4DThinker: THINKING WITH 4D IMAGERY FOR DYNAMIC SPATIAL UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic spatial reasoning from monocular video is essential for bridging visual intelligence and the physical world, yet remains challenging for vision-language models (VLMs). Prior approaches either verbalize spatial-temporal reasoning entirely as text, which is inherently verbose and imprecise for complex dynamics, or rely on external geometric modules that increase inference complexity without fostering intrinsic model capability. In this paper, we present *4DThinker*, the first framework that enables VLMs to "think with 4D" through *dynamic latent mental imagery*, i.e., internally simulating how scenes evolve within the continuous hidden space. Specifically, we first introduce a *scalable, annotation-free* data generation pipeline that synthesizes 4D reasoning data from raw videos. We then propose Dynamic-Imagery Fine-Tuning (**DIFT**), which jointly supervises textual tokens and 4D latents to ground the model in dynamic visual semantics. Building on this, 4D Reinforcement Learning (**4DRL**) further tackles complex reasoning tasks via outcome-based rewards, restricting policy gradients to text tokens to ensure stable optimization. Extensive experiments across multiple dynamic spatial reasoning benchmarks demonstrate that *4DThinker* consistently outperforms strong baselines and offers a new perspective toward 4D reasoning in VLMs.

1 INTRODUCTION

The physical world is inherently dynamic. For an intelligent agent to truly understand the environment, it must go beyond static perception and reason about *how things change in 3D space over time*. Dynamic spatial reasoning, the ability to decompose and interpret the interplay of camera ego-motion and object motion from monocular video, is therefore a cornerstone of real-world visual intelligence, with direct implications for autonomous driving, AR/VR, and robotics Zhang et al. (2025).

Despite rapid advances in vision-language models (VLMs), recent benchmarks expose that even the strongest models fail at basic dynamic reasoning Zhou et al. (2025c); Zhang et al. (2025). Existing efforts to close this gap broadly follow two directions. One constructs 4D post-training data that verbalizes spatial-temporal reasoning entirely as text Zhou & Lee (2025); Zhu et al. (2026); Huang et al. (2026), yet natural language is *inherently verbose and struggles to precisely convey complex dynamics* Yu et al. (2026). The other augments the model with external modules, e.g., injecting geometric priors via 3D foundation models Zhou et al. (2025a) or appending mask decoders for spatial grounding Zhou et al. (2025b), but *at the cost of increased inference complexity and non-intrinsic model capability*. A promising alternative is *latent reasoning* Yu et al. (2026), which encodes reasoning cues in continuous hidden space rather than explicit tokens. However, existing latent methods are limited to static scenes and depend on annotated reference images or distilled foundation models for supervision, *hindering their scalability to the dynamic, annotation-scarce video domain.*

These limitations motivate three core desiderata: **(D1) Imagery-Dynamic**: extend latent visual reasoning beyond static scenes to *capture 4D spatial-temporal evolution*; **(D2) Model-Intrinsic**: embed reasoning capabilities directly *within the model*, obviating the need for external geometric modules; **(D3) Data-Scalable**: scale the training paradigm *without relying on manual annotations*.

We take inspiration from how humans naturally reason about motion. When observing a dynamic scene, we naturally parse motion by mentally simulating salient landmarks, i.e., anchoring on static cues to infer ego-motion, and tracking trajectories to understand object dynamics. *4DThinker*

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.

operationalizes this insight by highlighting salient landmarks via mask overlays and *treating these highlighted frames as "imagery" that the model learns to simulate within its latent space.*

Specifically, *4DThinker* introduces a "think with 4D" framework with three components. *First*, we design a *scalable, annotation-free data generation pipeline* that synthesizes 4D reasoning data directly from raw videos (**D3**). The pipeline decomposes dynamic understanding along camera-motion and object-motion axes, generating motion-centric QA pairs with chain-of-thought that interleaves textual analysis and dynamic mental imagery. *Second*, we propose *Dynamic-Imagery Fine-Tuning* (DIFT), a supervised training that grounds the model's intrinsic 4D latents in dynamic visual semantics (**D1, D2**). DIFT jointly optimizes a cross-entropy loss on text tokens and a cosine-similarity loss on latent positions, teaching the model to *internally simulate* dynamics over time. *Third*, we introduce *4D Reinforcement Learning* (4DRL), a modified GRPO training that addresses challenging motions via outcome-based rewards (**D1**). The policy gradient is restricted to text tokens only, excluding latent positions where continuous hidden-state propagation is misaligned with discrete log-probabilities.

Our contributions can be summarized as follows.

- We propose *4DThinker*, the first "think with 4D" framework that equips VLMs with the capacity to *mentally simulate 4D dynamics*, enabling intrinsic reasoning about camera and object motion without any external geometric modules.
- We introduce a *scalable, annotation-free* pipeline that synthesizes 4D reasoning data from raw videos, featuring Chain-of-Thought (CoT) interleaved with dynamic mental imagery.
- We design a two-stage training recipe: *DIFT* jointly supervises text and dynamic imagery for reasoning warm-up, while *4DRL* selectively optimizes text tokens only, further refining compound-motion reasoning through outcome-based rewards.
- Extensive experiments across multiple benchmarks demonstrate that *4DThinker* consistently outperforms strong baselines, validating its effectiveness in dynamic spatial reasoning.

2 Related Work

2.1 Latent Reasoning

Chain-of-thought (CoT) prompting Wei et al. (2022) has proven effective at eliciting multi-step reasoning from large language models (LLMs) by verbalizing intermediate steps. However, verbal reasoning is inherently redundant in tokens and *struggles to precisely convey complex spatial-temporal cues (e.g., 3D layouts, dynamic trajectories) Yu et al. (2026).* This motivates *latent reasoning*, which shifts part of the reasoning from the explicit token space into the model's continuous hidden space.

Early explorations introduced dedicated tokens to structure the latent computation. Pause-pretraining Goyal et al. (2023) inserts learnable `<pause>` tokens that grant extra computation steps before committing to output. Implicit CoT Deng et al. (2024) distills explicit CoT traces into implicit hidden-state trajectories, internalizing reasoning without explicit token generation. COCONUT Hao et al. (2024) goes further by replacing CoT tokens entirely with continuous latent embeddings, showing that multi-hop reasoning paths can be effectively encoded on a continuous manifold.

Moving beyond text-only models, recent work has explored latent reasoning in multimodal settings. Mirage Yang et al. (2025c) introduces *machine mental imagery*, interleaving compact latent visual tokens with text by recasting hidden states as visual embeddings. LVR Li et al. (2025a) similarly employs latent visual tokens but performs intrinsic iterative refinement without auxiliary image supervision. Most recently, 3DThinker Chen et al. (2025) extends this paradigm to 3D, aligning latent tokens with a 3D foundation model to enable geometric imagination during spatial reasoning.

