within `<4d>` and directly answers. *4D-state CoT* adds reasoning over the predicted state changes, without explicit visual analysis. *Visual-only CoT* reasons from video observations without generating a `<4d>` block. Our *4D + Visual CoT* connects predicted 4D state changes with temporal visual observations of the queried entities over matching time intervals. Table 3 reports both benchmarks; Appendix B.2 describes the shared SFT schedule and perception replay.

| CoT Variant | Trace4D-Bench | VLM4D |
|---|---|---|
| State-only | 74.83 | 52.86 |
| 4D-state CoT | **79.42** | 54.68 |
| Visual-only CoT | 73.75 | <u>59.25</u> |
| **4D + Visual CoT** | <u>77.25</u> | **60.52** |

Table 3: CoT ablation. Bold and underlined values indicate the best and second-best results, respectively.

Supervising reasoning over 4D state changes improves QA accuracy beyond state prediction alone. Compared with *State-only*, *4D-state CoT* improves accuracy by $4.59$ percentage points on Trace4D-Bench and $1.82$ points on VLM4D. This comparison supports supervising how state changes lead to an answer, in addition to supervising the states themselves.

Combining state-based reasoning with temporal visual analysis yields the highest VLM4D accuracy among the four variants. Our *4D + Visual CoT* achieves $77.25\%$ on Trace4D-Bench and $60.52\%$ on VLM4D, improving over *Visual-only CoT* on both benchmarks. While *4D-state CoT* performs best on Trace4D-Bench ($79.42\%$), adding temporal visual analysis raises VLM4D accuracy from $54.68\%$ to $60.52\%$. These results suggest that temporal visual observations provide complementary evidence for interpreting 4D state changes.

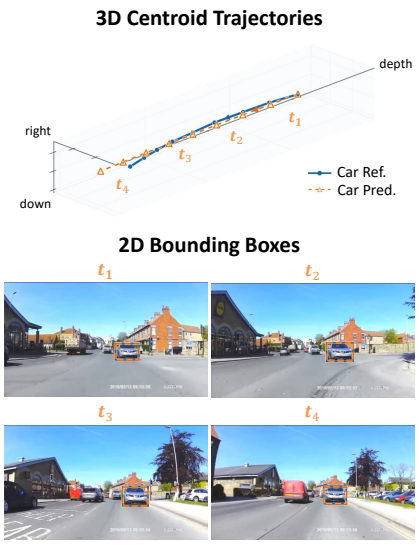



Figure 5: Predicted 3D centroid trajectories and 2D bounding boxes across timesteps.

| Training Strategy | Rel. | Abs. | VLM4D |
|---|---|---|---|
| Adaptation-only SFT | 76.33 | 67.50 | 57.54 |
| Perception-first SFT | 79.17 | 75.33 | 60.52 |
| Joint SFT | 79.33 | 76.33 | 62.39 |
| **Joint SFT + GRPO (Trace4D)** | **81.00** | **77.50** | **64.32** |

Table 4: Training strategy comparison.

| Setting | IoU ↑ | Centroid Error ↓ | Trajectory L2 ↓ |
|---|---|---|---|
| Relative | 0.7556 | 0.0994 | 0.0852 |
| Absolute | 0.7176 | 0.2654 | 0.1428 |

Table 5: Trace4D (4B) state prediction.

| Setting | Low L2 | Mid L2 | High L2 |
|---|---|---|---|
| Relative | 89.90 | 81.37 | 71.72 |
| Absolute | 91.92 | 75.00 | 65.66 |

Table 6: QA accuracy (%) by L2 error group.

**Training strategy.** Table 4 compares three SFT recipes and the GRPO continuation of Joint SFT. *Adaptation-only SFT* performs QA adaptation from the base model without perception supervision. *Perception-first SFT* first performs one epoch of perception pretraining, followed by QA adaptation with perception replay. *Joint SFT* jointly mixes perception and grounded QA supervision from the base model. Details are provided in Appendix B.2.

Adding perception supervision substantially improves over *Adaptation-only SFT*. *Joint SFT* further improves over *Perception-first SFT* across all three evaluation settings, indicating a benefit from jointly learning perception and grounded QA. Continuing Joint SFT with GRPO on 7,008 QA examples further improves performance in both coordinate settings and on VLM4D.

**Perception and reasoning analysis.** We evaluate 4D perception on clips held out at the source-video level. Table 5 reports bounding-box IoU, centroid error, and trajectory L2, defined in Appendix B.5. Absolute evaluation, in which object states must be expressed relative to a fixed anchor viewpoint, yields larger centroid and trajectory errors than Relative evaluation. Figure 5 shows a predicted car trajectory following the reference with deviations at individual timesteps.

We examine the association between trajectory L2 error and QA accuracy. Within each question type, we divide responses into three approximately equal-sized groups by error and then average QA accuracy across question types. Table 6 shows that QA accuracy decreases as trajectory error increases in both settings, with a larger low-to-high error gap in the Absolute setting (26.26 vs. 18.18 percentage points). This pattern suggests a connection between 4D perception quality and downstream reasoning performance.