Despite these advances, existing methods remain confined to pure text, 2D images, or static scenes. *4DThinker* is *the first to extend latent visual tokens to spatial-temporal dynamics*, enabling the model to internally simulate object trajectories, camera motion, and their interplay in video.

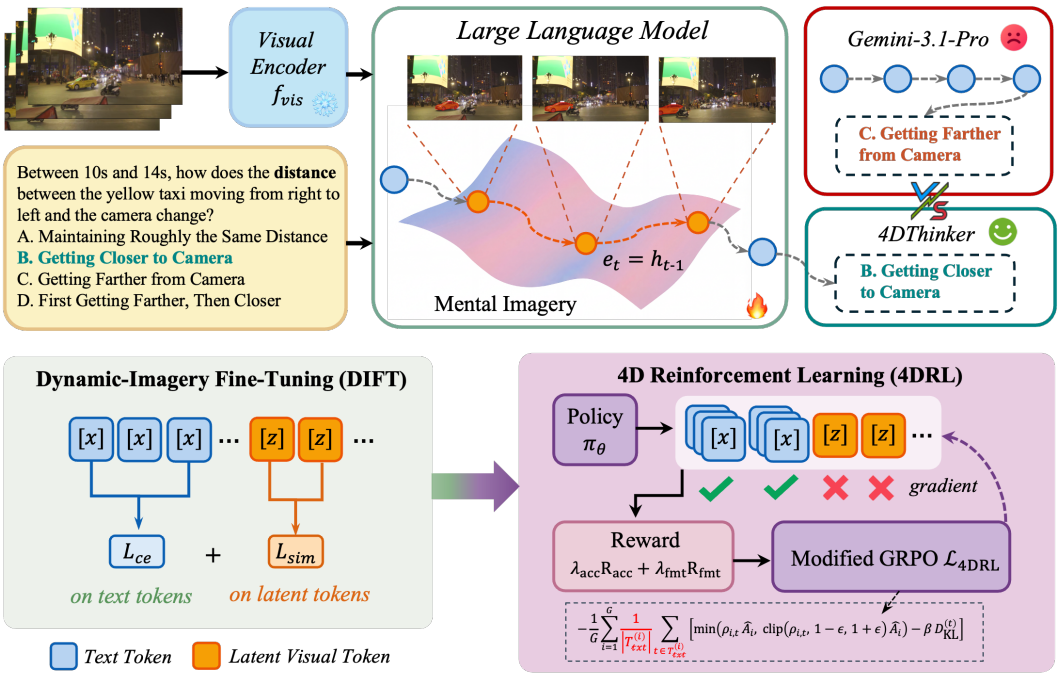


Figure 1: **Overview of *4DThinker*. Top:** Inference architecture. The model interleaves text reasoning with *latent visual tokens* as "mental imagery" on a continuous manifold, enabling correct dynamic reasoning where purely textual CoT (e.g., Gemini-3.1-Pro) fails. **Bottom:** Two-stage training pipeline built on the data from Fig. 2. DIFT (left) jointly supervises text tokens via $\mathcal{L}_{ce}$ and latent tokens via $\mathcal{L}_{sim}$; 4DRL (right) then applies modified GRPO with gradients on text tokens only, excluding latent positions to avoid noise from continuous-discrete mismatch.

2.2 Visual-Spatial Understanding

Spatial understanding has received increasing attention as a core capability for VLMs interacting with the 3D world Chen et al. (2024); Cai et al. (2024). On the benchmark side, a series of works Tong et al. (2024); Yang et al. (2025a); Ma et al. (2024) systematically probe spatial competencies such as distance estimation and relative positioning. On the modeling side, one line of methods augment inputs with explicit geometric signals, i.e., depth maps Liu et al. (2025a), or point clouds Fan et al. (2025), to supply 3D priors directly. Another enhances intrinsic reasoning without external geometry, e.g., MindCube Yin et al. (2025) constructs textual cognitive maps while 3DThinker Chen et al. (2025) generates latent 3D tokens for geometric imagination. Despite notable progress, these efforts remain largely confined to *static* scenes, with dynamic spatial reasoning in video still underexplored.

Extending spatial reasoning to dynamic scenes from monocular video, where both the camera and objects may move, poses a harder and more practically relevant challenge. VLM4D Zhou et al. (2025c) first highlights this gap with a benchmark showing that even strong VLMs fail at basic motion direction reasoning. DSI-Bench Zhang et al. (2025) further decouples camera and object motion, revealing that VLMs systematically conflate the two. More recently, DSR Suite Zhou et al. (2025a) provides a large-scale dataset alongside a Geometry Selection Module (GSM) that injects geometric priors for dynamic spatial reasoning. VideoLoom Zhou et al. (2025b) introduces SlowFast token designs that decouple temporal context from spatial detail for joint spatial-temporal understanding.

However, existing methods for dynamic spatial understanding all rely on external geometric modules that *increase inference complexity*. *4DThinker* enables the model to *internally simulate* through latent visual tokens, *without additional modules or priors.* That is, *4DThinker* develops an intrinsic capacity for 4D reasoning, *arriving at answers through "mental imagery" of the dynamic scene.*

3 METHODOLOGY

Understanding dynamic scenes from monocular video requires reasoning about how objects and the camera move through 3D space over time. Inspired by the cognitive mechanism of mental imagery, we propose *4DThinker*, a framework that enables VLMs to *internally visualize spatial-temporal dynamics during reasoning via latent visual tokens, without relying on any external geometric modules.* As illustrated in Fig. 1, *4DThinker* consists of three key components: (1) a *scalable, annotation-free* data generation pipeline that synthesizes 4D reasoning data from raw videos (Sec. 3.1); (2) *Dynamic-Imagery Fine-Tuning* (DIFT), which grounds 4D latents in dynamic visual semantics through joint supervision (Sec. 3.2); and (3) *4D Reinforcement Learning* (4DRL), which further refines reasoning on complex compound motions via outcome-based rewards (Sec. 3.2).

3.1 SCALABLE 4D DATA GENERATION

Manual annotation for spatial-temporal understanding data is *expensive* and inherently *unscalable*. On the other hand, our method requires reformulating conventional QA data into CoT reasoning *grounded in dynamic mental imagery*. To bridge this gap, we propose a *scalable, annotation-free* pipeline to synthesize 4D reasoning data from raw videos. This pipeline sequentially executes video preprocessing, motion-centric QA construction, and imagery-based CoT synthesis as shown in Fig. 2.

Video preprocessing. Our training corpus is built from SpatialVID Wang et al. (2025a), a large-scale video collection. We use only its videos and geometric annotations estimated automatically by MegaSaM Li et al. (2025d), introducing *no information that requires human annotation*.

Since dynamic reasoning relies on salient objects to gauge relative motion, our first step is to identify these landmarks and extract their masks. Given a video $\mathcal{V}$, we uniformly sample frames to obtain $\{I_t\}_{t=0}^{T-1}$, where T is the video duration in seconds. Based on predefined rules $\mathcal{R}$ (see Appendix D), we query a high-level model M_{high} (e.g., Gemini3-pro) to identify a representative *static* object o^s (e.g., the red building) and a *dynamic* object o^d (e.g., the person riding the blue bike) that persist throughout the video. A promptable video segmentation model (SAM3 Carion et al. (2025)) then tracks each object across all frames, producing temporally consistent binary mask sequences $\{M_t^s\}$ and $\{M_t^d\}$. Leveraging these masks, we generate the *mask overlays* to highlight the target object:

$$\hat{I}_t = (1 - \alpha \cdot M_t) \odot I_t \; + \; \alpha \cdot M_t \odot \mathbf{c}, \tag{1}$$

where $\alpha \in (0, 1]$ is the opacity, $\mathbf{c} \in \mathbb{R}^3$ is the highlight color, and $\odot$ denotes element-wise multiplication. To prevent identity drift, we also apply a *consistency filter*. Specifically, we prompt M_{high} to cross-verify the entire set, retaining only temporally consistent overlays:

$$\mathcal{T}_{\text{valid}} = \big\{\, t \;\big|\; \Phi_{M_{\text{high}}}\big(\hat{I}_t,\ \{\hat{I}_{t'}\}_{t' \neq t}\big) = \texttt{True} \,\big\}. \tag{2}$$

This yields a set of *valid overlays* $\{\hat{I}_t\}_{t \in \mathcal{T}_{\text{valid}}}$ together with their corresponding *valid frames* $\{I_t\}_{t \in \mathcal{T}_{\text{valid}}}$, which serve as the foundation for all subsequent data construction.

After preprocessing, we decouple dynamic understanding for monocular videos into *camera motion* and *object motion*, and structure our data generation along these two axes.

Camera motion data. From the camera trajectories produced by MegaSaM, SpatialVID derives per-segment camera motion labels L_c covering 12 canonical movement types. We first partition the video timeline into temporally contiguous segments based on these labels, yielding a sequence of labeled intervals $\{[t_a^i,\, t_b^i]\}_{i=1}^{M}$. For a given segment $[t_a, t_b]$ within this sequence, we leverage its associated motion label in conjunction with the valid images $\{I_t\}_{t \in \mathcal{T}_{\text{valid}}}$ to prompt M_{high}, formulating the camera motion Multiple-Choice Question (MCQ), denoted as (Q^s, A^s).

To establish the corresponding visual imagery, for each labeled segment $[t_a,\, t_b]$, we extract the *static mask overlays* at the boundary frames from the valid set $\{\hat{I}_t^s\}_{t \in \mathcal{T}_{\text{valid}}}$ (Eq. equation 2), yielding $\hat{I}_{t_a}^s$ and $\hat{I}_{t_b}^s$. The key insight here is that for a static object ($\Delta \mathbf{p}_t^{\text{obj}} = \mathbf{0}$), its apparent displacement in the image plane is entirely attributable to camera motion:

$$\bar{\mathbf{p}}_{t_b}^{\,s} - \bar{\mathbf{p}}_{t_a}^{\,s} \;=\; \Delta \mathbf{p}_{[t_a,\, t_b]}^{\text{cam}}, \tag{3}$$

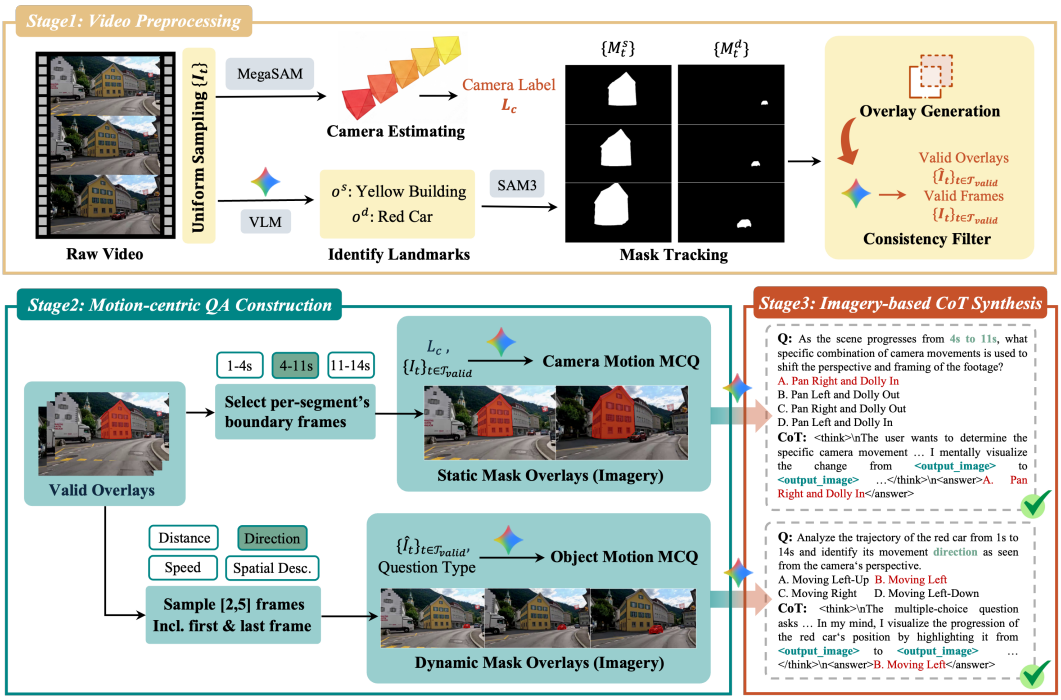


Figure 2: Overview of our **scalable, annotation-free** 4D data generation pipeline in three stages. **(1) Video preprocessing:** raw videos are processed via MegaSaM and SAM3 to extract camera trajectories and consistent mask overlays for landmarks. **(2) Motion-centric QA construction:** the pipeline formulates MCQs and imagery for both camera and object motions, grounded by sampled boundary or interval overlays. **(3) Imagery-based CoT synthesis:** M_{high} generates "think with 4D" data that interleaves text and dynamic mental imagery, culminating in the final training sample.

where $\bar{\mathbf{p}}_t^s$ denotes the centroid of M_t^s in image coordinates. Consequently, these boundary overlays serve as explicit visual evidence of camera movement.