## 5 Conclusion

We presented Trace4D, a framework for grounded spatiotemporal reasoning via predicted 4D state sequences. Trace4D predicts temporally ordered 4D states and uses their inter-timestep changes together with temporal visual observations as intermediate evidence for reasoning. Experiments on Trace4D-Bench and VLM4D show improved spatiotemporal reasoning, while ablations demonstrate the benefits of explicit 4D state-change reasoning, complementary temporal visual evidence, and geometry-aware optimization. Our analysis further reveals a connection between 4D perception quality and downstream QA performance, highlighting the role of explicit 4D states in bridging perception and reasoning.

### AI USE STATEMENT

In this work, we used generative AI tools to generate chain-of-thought (CoT) annotations for our training data. We have not used generative AI tools for other research tasks requiring disclosure, including proposing or refining hypotheses or interpreting results. Additionally, we used generative AI tools to improve the grammar, readability, and flow of the manuscript. All AI-assisted work was reviewed and verified by the authors, and we take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

## REPRODUCIBILITY STATEMENT

We describe dataset construction, coordinate normalization, and chain-of-thought generation in Appendix A. Training hyperparameters, geometry-aware GRPO rewards, and evaluation protocols are detailed in Appendix B, including the model-specific reward weights and decoding settings. Training and inference prompts are provided in Appendix B.4. We plan to publicly release our code and trained model checkpoints upon publication to facilitate reproduction of our results.

## REFERENCES

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

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

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14455–14465, 2024a.

Yiming Chen, Zekun Qi, Wenyao Zhang, Xin Jin, Li Zhang, and Peidong Liu. Reasoning in space via grounding in the world. In *International Conference on Learning Representations*, volume 2026, pp. 65549–65570, 2026a.

Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xiang An, Bo Li, Xin Xie, ZiDong Wang, Mingze Sun, Shuang Chen, Hongyu Li, et al. 4dthinker: Thinking with 4d imagery for dynamic spatial understanding. *arXiv preprint arXiv:2605.05997*, 2026b.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

Seokju Cho, Abhishek Badki, Hang Su, Jindong Jiang, Ziyao Zeng, Seungryong Kim, Sifei Liu, and Orazio Gallo. 4dp-qa: Scalable qa for 4d perception in vision language models. *arXiv preprint arXiv:2606.11568*, 2026.

Andong Deng, Tongjia Chen, Shoubin Yu, Taojiannan Yang, Lincoln Spencer, Yapeng Tian, Ajmal Saeed Mian, Mohit Bansal, and Chen Chen. Motion-grounded video reasoning: Understanding and perceiving motion at pixel level. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8625–8636. IEEE, 2025.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao Wang, Huaizhi Qu, Shijie Zhou, et al. 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.

Rongxin Gao, Yuzhi Huang, Dongxuan Liu, Chu Li, Zhenye Wang, Jie Wu, Shuzhao Xie, Jingyan Jiang, Xinghao Ding, Xiaotong Tu, et al. Dyntrace: Tracking dynamic object evidence for 4d spatio-temporal reasoning in mllms. *arXiv preprint arXiv:2607.12503*, 2026.

Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. Videoespresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26181–26191. IEEE, 2025.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*, 2024.

Dohwan Ko, Sihyeon Kim, Yumin Suh, Minseo Yoon, Manmohan Chandraker, Hyunwoo J Kim, et al. St-vlm: Kinematic instruction tuning for spatio-temporal reasoning in vision-language models. *arXiv preprint arXiv:2503.19355*, 2025.

Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, Xuan-Jing Huang, and Zhongyu Wei. Vocot: Unleashing visually grounded multi-step reasoning in large multi-modal models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3769–3798, 2025.

Xiongkun Linghu, Jiangyong Huang, Ziyu Zhu, Baoxiong Jia, and Siyuan Huang. Scenecot: Eliciting grounded chain-of-thought reasoning in 3d scenes. In *International Conference on Learning Representations*, volume 2026, pp. 2754–2779, 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Zhaochen Liu, Limeng Qiao, Guanglu Wan, and Tingting Jiang. Geoalign: Geometric feature realignment for mllm spatial reasoning. *arXiv preprint arXiv:2604.12630*, 2026.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. *Advances in Neural Information Processing Systems*, 38:140751–140774, 2026.

Jiahao Meng, Xiangtai Li, Haochen Wang, Yue Tan, Tao Zhang, Lingdong Kong, Yunhai Tong, Anran Wang, Zhiyang Teng, Yujing Wang, et al. Open-o3-video: Grounded video reasoning with explicit spatio-temporal evidence. *arXiv preprint arXiv:2510.20579*, 2025.

Deshui Miao, Xingsen Huang, Yameng Gu, Xin Li, Haijun Zhang, and Ming-Hsuan Yang. Geoweaver: Grounding visual tokens with geometric evidence before scene reasoning. *arXiv preprint arXiv:2605.22558*, 2026.