Ultimately, the MCQ and imagery components are aggregated, culminating in the complete sample s^s: $\big(Q^s,\ A^s,\ \{I_t\}_{t\in\mathcal{T}_{\text{valid}}},\ \{\hat{I}_{t_a}^s, \hat{I}_{t_b}^s\}\big)$. Additional details are provided in the Appendix E.

Object motion data. For the dynamic object o^d, we formulate candidate question types encompassing direction, distance, speed, and spatial descriptions grounded by bounding boxes (derived from masks). To deduce the ground-truth motion attributes, we prompt M_{high} with the valid overlays of dynamic object $\{\hat{I}_t^d\}_{t\in\mathcal{T}_{\text{valid}}}$ (Eq. equation 2) to analyze the trajectory (see Appendix E), explicitly accounting for both in-plane displacements and apparent scale variations. By integrating these trajectory analyses with the predefined question types, we construct the object motion MCQ, denoted as (Q^d, A^d).

Concurrently, to establish the visual imagery, we generate the *dynamic mask overlays* $\{\hat{I}_{t_i}^d\}_{i=1}^N$ by sampling $N \in [2, 5]$ frames from the valid overlays. To capture the complete motion extent, this sampling process mandates the inclusion of the first and last frames of the object's active interval. The final object motion sample s^d is thus formulated: $\big(Q^d,\ A^d,\ \{I_t\}_{t\in\mathcal{T}_{\text{valid}}},\ \{\hat{I}_{t_i}^d\}_{i=1}^N\big)$.

Imagery-based CoT synthesis. We argue that *discriminating complex motion inherently requires mentally visualizing the temporal dynamics of the attended object.* To emulate this, we synthesize structured CoT reasoning that interleaves textual analysis with *dynamic mental imagery*. Given the previously formulated s^s or s^d, M_{high} is prompted to produce a "think with 4D" reasoning trace r. Specifically, each CoT r adheres to a structured format: `<think>...<imagery>..<imagery>...</think><answer>...</answer>`. An automated validator subsequently verifies the placeholder count, chronological consistency, and answer isolation; non-compliant samples are either regenerated or discarded. Additional details are in the Appendix I.

Training data composition. Executing the proposed pipeline yields ∼38K pairs tailored for supervised training. Each sample encapsulates CoT with mental imagery, formally defined as:

$$\mathcal{S} = \Big(Q,\ A,\ \{I_t\}_{t\in\mathcal{T}_{\text{valid}}},\ \{\hat{I}_{t_i}\},\ \ r \Big). \tag{4}$$

While the supervised corpus warms up reasoning on single-category motions, we introduce an RL stage using DSR-Train (∼37K samples) Zhang et al. (2025) to master complex, compound motions. Lacking explicit reasoning traces, this QA-only dataset compels the model to autonomously explore reasoning paths, guided solely by outcome-based rewards.

3.2 Learning to Think with 4D

Building upon the dataset generated in Sec. 3.1, we now describe how *4DThinker* learns to internalize dynamic imagery as part of its reasoning process. That is, we represent mental imagery as *latent visual tokens*, the compact continuous embeddings that reside within the hidden space of the language model. We first formalize this representation, then present a two-stage training framework: dynamic-imagery fine-tuning (DIFT), followed by 4D reinforcement learning (4DRL).

Latent visual token representation. Let f_{vis} denote the visual encoder of the base VLM. For an overlay image (i.e., imagery) $\hat{I}_{t_i}$ (Eq. equation 1), the encoder produces a patch-level embedding sequence $\mathbf{E}_{t_i} = f_{\text{vis}}(\hat{I}_{t_i}) \in \mathbb{R}^{L\times D}$, where L is the number of visual tokens and D is the hidden dimension. We compress this sequence into K *latent visual tokens* via partitioned mean pooling:

$$\mathbf{z}_{t_i}^{(k)} = \frac{1}{|\mathcal{P}_k|} \sum_{j\in\mathcal{P}_k} \mathbf{E}_{t_i}[j], \quad k = 1, \ldots, K, \tag{5}$$

where $\{\mathcal{P}_k\}_{k=1}^{K}$ is an equal partition of $\{1, \ldots, L\}$. Each `<imagery>` placeholder (i.e., special token) in the CoT r is then replaced by a *latent block*:

$$\texttt{<lat_s>}\ \mathbf{z}_{t_i}^{(1)}\ \mathbf{z}_{t_i}^{(2)}\ \cdots\ \mathbf{z}_{t_i}^{(K)}\ \texttt{<lat_e>}, \tag{6}$$

where `<lat_s>` and `<lat_e>` serve as learnable delimiter tokens. Consequently, the training sequence interleaves discrete text tokens with continuous latent blocks, enabling the model to *reason through dynamic imagery* without leaving the autoregressive generation loop.

Dynamic-imagery fine-tuning (DIFT). Given the sample $\mathcal{S}$ (Eq. equation 4), we form the input by encoding video frames $\{I_t\}$ as visual tokens, appending the question Q, and substituting each imagery placeholder in r with its corresponding latent block. The visual encoder f_{vis} is kept frozen throughout this stage, providing a stable target embedding space. We optimize a dual-objective loss:

$$\mathcal{L}_{\text{DIFT}} = \lambda_{\text{ce}}\,\mathcal{L}_{\text{ce}} \ +\ \lambda_{\text{sim}}\,\mathcal{L}_{\text{sim}}. \tag{7}$$

The first term is the standard causal language modeling loss restricted to text token positions $\mathcal{T}_{\text{txt}}$:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{|\mathcal{T}_{\text{txt}}|} \sum_{t\in\mathcal{T}_{\text{txt}}} \log\, p_\theta(x_{t+1} \mid x_{\le t})\,. \tag{8}$$

The second term introduces a *next-embedding prediction* objective at latent positions. Let $\mathcal{T}_{\text{lat}}$ denote the set of all latent token positions and $\mathbf{h}_t$ the hidden state at position t. Adhering to the autoregressive paradigm, $\mathbf{h}_{t-1}$ serves as the predictive representation for position t; we enforce its alignment with the ground-truth visual embedding $\mathbf{z}_t$ via cosine similarity:

$$\mathcal{L}_{\text{sim}} = 1 - \frac{1}{|\mathcal{T}_{\text{lat}}|} \sum_{t\in\mathcal{T}_{\text{lat}}} \frac{\mathbf{h}_{t-1}^{\top}\,\mathbf{z}_t}{\|\mathbf{h}_{t-1}\|\ \|\mathbf{z}_t\|}. \tag{9}$$