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

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning. *Advances in Neural Information Processing Systems*, 38:150977–151013, 2026.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. OpenAI GPT-5 System Card. *arXiv preprint arXiv:2601.03267*, 2025.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, et al. Traceable evidence enhanced visual grounded reasoning: Evaluation and method. In *International Conference on Learning Representations*, volume 2026, pp. 148769–148794, 2026a.

Jiacong Wang, Zijian Kang, Haochen Wang, Xiao Liang, Ya Wang, Jiawen Li, Bohong Wu, Jiao Ran, Haiyong Jiang, Chao Feng, et al. Vgr: Visual grounded reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 89506–89528, 2026b.

Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, et al. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8428–8437. IEEE, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Wentao Wang, Heqing Zou, Tianze Luo, Rui Huang, Yutian Zhao, Zhuochen Wang, Hansheng Zhang, Chengwei Qin, Yan Wang, Lin Zhao, et al. Video-str: Reinforcing mllms in video spatio-temporal reasoning with relation graph. *arXiv preprint arXiv:2510.10976*, 2025c.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi^3$: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Qiong Wu, Xiangcong Yang, Yiyi Zhou, Chenxin Fang, Baiyang Song, Xiaoshuai Sun, and Rongrong Ji. Grounded chain-of-thought for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33577–33587, 2026.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2087–2098. IEEE, 2025.

Runsen Xu, Weiyao Wang, Hao Tang, Xingyu Chen, Xiaodong Wang, Fu-Jen Chu, Matt Feiszli, and Kevin J Liang. Multi-spatialmllm: Multi-frame spatial understanding with multi-modal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31078–31088, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643. IEEE, 2025.

Jihan Yang, Zifan Zhao, Xichen Pan, Shusheng Yang, Junyi Zhang, Bingyi Kang, Hu Xu, Shang-Wen Li, and Saining Xie. Cambrian-p: Pose-grounded video understanding. *arXiv preprint arXiv:2605.22819*, 2026.

Xingyilang Yin, Chengzhengxu Li, Jiahao Chang, Chi-Man Pun, and Xiaodong Cun. Mllm-4d: Towards visual-based spatial-temporal intelligence. *arXiv preprint arXiv:2603.00515*, 2026.

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

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

Duo Zheng, Yanyang Li, Liwei Wang, et al. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *Advances in neural information processing systems*, 38:20560–20586, 2026.

Shengchao Zhou, Yuxin Chen, Yuying Ge, Wei Huang, Jiehong Lin, Ying Shan, and Xiaojuan Qi. Learning to reason in 4d: Dynamic spatial understanding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9637–9646, 2026.

Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. Vlm4d: Towards spatiotemporal awareness in vision language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8600–8612. IEEE, 2025.

# APPENDIX

## A DATA GENERATION AND DATASET DETAILS

### A.1 STATE REPRESENTATION AND COORDINATE NORMALIZATION

Each video input contains 32 sampled frames, indexed from 0 to 31. Reference states may cover a subset of these timesteps; state and anchor frame indices use the same sampled timeline. The reference 4D state sequence is serialized as a time-ordered JSON array within `<4d>`, with `frame` and `entities` fields. Object entries contain `label`, `bbox_2d`, and `centroid`. Camera entries, when required for absolute tasks, use the label `cam` and omit the `bbox_2d` field.

Bounding-box coordinates are normalized by image width and height and scaled to $[0, 1000]$. The 3D axes point right, down, and forward. For perception examples and spatial QA, 3D positions are divided by the largest positive depth of the referenced objects at the first clip frame. Motion QA uses the median valid positive object depth within the selected window. Each example uses one scale across all targets and annotated timesteps, yielding dimensionless 3D coordinates.

### A.2 QA CONSTRUCTION

Table 7 lists the templates. *[Prefix]* and *[Camera]* are "Following the camera," and "the camera" for relative questions, and "Anchored to frame {anchor} of the clip," and "the anchor camera" for absolute spatial questions. Objects are identified by category and, when needed, an initial box in $[0, 1000]$. Object-motion questions use separate paraphrases; their instructions specify the fixed anchor-camera frame. Table 8 gives geometric quantities and answer forms.

| **Question type** | **Setting** | **Question template** |
|---|---|---|
| Camera–object distance | Rel./Abs. | *[Prefix]* how does the distance between *[Object 1]* and *[Camera]* change? |
| Object–object distance | Rel. | *[Prefix]* how does the distance between *[Object 1]* and *[Object 2]* change? |
| Camera–object direction | Rel./Abs. | *[Prefix]* how does the direction from *[Camera]* to *[Object 1]* change? |
| Object–object direction | Rel./Abs. | *[Prefix]* how does the direction from *[Object 1]* to *[Object 2]* change? |
| Speed | Rel./Abs. | *[Prefix]* how does the speed of *[Object 1]* change? |
| Speed comparison | Rel./Abs. | *[Prefix]* compare the speed between *[Object 1]* and *[Object 2]*. |
| Object motion | Abs. | How does *[Object 1]* move over time? Answer in temporal order. |

Table 7: Question templates for the 12 benchmark categories. Shared templates are grouped across coordinate settings. The object-motion row shows one of six paraphrases used in the data.

**Reference-answer derivation.** Table 9 defines the quantities and answer criteria in the question's reference frame. Answers combine one or two ordered phases; directions may combine axes (e.g., "front-left").

**Filtering.** Spatial QA retains tracks whose box area averages at least $1\%$ across frames and never exceeds $50\%$ of the image. A multimodal verifier rejects empty or background regions and clear label mismatches. We require valid target boxes and 3D centroids and readable clips. Two-target candidates require initial bounding-box IoU at most $0.5$. Selected windows within each video and QA collection have pairwise temporal IoU at most $0.5$.

**Distractors.** We favor the reference answer's phase count and exclude duplicate or equivalent options. For motion QA, we replace distractors consistent with the reference trajectory to retain one correct answer.

| Question family | Reference quantity | Example choice text |
|---|---|---|
| Camera–object distance | Distance from the object to the current or anchor-camera origin. | It moves closer to the camera, then moves farther from the camera. |
| Object–object distance | Distance between two object centroids. | They move closer together, then move farther apart. |
| Direction | Directed 3D offset from the first entity to the second. | It is to the front, then to the front-right. |
| Speed | Magnitude of consecutive coordinate increments. | It slows down, then moves at a steady pace. |
| Speed comparison | Two objects' increment magnitudes over corresponding intervals. | The first is faster, then the second is faster. |
| Object motion | Translation along the fixed anchor camera's forward and lateral axes. | First moves toward the camera, then left |

Table 8: Geometric quantities and example answer-option forms. Each option describes one or two ordered phases; the reference answer identifies one of the four options. The examples are drawn from the stored QA records.

| Question type | How the answer is determined |
|---|---|
| **Distance**<br>3D separation between the queried entities. | **Nearly constant:** within $20\%$ of the phase's starting distance.<br>**Increasing:** each distance exceeds the preceding one by more than $20\%$.<br>**Decreasing:** each distance is more than $20\%$ below the preceding one. |
| **Speed**<br>3D displacement magnitude per interval. | **Steady pace:** within $20\%$ of the phase's starting displacement.<br>**Speeding up:** each displacement exceeds the preceding one by more than $20\%$.<br>**Slowing down:** each displacement is more than $20\%$ below the preceding one. |
| **Speed comparison**<br>Two objects over the same interval. | **Faster:** an object's displacement is more than $1.2$ times the other's.<br>**Nearly equal:** neither object meets this condition. |
| **Direction**<br>Unit direction: first to second entity. | **Front / left / above:** forward / left / up component above $0.3$.<br>**Behind / right / below:** that component is below $-0.3$.<br>Combine directions when multiple axes qualify. |
| **Moving**<br>Scene-normalized path in the anchor frame. | **Stationary:** both forward–lateral and vertical path lengths are at most $0.04$.<br>**Moving:** dominant forward, backward, left, or right displacement in each valid phase. |

Table 9: Main criteria for determining reference answers. Distance, speed, speed comparison, and direction criteria are adapted from DSR-Suite (Zhou et al., 2026) (Table 8 and implementation). Moving questions use our additional criteria for fixed-frame translation.

**Missing-entity augmentation.** We rewrite training questions to refer to objects verified by a multimodal model as absent throughout the clip. These objects have `bbox_2d` and `centroid` set to `null` at each annotated timestep; the video and other states are unchanged. Template responses select an absence option.

### A.3 CoT Generation Settings

**Teacher and inputs.** GPT-5.6 (`gpt-5.6-sol`) receives the question, four options, reference states, correct answer, and 32 uniformly sampled frames. Frames are indexed, ordered temporally, and resized to fit $384 \times 216$ pixels while preserving aspect ratio. Task-specific prompts specify coordinate conventions and link visual observations to state changes (Figure 6).

**Validation.** We regenerate outputs from the same inputs if they alter reference states or the answer, lack a nonempty `<visual>` block, or violate the response format.