Essentially, this objective imparts 4D patterns through continuous supervision, i.e., the model learns to *internally simulate the visual dynamics of attended objects at each imagery step.*

During inference, the DIFT formulation naturally gives rise to a *recurrent* mental imagery mechanism that operates in a purely self-conditioned manner. This is achieved by directly feeding the preceding hidden state as the input embedding whenever the current position falls within a latent block:

$$\mathbf{e}_t = \begin{cases} \text{Embed}(x_t), & t \notin \mathcal{T}_{\text{lat}}, \\ \mathbf{h}_{t-1}, & t \in \mathcal{T}_{\text{lat}}, \end{cases} \tag{10}$$

where $\mathrm{Embed}(\cdot)$ denotes the standard discrete token embedding lookup. This establishes a recurrent loop: *the model's own "imagination" at one imagery step feeds forward as context for subsequent reasoning, allowing it to mentally track how objects move in 3D space over time.*

4D reinforcement learning (4DRL). Although DIFT equips the model with the ability to reason via dynamic imagery, the supervised signal is limited to single-category motion, leaving its understanding of complex 4D scenes somewhat constrained. To overcome this limitation, we further apply a modified version of GRPO Shao et al. (2024) utilizing the QA-only dataset introduced in Sec. 3.1.

For a given question, the policy π_θ samples a group of G candidate responses $\{y_i\}_{i=1}^{G}$. Each response is evaluated using a composite reward function:

$$R(y_i) = \lambda_{\text{acc}}\, R_{\text{acc}}(y_i) + \lambda_{\text{fmt}}\, R_{\text{fmt}}(y_i), \tag{11}$$

where $R_{\text{acc}}, R_{\text{fmt}} \in \{0, 1\}$ reward answer correctness and the "think with 4D" format, respectively. The group-normalized advantages are then computed as follows:

$$\hat{A}_i = \frac{R(y_i) - \mu_G}{\sigma_G}, \quad \mu_G = \tfrac{1}{G}\textstyle\sum_j R(y_j), \;\; \sigma_G = \sqrt{\tfrac{1}{G}\textstyle\sum_j (R(y_j) - \mu_G)^2}. \tag{12}$$

The policy is optimized via a clipped surrogate objective, regularized by the KL divergence against the frozen DIFT reference policy π_{ref}. A key modification over standard GRPO is that we restrict the policy gradient to the index set $\mathcal{T}_{\text{txt}}^{(i)} = \{1, \ldots, |y_i|\} \setminus \mathcal{T}_{\text{lat}}^{(i)}$, which *explicitly excludes all latent token positions*. This is to *avoid destabilizing gradient noise caused by the mismatch between continuous latent propagation (Eq. equation 10) and discrete log-probabilities.* The resulting 4DRL objective is:

$$\mathcal{L}_{\text{4DRL}} = -\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|\mathcal{T}_{\text{txt}}^{(i)}|} \sum_{t \in \mathcal{T}_{\text{txt}}^{(i)}} \left[\min\Big(\rho_{i,t}\, \hat{A}_i,\; \text{clip}\left(\rho_{i,t},\, 1{-}\epsilon,\, 1{+}\epsilon\right)\hat{A}_i \Big) - \beta\, D_{\text{KL}}^{(t)} \right], \tag{13}$$

where $\rho_{i,t} = \frac{\pi_\theta(x_t | x_{<t})}{\pi_{\text{ref}}(x_t | x_{<t})}$ and $D_{\text{KL}}^{(t)}$ are the per-token importance ratio and KL divergence, respectively.

4 Experiments

Experimental setup. We follow the two-stage training pipeline described in Sec. 3.2. Implementation details are provided in Appendix K, training data composition and evaluation benchmarks in Appendix G, and subtask definitions in Appendix H. All benchmarks are formatted as multiple-choice questions; we report accuracy via exact match, with "Avg." denoting the mean across all subtasks.

Baselines. We compare with three groups of models: (1) *proprietary VLMs*: GPT-5 Singh et al. (2025) and Gemini-2.5-Pro Comanici et al. (2025); (2) *spatial understanding models*: VLM-3R Fan et al. (2025), VG-LLM Zheng et al. (2025), DSR Suite-Model Zhou et al. (2025a), SpaceR-7B Ouyang et al. (2025), VST-7B-RL Yang et al. (2025b), Spatial-SSRL-7B Liu et al. (2025b), and SpatialLadder-3B Li et al. (2025b); and (3) *base VLMs* on which *4DThinker* is applied: Qwen2.5-VL-3B/7B Bai et al. (2025), Qwen3-VL-8B/32B Team (2025), and InternVL3.5-8B/38B Wang et al. (2025b).

4.1 Benchmarking Fine-Grained 4D Reasoning

DSR-Bench Zhou et al. (2025a) targets *quantitative geometric measurement* in dynamic scenes, requiring models to produce procedural answers that precisely characterize how spatial attributes (e.g., distance, orientation, speed) evolve over time.

As shown in Tab. 1, *4DThinker* delivers consistent improvements across all base VLMs. For Qwen2.5-VL-3B, DIFT yields **+6.5 pp** (31.1 vs. 24.6), and the full *4DThinker* pipeline (DIFT+4DRL) achieves **+9.6 pp** (34.2 vs. 24.6). The gains become more pronounced at larger scales, i.e., Qwen3-VL-32B improves by **+34.0 pp** (**62.0** vs. 28.0), *surpassing both the proprietary Gemini-2.5-Pro (31.7) and the best task-specific DSR Suite-Model (58.9).* Cross-architecture generalization is also evident, with InternVL3.5-38B gaining **+32.7 pp** (59.4 vs. 26.7).

Notably, the gains are most pronounced on absolute subtasks (e.g., A.Dis, A.Ori) where base models near chance ($\sim$20%), indicating that *4D latents supply the geometric grounding that pure language*

Table 1: **Fine-grained 4D reasoning evaluation on DSR-Bench.** Top three performers in each column are highlighted from Dark to Light, and overall Avg. rankings from Dark to Light. For 4DThinker, *+DIFT* adds supervised training on top of the base model; *+DIFT+4DRL* further adds RL training. Gains (↑) are relative to the base model.

Models	A.Dis	A.Dir	A.Ori	A.Spd	A.SpdC	A.DirP	R.Dis	R.Dir	R.Ori	R.Spd	R.SpdC	R.DirP	N-Temp	Avg.
Proprietary Models														
GPT-5	21.1	41.5	48.7	34.5	33.3	34.7	17.2	44.3	41.9	21.2	25.0	30.9	26.7	**30.8**
Gemini-2.5-Pro	20.0	44.6	53.6	27.3	38.7	30.5	23.2	32.9	43.2	17.1	28.5	27.9	34.3	**31.7**
Spatial Understanding Models														
VLM-3R	28.2	27.6	31.7	42.8	38.7	33.0	34.4	23.8	30.8	22.2	26.4	29.1	35.0	**31.4**
VG-LLM	55.2	32.3	58.5	57.1	51.6	32.2	56.0	36.3	32.0	30.3	32.1	29.1	27.9	**38.4**
DSR Suite-Model	87.0	73.8	84.1	73.8	72.0	35.5	75.8	76.1	77.7	60.6	37.1	35.1	46.4	**58.9**
Different Base Models (4DThinker)														
Qwen2.5-VL-3B	20.5	20.3	20.3	37.0	17.6	22.6	25.5	20.7	25.6	23.7	29.0	24.2	27.0	**24.6**
+*DIFT*	48.2	28.1	25.3	46.9	28.6	28.7	35.8	29.9	37.2	24.7	25.3	31.1	26.6	**31.1**$^{\uparrow 6.5}$
+*DIFT+4DRL*	50.6	34.4	29.1	56.8	34.1	34.8	49.1	33.3	30.8	27.8	26.8	28.6	27.5	**34.2**$^{\uparrow 9.6}$
Qwen2.5-VL-7B	18.8	15.3	14.6	42.8	29.0	19.4	31.8	19.3	11.1	22.2	19.2	20.2	30.1	**23.5**
+*DIFT*	45.8	35.9	34.2	50.6	37.4	29.6	40.6	37.9	35.9	30.9	19.6	27.3	38.6	**34.8**$^{\uparrow 11.3}$
+*DIFT+4DRL*	71.1	62.5	60.8	70.4	49.5	39.1	60.4	47.1	43.6	38.1	32.6	32.3	42.5	**47.1**$^{\uparrow 23.6}$
Qwen3-VL-8B	23.5	24.6	42.6	29.7	27.9	33.8	18.1	28.4	34.5	24.2	22.1	27.9	33.5	**28.7**
+*DIFT*	51.8	45.3	55.7	44.4	37.4	42.6	50.0	47.1	44.9	42.3	35.5	32.9	40.8	**42.6**$^{\uparrow 13.9}$
+*DIFT+4DRL*	78.3	65.6	82.3	59.3	58.2	46.1	59.4	63.2	61.5	58.8	39.1	36.6	47.2	**54.6**$^{\uparrow 25.9}$
Qwen3-VL-32B	14.5	37.5	24.1	43.2	27.5	38.3	13.2	32.2	38.5	17.5	20.3	28.0	31.8	**28.0**
+*DIFT*	53.0	65.6	40.5	45.7	56.0	41.7	50.9	48.3	46.2	43.3	37.7	34.8	44.2	**45.2**$^{\uparrow 17.2}$
+*DIFT+4DRL*	84.3	79.7	82.3	84.0	67.0	47.8	67.0	82.8	79.5	62.9	44.2	41.0	48.5	**62.0**$^{\uparrow 34.0}$
InternVL3.5-8B	23.5	27.6	28.0	34.5	24.7	27.9	22.4	17.0	19.7	28.2	30.0	14.2	30.1	**25.4**
+*DIFT*	43.4	42.2	36.7	37.0	33.0	30.4	41.5	39.1	37.2	35.1	30.4	28.0	41.2	**36.2**$^{\uparrow 10.8}$
+*DIFT+4DRL*	69.9	64.1	65.8	53.1	51.6	39.1	52.8	55.2	52.6	47.4	39.1	36.6	49.8	**50.0**$^{\uparrow 24.6}$
InternVL3.5-38B	25.8	27.8	29.2	34.2	24.7	28.5	26.7	16.3	23.4	29.2	32.1	15.4	31.3	**26.7**
+*DIFT*	59.0	53.1	49.4	40.7	41.8	32.2	51.9	41.4	38.5	44.3	39.9	29.8	44.2	**42.5**$^{\uparrow 15.8}$
+*DIFT+4DRL*	85.5	73.4	83.5	80.2	60.4	37.4	67.9	74.7	70.5	64.9	44.9	37.3	49.8	**59.4**$^{\uparrow 32.7}$

reasoning lacks. Moreover, our best model surpasses previous state-of-the-art (SOTA) DSR Suite-Model (**62.0** vs. 58.9) without external 3D priors (e.g., Geometry Selection Module (GSM)), showing that *internalized 4D imagery can be more effective than modular geometric injection.*

4.2 Comparison on Holistic Dynamic Understanding

Beyond geometric precision, Dyn-Bench Huang et al. (2026) evaluates *semantic-level* dynamic understanding along three axes: *inter-object*, *object-scene*, and *camera-object*, testing whether models can reason about interactions, trajectories, and causal relationships in 4D environments.

Tab. 2 shows that *4DThinker* yields strong and consistent gains, *with the top three performers on the leaderboard all produced by our method.* In terms of Qwen2.5-VL-7B, DIFT+4DRL achieves **+11.6 pp** (65.9 vs. 54.3), already outperforming GPT-5 (61.4) and Gemini-2.5-Pro (58.8), while Qwen3-VL-32B reaches **75.4** (**+10.9 pp**), establishing a new SOTA. Even with the smaller Qwen2.5-VL-7B backbone, our model surpasses all dedicated spatial understanding baselines, including SpaceR-7B (59.2), VST-7B-RL (58.8), and SpatialLadder-3B (56.2), by a substantial margin.

The strong performance on Dyn-Bench reveals that the 4D latents learned by *4DThinker* encode not only low-level geometric trajectories (as validated of DSR-Bench on Tab. 1), but also *higher-level motion semantics*. Notably, improvements on the camera-object axis, *which demands disentangling ego-motion from object dynamics, confirm that 4DThinker acquires a genuine internal 4D representation rather than relying on single-viewpoint heuristics.*

4.3 Ablation Studies

We further ablate key design choices of *4DThinker* using Qwen2.5-VL-3B on DSR-Bench.

Training strategy. As shown in Tab. 3, under supervised training, Raw QA SFT (25.1) and CoT SFT (26.8) barely exceed the base model, while DIFT reaches **31.1** (**+5.2 pp**) by jointly supervising latent visual tokens and textual reasoning *in a unified "think with 4D" paradigm.* Under reinforced training,

Table 2: **Holistic dynamic understanding evaluation on Dyn-Bench**. Top three performers in each column are highlighted.