## 4D-Grounded CoT Generation

```
You are generating relative distance-to-camera Chain-of-Thought training data.

You will be given a video, a per-frame `<4d>` perception, a multiple-choice
question, and the verified answer. Each target centroid is `[x,y,z]` relative to the current camera for that frame.

# Task

Produce:

<think>
<4d>{copy the given 4d perception exactly}</4d>
<visual>{direct question-relevant observation of the video}</visual>
{reasoning grounded in both}
</think>
<answer>X</answer>

## Reasoning rules

1. **Copy `<4d>` exactly.**
   Keep every frame, entity, bounding box, centroid, and camera-related field unchanged.

2. **Observe the video first.**
   After `</4d>`, inspect the full clip and write a concrete `<visual>`observation before discussing the 4D values.
   - Explicitly describe whether the target visibly approaches, recedes, or stays at a similar depth phase by phase.
     Use apparent scale, perspective, occlusion,
     scene contact, and camera motion together; apparent size alone is not sufficient.
   - Follow the full temporal sequence and describe distinct phases when they matter. Even when the answer is
     constant, explain what remains visibly
     consistent across the clip.
   - Use concrete cues from the targets, their actions, and surrounding scene. State ambiguity when camera motion,
     viewpoint, occlusion, deformation, or weak depth cues make the visual evidence uncertain.

3. **Relate the visual phases to `<4d>`.**
   After `</visual>`, keep the reasoning centered on the video.
   - For each phase, first write a sentence describing the question-relevant visible position, motion, pace, action, or
     event using concrete video cues.
   - Then check whether that same visible phase aligns with the corresponding evidence from `<4d>`, and use the 4D
     evidence to support or clarify the observation when useful.
   - Use the target centroid's 3D distance from the current camera origin at each listed frame as the corresponding
     4D evidence.
   - Prefer representative 4D evidence rather than routinely listing every value. A fuller sequence may be used when
     it is genuinely needed to distinguish
     multiple phases, establish a subtle trend, or explain an apparent mismatch.
   - Distinguish a sustained phase change from small interval-to-interval fluctuation or tracking noise.
   - If the initial visual interpretation and 4D evidence do not align, re-examine camera motion, viewpoint, occlusion,
     target deformation, and weak depth cues.
     Explain an observable reason when one exists. Do not invent visual evidence to force agreement; if they remain
     inconsistent, state the mismatch.
   - Select the available option that best matches the combined visual and 4D evidence and explain why it is the
     closest choice.

Do not mention private labels, thresholds, dataset construction, the verified answer, or an internal target.
Do not use the words `vector` or `vectors`; use `3D offset`, `direction`, or `trajectory` instead.

# Output format

<think>
<4d>[{"frame": 0, "entities": [...]}, ...]</4d>
<visual>Direct phase-by-phase video observation.</visual>
Reasoning grounded in the visible behavior and corresponding 4D evidence.
</think>
<answer>X</answer>

Return only the requested item blocks with no Markdown fences or extra text.
```

Figure 6: Teacher prompt for generating 4D-grounded CoT supervision, shown for relative distance-to-camera questions.

**4D Perception Instruction (Rel)**

Derive the 4D perception of {targets} in the video, following the camera. For each frame, give each entity's 2D bounding box and 3D centroid [x, y, z] (camera frame, z = depth from camera).
In each frame's camera coordinates, x increases to the right, y increases downward, and z increases forward along the viewing direction.

**4D Perception Instruction (Abs)**

Anchored to frame {anchor_pos} of the clip, derive the 4D perception of {targets}. For each frame, give each entity's 2D bounding box and 3D centroid [x, y, z] in that anchor camera coordinate, and the camera position as an entity labeled 'cam'.
In the anchor camera's fixed coordinates, x increases to the right, y increases downward, and z increases forward along its viewing direction.

Figure 7: Instructions for 4D perception supervision: relative (left) and absolute (right).

# B Training and Evaluation Details

## B.1 Optimization Settings

Joint SFT trains for one epoch on Trace4D-Train: 600K 4D perception examples and approximately 70K grounded QA examples, mixed from the outset.

LoRA uses rank $r = 128$, scaling factor $\alpha = 256$, and dropout $0.05$. The language model and multimodal projector use cosine learning-rate decay from $1\mathrm{e}{-4}$ to $1\mathrm{e}{-5}$ with $3\%$ warmup. SFT runs on eight H100 GPUs with an effective batch size of 128.

## B.2 Ablation Settings

Perception-first SFT uses one epoch of perception pretraining and one of QA adaptation. Adaptation uses all grounded QA examples and resamples $28\%$ of each relative and absolute perception pool per epoch for replay; this is the pool sampling rate, not the proportion in the training mixture.

All CoT variants, including Visual-only CoT, share this schedule without GRPO and differ only in QA supervision and instructions. Adaptation-only SFT runs the adaptation phase from the base VLM, whereas Joint SFT mixes perception and QA from the outset. Their budgets and data exposure differ. Table 4 also reports Joint SFT followed by GRPO.

## B.3 Geometry-Aware GRPO

GRPO trains on 7,008 QA examples for one epoch with eight responses per input, cosine learning-rate decay from $1\mathrm{e}{-5}$, $3\%$ warmup, and KL coefficient $\beta = 0.1$. In Equation 3, 4B uses $w_{\mathrm{acc}} = 0.5$ and $w_{\mathrm{pos}} = w_{\Delta} = 0.15$, whereas 8B uses $w_{\mathrm{acc}} = 0.3$ and $w_{\mathrm{pos}} = w_{\Delta} = 0.25$. Both use $w_{\mathrm{fmt}} = w_{\mathrm{bbox}} = 0.1$.

The answer-accuracy reward checks the final option against $a_{\mathrm{ref}}$. The format reward requires valid 4D JSON and nonempty temporal visual analysis and 4D-grounded reasoning:

$$R_{\mathrm{acc}}(y) = \begin{cases} 1, & \text{if the answer is correct,} \\ 0, & \text{otherwise,} \end{cases} \qquad R_{\mathrm{fmt}}(y) = \begin{cases} 1, & \text{if } y \text{ is correctly formatted,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

For non-null object boxes, the IoU reward is

$$R_{\mathrm{bbox}}(y) = \operatorname*{mean}_{i} \Big[ \operatorname*{mean}_{t} \mathrm{IoU}\Big(\hat{\mathbf{b}}_{i,t}, \bar{\mathbf{b}}_{i,t}\Big) \Big], \tag{5}$$

averaging over matched timesteps $t$ within each object track $i$, then across objects, excluding cameras. Matching uses exact entity labels and frame indices. Invalid predictions contribute zero to the corresponding geometric reward. Reference-null entries receive reward for matching absence. Displacements require valid matches at consecutive reference timesteps; entities without such intervals use position reward alone.

## B.4 Training and Inference Prompts

Figure 7 shows the relative and absolute instructions for state prediction; these examples contain no visual analysis or QA answer. Figure 8 shows the suffixes appended to questions and options during grounded QA training and inference. They specify the coordinate convention and request the full 4D-grounded CoT and answer. Reference states and answer labels are training targets, not extra student inputs. The teacher prompt used to construct CoT supervision is shown in Figure 6.

**QA Suffix (Rel)**

First, inspect the video and derive the 4D perception following the camera coordinate frame at each frame, and put it in `<4d>...</4d>`.
Then, describe the question-relevant visual evidence phase by phase in `<visual>...</visual>`.
Finally, relate the visual evidence to the corresponding 4D evidence and combine both to derive the answer.

**QA Suffix (Abs)**

First, inspect the video and derive the 4D perception in the fixed anchor-camera frame used for the question, and put it in `<4d>...</4d>`.
Then, describe the question-relevant visual evidence phase by phase in `<visual>...</visual>`.
Finally, relate the visual evidence to the corresponding 4D evidence and combine both to derive the answer.

Figure 8: QA suffixes for grounded QA training and inference: relative (left) and absolute (right).

### B.5 Evaluation Protocol

**QA evaluation.** Trace4D uses 32 sampled video frames and greedy decoding with a maximum of 4,096 generated tokens. We extract the option letter from the final `<answer>` block; responses without a valid answer are counted as incorrect. Trace4D-Bench contains 1,200 questions over 1,178 distinct temporal clips, with 600 questions in each coordinate setting. VLM4D contains 1,816 questions over 1,000 videos (600 real and 400 synthetic). Overall accuracy is computed over all questions in each benchmark.

**Perception metrics.** We evaluate predicted states from 600 Relative and 600 Absolute spatial QA responses at exact matching frames (Tables 5 and 10). IoU excludes supplied initial boxes; centroid and trajectory L2 measure position and consecutive-matched-frame displacement errors. We average within each valid object track, then across tracks, excluding cameras. For Relative, all reference coordinates are divided by the largest positive depth of the referenced objects at the first clip frame, shared across objects and timesteps. Absolute references are already normalized and evaluated as stored. Predictions are unchanged; no alignment is fitted.

## C GRPO Ablation

Table 10 compares different reward configurations and their effects on QA accuracy and 4D state prediction.

| **Training / feedback** | **QA accuracy (%)** | | **4D state prediction** | | |
|---|---|---|---|---|---|
| | **Overall** | **VLM4D** | **Centroid L2** ↓ | **Trajectory L2** ↓ | **IoU** ↑ |
| Joint SFT | 77.83 | 62.39 | 0.1873 | 0.1178 | 0.7353 |
| + GRPO: accuracy only | 78.33 | 62.67 | 0.1831 | 0.1164 | 0.7362 |
| + GRPO: position only | 78.83 | 63.11 | 0.1805 | 0.1151 | **0.7403** |
| **Trace4D (Joint SFT + GRPO)** | **79.25** | **64.32** | **0.1775** | **0.1123** | 0.7377 |

Table 10: QA accuracy and 4D state prediction in the GRPO ablation. State metrics use all 1,200 Trace4D-Bench questions, with valid-track averaging.

**Settings.** All variants use the same 7,008 QA examples and optimization settings (Appendix B.3). Accuracy-only sets $w_{\text{acc}} = 0.9$ and $w_{\text{fmt}} = 0.1$, with all geometric reward weights set to zero. Position-only and Trace4D share $w_{\text{acc}} = 0.5$ and $w_{\text{fmt}} = w_{\text{bbox}} = 0.1$. Position-only uses $w_{\text{pos}} = 0.3$ and $w_{\Delta} = 0$, while Trace4D uses $w_{\text{pos}} = w_{\Delta} = 0.15$.

**State evaluation.** We score 4D states in saved QA responses, averaging first within and then across object tracks. Cameras and initial-frame boxes are excluded. Models use their own matched timesteps.

**Results.** Trace4D outperforms Accuracy-only by 0.92 and 1.65 percentage points on Trace4D-Bench and VLM4D, respectively. Relative to Position-only, Trace4D also improves QA accuracy and reduces both centroid and trajectory L2 errors, though with slightly lower IoU. These results indicate that incorporating geometric rewards can improve QA accuracy while reducing coordinate and trajectory errors in the predicted 4D states.

## D QUALITATIVE EXAMPLES

Figures 9, 10, and 11 show Trace4D predictions on relative QA, absolute QA, and VLM4D.

**Input clip**



Q. Following the camera, how does the distance between the dog and the camera change?

A. It moves closer to the camera, then stays about the same distance.
B. It stays about the same distance, then moves farther from the camera.
C. It moves farther from the camera, then stays about the same distance.
D. It stays about the same distance, then moves closer to the camera.

<think>
<4d>[{"frame": 0, "entities": [{"label": "dog", "bbox_2d": [458, 700, 645, 978], "centroid": [0.032, 0.207, 1.0]}]}, ... {"frame": 31, "entities": [{"label": "dog", "bbox_2d": [509, 333, 536, 403], "centroid": [0.039, -0.14, 2.117]}]}]</4d>
<visual>At first, the dog remains beside the seated dog and the handler at roughly the same depth on the path, with only small posture and lateral changes. It then turns and walks away along the path, becoming progressively smaller and more distant relative to the handler and fence. The camera follows slightly, but the dog still recedes clearly through the middle and final portions of the clip.</visual>
During the initial phase, the dog's visible depth is nearly constant; its camera distance is about 1.02 at frame 0, 1.02 at frame 4, and 1.02 at frame 9, while the modest variation through frame 13 is not a sustained approach or recession. Once the dog turns away, its visible shrinkage and movement deeper along the path align with a sustained increase in 3D distance: approximately 1.30 at frame 18, 1.60 at frame 22, 1.81 at frame 27, and 2.12 at frame 31. Thus, it stays about the same distance initially and then moves farther from the camera, matching option B.
</think>
<answer>B</answer>

**Input clip**



Q. Following the camera, how does the direction from the camera to the guinea pig with initial bounding box [56,305,267,825] change?

A. It is to the front-right, then to the front-left.
B. It is to the front-left, then to the front.
C. It is to the back, then to the back-right.
D. It is to the front, then to the back-right.

<think>
<4d>[{"frame": 0, "entities": [{"label": "guinea pig", "bbox_2d": [56, 306, 267, 825], "centroid": [-0.351, 0.038, 1.0]}]}, ... {"frame": 31, "entities": [{"label": "guinea pig", "bbox_2d": [375, 192, 684, 425], "centroid": [0.019, -0.121, 1.061]}]}]</4d>
<visual>At the beginning, the black-and-white guinea pig is visibly on the left side of the overhead view, slightly below the middle, while remaining in front of the camera. It then moves steadily rightward across the bedding toward the center and the food bowl. In the later phase it reaches the bowl and stays near the center-right, with only a slight rightward fluctuation. Its vertical placement changes somewhat as it moves, but the important sustained change is from left of center to nearly straight ahead.</visual>
In the initial phase, the guinea pig is visibly left of the camera's forward center. This agrees with the negative horizontal offsets at frames 0–13, such as x = -0.351 to -0.309, while z remains positive near 1.0, placing it in front.
During the middle phase, it visibly crosses toward the center. The 4D evidence shows the horizontal offset shrinking from x = -0.181 at frame 18 to x = -0.006 at frame 22, while z remains positive at about 1.06.
In the final phase, it remains near the center rather than moving distinctly to the right. Frames 27 and 31 have only small positive x offsets of 0.028 and 0.019, compared with the much larger initial left offset, while z stays positive near 1.06. Thus the direction changes from front-left to approximately front, making B the closest choice.
</think>
<answer>B</answer>

Figure 9: Trace4D predictions for camera-relative distance (top) and direction (bottom). Each example connects temporal visual observations with 4D state changes to support the answer. The displayed `<4d>` blocks are abbreviated.

**Input clip**



Q. Anchored to frame 27 of the clip, compare the speed between the white dog and the labrador with initial bounding box [300,672,437,911].

A. The second is faster, then they move at a similar speed.
B. They move at a similar speed, then the second is faster.
C. The first is faster, then the second is faster.
D. The second is faster, then the first is faster.

<think>
<4d>[{"frame": 0, "entities": [{"label": "white dog", "bbox_2d": [481, 564, 572, 919], "centroid": [0.014, 0.139, 0.998]}, ... , {"label": "cam", "centroid": [-0.001, -0.001, -0.001]}]}</4d>
<visual>From the beginning through frame 27, the white dog remains seated on the right and the labrador remains on the left, with only small posture and head adjustments; neither makes sustained scene motion, so their visible paces are similar. The path, handler, and background remain nearly fixed, indicating little camera motion. ... , but the approach is clear.</visual>
In the first phase, both dogs visibly hold their places while making only minor adjustments. This agrees with the fixed-coordinate displacements: over frames 0–4, 4–9, 9–13, 13–18, and 18–22, the white dog moves about 0.000, 0.000, 0.000, 0.000, and 0.000, while the labrador moves about 0.012, 0.001, 0.000, 0.002, and 0.005. The labrador's small early fluctuations are consistent with posture or tracking variation rather than sustained travel, so the two are best characterized as moving at a similar speed.
...
After frame 27, the labrador visibly advances toward the camera while the white dog remains seated. The 4D evidence strongly supports this change: from frames 27–31, the white dog moves about 0.000, whereas the labrador moves about 0.118, dominated by a depth change from 0.979 to 0.869. The camera centroid changes only slightly, so this late difference is not explained by camera motion. Thus the closest choice is that they initially move at a similar speed, then the second dog becomes faster.
</think>
<answer>B</answer>

**Input clip**



Q. Anchored to frame 0 of the clip, how does the distance between the plastic wrap with initial bounding box [362,11,640,475] and the anchor camera change?

A. It moves farther from the camera, then stays about the same distance.
B. It stays about the same distance, then moves closer to the camera.
C. It moves closer to the camera, then stays about the same distance.
D. It moves farther from the camera, then moves closer to the camera.

<think>
<4d>[{"frame": 0, "entities": [{"label": "plastic wrap", "bbox_2d": [362, 11, 641, 475], "centroid": [0.001, -0.077, 1.0]}, ... ,{"label": "cam", "centroid": [0.001, 0.001, -0.001]}]}</4d>
<visual>At the beginning, the person holds the plastic-wrapped item close to the lens, where it occupies a large portion of the image ... and position change only slightly, so its distance from the anchor camera appears roughly constant. The viewpoint itself remains essentially fixed, with no meaningful camera movement.</visual>
During the initial lowering phase, the visible motion away from the lens agrees with the anchor-relative centroid distance increasing from about 1.00 at frame 0 to about 1.38 at frame 4. Once the wrap is on the counter, it remains visibly stationary through most of the clip, matching the nearly unchanged centroids from frames 4 through 23, whose distance is about 1.38. The later bounding-box shift is not a sustained depth change: the centroid remains near [0.149, 0.104, 1.353] at frames 27 and 31, still about 1.37 from the fixed origin. Thus the best description is that it moves farther away and then stays about the same distance.
</think>
<answer>A</answer>

Figure 10: Trace4D predictions for absolute speed comparison (top) and distance to the anchor camera (bottom). Both examples use the fixed anchor-camera frame specified in the question. Ellipses indicate omitted portions of the responses.

**Input clip**



Q. Which direction did the robot arm move?

A. not moving
B. left
C. no robot arm there
D. right

<think>
<4d>[{"frame": 0, "entities": [{"label": "robot arm", "bbox_2d": [334, 433, 572, 861], "centroid": [-0.04392, 0.090311, 0.999999]}, ... , {"frame": 31, "entities": [{"label": "robot arm", "bbox_2d": [756, 667, 998, 867], "centroid": [0.374002, 0.130001, 0.950511]}, {"label": "cam", "bbox_2d": null, "centroid": [-0.000111, -0.000111, -0.000121]}]}]</4d>
<visual>The robot arm begins near the center of the view and steadily travels toward the right, with its base and articulated arm shifting rightward across the floor. It then continues rightward while rotating its articulated arm, eventually extending toward the right edge and becoming partly cropped. The background and floor remain nearly fixed, indicating little camera movement.</visual>
At the beginning, the arm is visibly left of center; correspondingly, its 4D x position is -0.04392. During the middle phase, it advances steadily rightward, matching the x increase through 0.099002, 0.18003, and 0.262002. In the final phase, the arm reaches the right edge and is partly cropped; its x position reaches 0.375002 and remains near 0.374002 at the end. The camera centroid changes only negligibly, so this sustained increase in the arm's x position reflects its own motion rather than a viewpoint shift. Thus, the best match is right.
</think>
<answer>D</answer>

**Input clip**



Q. Is the swan swimming forwards or backwards from its own perspective?

A. both ways
B. backwards
C. not moving
D. forwards

<think>
<4d>[{"frame": 0, "entities": [{"label": "swan", "bbox_2d": [173, 153, 598, 794], "centroid": [-0.076, 0.048, 0.981]}, ... {"frame": 31, "entities": [{"label": "swan", "bbox_2d": [158, 181, 603, 872], "centroid": [0.223, 0.011, 1.094]}, {"label": "cam", "centroid": [0.001, -0.002, 0.001]}]}]</4d>
<visual>The black swan glides steadily to the right across the pond throughout the clip. Its body and raised neck shift rightward relative to the concrete bank and vegetation, while its tail and webbed feet leave a gentle wake behind it. The viewpoint remains nearly fixed, with only slight framing drift, so the swan's rightward progression is directly visible rather than being caused by a camera pan.</visual>
Early in the clip, the swan visibly advances rightward with its body and neck moving in that direction; correspondingly, its fixed-frame x position increases from -0.076 at frame 0 to 0.009 at frame 8. Through the middle, it continues gliding right without reversing, and the 4D x position rises further to 0.081 at frame 16 and 0.151 at frame 23. By the end, the swan remains oriented and moving right, with its x position reaching 0.223 at frame 31. The camera centroid changes only minimally, so this sustained increase belongs to the swan's motion. Thus, from its own perspective, the swan is swimming forwards.
</think>
<answer>D</answer>

Figure 11: Trace4D predictions on VLM4D: robot-arm movement direction (top) and swan motion from its own perspective (bottom). The displayed <4d> blocks are abbreviated.