Methods	Inter-Obj	Obj-Scene	Cam-Obj	Avg.
Proprietary Models				
GPT-5 Singh et al. (2025)	54.7	70.2	59.2	**61.4**
Gemini-2.5-Pro Comanici et al. (2025)	56.1	64.4	55.8	**58.8**
Spatial Understanding Models				
SpaceR-7B Ouyang et al. (2025)	56.2	72.7	48.6	**59.2**
VST-7B-RL Yang et al. (2025b)	56.3	74.4	45.7	**58.8**
Spatial-SSRL-7B Liu et al. (2025b)	47.5	69.4	36.7	**51.2**
SpatialLadder-3B Li et al. (2025b)	52.0	72.7	44.0	**56.2**
Different Base Models (4DThinker)				
Qwen2.5-VL-7B Bai et al. (2025)	50.8	69.9	42.1	**54.3**
+*DIFT*	56.3	74.4	47.8	**59.5**$^{\uparrow 5.2}$
+*DIFT+4DRL*	63.5	79.2	54.9	**65.9**$^{\uparrow 11.6}$
Qwen3-VL-8B Team (2025)	59.0	76.2	56.0	**63.7**
+*DIFT*	64.1	79.0	60.8	**68.0**$^{\uparrow 4.3}$
+*DIFT+4DRL*	70.9	84.5	68.0	**74.5**$^{\uparrow 10.8}$
Qwen3-VL-32B Team (2025)	61.1	76.0	56.5	**64.5**
+*DIFT*	66.3	79.8	61.5	**69.2**$^{\uparrow 4.7}$
+*DIFT+4DRL*	73.6	84.0	68.7	**75.4**$^{\uparrow 10.9}$
InternVL3.5-8B Wang et al. (2025b)	50.8	65.3	42.7	**52.9**
+*DIFT*	56.3	69.8	48.2	**58.1**$^{\uparrow 5.2}$
+*DIFT+4DRL*	63.1	75.5	54.9	**64.5**$^{\uparrow 11.6}$
InternVL3.5-38B Wang et al. (2025b)	49.4	64.9	44.3	**52.9**
+*DIFT*	55.4	70.1	50.3	**58.6**$^{\uparrow 5.7}$
+*DIFT+4DRL*	62.9	76.6	57.9	**65.8**$^{\uparrow 12.9}$

Table 3: Comparison of different VLM training strategies.

Training strategies	Accuracy
Raw QA SFT	25.1
CoT SFT	26.8
DIFT	**31.1**
GRPO	27.1
CoT SFT + GRPO	29.7
DIFT + 4DRL	**34.2**

Table 4: Ablation on different loss and reward components.

Loss & reward components	Accuracy
Full	**34.2**
w/o $\mathcal{L}_{\text{sim}}$	28.5
w/o $\mathcal{L}_{\text{ce}}$	19.3
w/o R_{fmt}	33.4
w/o R_{acc}	32.0

Table 5: Effect of different latent token sizes for DIFT training.

Size	1	2	4	8	16
Accuracy	29.6	30.4	**31.1**	30.8	29.3

vanilla GRPO (27.1) and CoT SFT + GRPO (29.7) *remain limited by the expressiveness bottleneck of discrete text*, whereas DIFT + 4DRL achieves **34.2**, confirming that 4D latent representations offer a richer optimization landscape than discrete textual CoT.

Loss and reward components. Tab. 4 reveals the relative importance of each component. Removing $\mathcal{L}_{\text{ce}}$ causes catastrophic degradation (34.2 $\rightarrow$ 19.3), as the model *loses the ability to generate coherent text*. Removing $\mathcal{L}_{\text{sim}}$ yields the next largest drop (34.2 $\rightarrow$ 28.5), confirming that visual alignment supervision is essential, i.e., *4D latents degenerate into ungrounded representations without it.* For RL rewards, R_{acc} (34.2 $\rightarrow$ 32.0) contributes more than R_{fmt} (34.2 $\rightarrow$ 33.4), *as accuracy reward directly shapes reasoning quality* while format reward mainly ensures structural compliance.

Latent token size. Tab. 5 studies the capacity-performance trade-off. Accuracy increases from K=1 (29.6) to K=4 (31.1), as *additional tokens encode richer spatial-temporal information for imagery.* Beyond K=4, performance slightly declines (30.8 at K=8, 29.3 at K=16). We attribute this to *excessive latent tokens diluting the textual context*, disrupting the model's language coherence.

5 Conclusion and Limitation

In this paper, we propose the "think with 4D" framework *4DThinker*, which enables VLMs to reason about dynamic scenes through latent visual imagery. To this end, we integrate a scalable and annotation-free data generation pipeline, joint text-imagery supervision fine-tuning(DIFT), and outcome-based 4D reinforcement learning (4DRL) into a unified training recipe. Consistent improvements across multiple benchmarks confirm that grounding chain-of-thought in continuous 4D imagery is more effective than purely textual or module-augmented approaches.

Limitation & Future Work. While *4DThinker* shows consistent gains on dynamic spatial reasoning, further studying how latent 4D imagery scales with longer contexts and richer multi-agent interactions remains interesting future work.

AI USE STATEMENT

In this work, we used generative AI tools to polish manuscript wording after an author-written draft, and to assist with data-processing and ablation-experiment code. All AI-assisted text and code were manually verified by the authors. We take responsibility for the final content of this work.

ETHICS STATEMENT

This study strictly adheres to the ICLR Code of Ethics. The datasets utilized in our experiments are publicly available, fully anonymized, and do not involve human subjects, privacy infringement, or harmful discrimination concerns.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide full implementation details, hyperparameter configurations, and training scripts in the anonymized code repository submitted as supplementary material.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *arXiv preprint arXiv:2406.13642*, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Boyu Chang, Qi Wang, Xi Guo, Zhixiong Nan, Yazhou Yao, and Tianfei Zhou. Abductivemllm: Boosting visual abductive reasoning within mllms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2698–2706, 2026.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Yan Feng, Peng Pei, Xunliang Cai, and Ruqi Huang. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. *arXiv preprint arXiv:2510.18632*, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. *arXiv preprint arXiv:2505.20279*, 2025.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Yuzhi Huang, Kairun Wen, Rongxin Gao, Dongxuan Liu, Yibin Lou, Jie Wu, Jing Xu, Jian Zhang, Zheng Yang, Yunlong Lin, et al. Thinking in dynamics: How multimodal large language models perceive, track, and reason dynamics in physical 4d world. *arXiv preprint arXiv:2603.12746*, 2026.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025a.

Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. Spatialladder: Progressive training for spatial reasoning in vision-language models. *arXiv preprint arXiv:2510.08531*, 2025b.

Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. Sti-bench: Are mllms ready for precise spatial-temporal world understanding? *arXiv preprint arXiv:2503.23765*, 2025c.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10486–10496, 2025d.

Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanguang Zhang, Yaochen Hu, Lingfeng Zhang, Yingxue Zhang, Shuang Wu, Tongtong Cao, Guowei Huang, et al. Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning. *arXiv preprint arXiv:2501.10074*, 2025a.

Yuhong Liu, Beichen Zhang, Yuhang Zang, Yuhang Cao, Long Xing, Xiaoyi Dong, Haodong Duan, Dahua Lin, and Jiaqi Wang. Spatial-ssrl: Enhancing spatial understanding via self-supervised reinforcement learning. *arXiv preprint arXiv:2510.27606*, 2025b.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Celso M de Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. *arXiv preprint arXiv:2412.07825*, 2024.

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.

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

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.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 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.

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.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10632–10643, 2025a.

Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, et al. Visual spatial tuning. *arXiv preprint arXiv:2511.05491*, 2025b.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025c.

Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. *arXiv preprint arXiv:2506.21458*, 2025.

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.

Xinlei Yu, Zhangquan Chen, Yongbo He, Tianyu Fu, Cheng Yang, Chengming Xu, Yue Ma, Xiaobin Hu, Zhe Cao, Jie Xu, et al. The latent space: Foundation, evolution, mechanism, ability, and outlook. *arXiv preprint arXiv:2604.02029*, 2026.

Wanyue Zhang, Wenxiang Wu, Wang Xu, Jiaxin Luo, Helu Zhi, Yibin Huang, Shuo Ren, Zitao Liu, and Jiajun Zhang. World2vlm: Distilling world model imagination into vlms for dynamic spatial reasoning. *arXiv preprint arXiv:2604.26934*, 2026.

Ziang Zhang, Zehan Wang, Guanghao Zhang, Weilong Dai, Yan Xia, Ziang Yan, Minjie Hong, and Zhou Zhao. Dsi-bench: A benchmark for dynamic spatial intelligence. *arXiv preprint arXiv:2510.18873*, 2025.

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

Hanyu Zhou and Gim Hee Lee. Llava-4d: Embedding spatiotemporal prompt into lmms for 4d scene understanding. In *The Fourteenth International Conference on Learning Representations*, 2025.

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. *arXiv preprint arXiv:2512.20557*, 2025a.

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. *arXiv preprint arXiv:2512.20557*, 2025b.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8600–8612, 2025c.

Fangrui Zhu, Yunfeng Xi, Jianmo Ni, Mu Cai, Boqing Gong, Long Zhao, Chen Qu, Ian Miao, Yi Li, Cheng Zhong, et al. Egoreasoner: Learning egocentric 4d reasoning via task-adaptive structured thinking. *arXiv preprint arXiv:2603.06561*, 2026.

A LATENT MECHANISM ANALYSES

We conduct the following analyses with Qwen2.5-VL-3B. Matched supervised comparisons use the same 38K training samples, teacher signals, and optimization budget; the primary metric is DSR-Bench accuracy.

Table 6: **Matched controls for dynamic formulation, data versus latent mechanism, and 4DRL.** All supervised variants use Qwen2.5-VL-3B with the same 38K training samples and optimization budget. RL variants start from the same DIFT checkpoint with fixed RL data and rewards. Accuracy is reported on DSR-Bench.

(a) Dynamic visual target.

Condition	Acc.
CoT SFT	26.8
Last-frame DIFT	26.7
w/o ordered	23.9
w/o grounded mask	28.0
Full ordered DIFT	**31.1**

(b) Overlay vs. latent.

Condition	Acc.
Raw QA SFT	25.1
CoT SFT	26.8
Overlay-input SFT	27.0
DIFT	**31.1**

(c) 4DRL vs. GRPO.

Stage	Acc.
DIFT	31.1
+ vanilla GRPO	32.5
+ 4DRL w/o R_{fmt}	33.4
+ 4DRL	**34.2**

Ordered and grounded dynamic supervision. Tab. 6(a) ablates the construction of the dynamic visual target under DIFT. Last-frame DIFT aligns latent blocks only with the final overlay and provides no trajectory supervision, resembling a static latent-visual adaptation. The unordered variant retains all-timestamp overlays but randomly permutes their temporal order; the ungrounded variant restores correct order while removing landmark-mask highlighting. Last-frame DIFT does not improve over text CoT (26.7 vs. 26.8), and shuffled order reduces accuracy to 23.9, below the untrained base (24.6). Restoring correct order without grounded masks reaches 28.0, whereas the full ordered and mask-grounded formulation attains **31.1**. *The improvement therefore depends on ordered, time-indexed grounding rather than on the mere introduction of latent tokens.*

Latent mechanism versus overlay input. Tab. 6(b) isolates the recurrent latent mechanism while holding teacher-generated supervision fixed. Raw QA SFT, CoT SFT, overlay-input SFT, and DIFT share the same QA pairs; overlay-input SFT and DIFT further share the same text CoT and highlighted overlays, but only DIFT generates 4D latents. Relative to the untrained base (24.6), overlay-input SFT improves accuracy by 9.8% (27.0), whereas DIFT improves by **26.4%** (31.1). Together with the gap between removing $\mathcal{L}_{\text{sim}}$ (28.5) and the full model (34.2) in Tab. 4, *these results attribute the primary gain to latent-mechanism training rather than to the generated data alone.*

4DRL versus vanilla GRPO. Tab. 6(c) compares RL adaptations from the same DIFT checkpoint, with RL data, rewards, and all other training conditions held fixed. Vanilla GRPO improves DIFT from 31.1 to 32.5, whereas 4DRL reaches 34.2 (33.4 without R_{fmt}), a **5.2%** relative gain over directly applying GRPO (34.2 vs. 32.5). *Restricting policy gradients to text positions is therefore beneficial for the hybrid continuous–discrete trajectory induced by DIFT.*

What latent states encode. We next examine *whether latent blocks encode dynamic information and whether the model uses them during answer generation.* All analyses use a frozen Qwen2.5-VL-3B DIFT-only checkpoint. The probe-only set is constructed from SpatialVID source videos excluded from the 38K DIFT corpus and covers 12 camera-direction, 11 object-direction, 3 distance-change, and 6 speed-pattern classes, with 100 examples per class (3,200 examples in total). Within each category, source-video hashes define fixed 60/20/20 train/validation/test splits so that all examples from the same video appear in only one split.

Table 7: **Linear attribute and trajectory probes** on a frozen Qwen2.5-VL-3B DIFT-only checkpoint.

Probe target	**Latent**	**Text**	**Shuf.**
Camera dir. (12-way)	61.8	31.6	8.7
Object dir. (11-way)	66.4	35.2	9.8
Distance change (3-way)	72.9	44.8	33.9
Speed pattern (6-way)	53.7	28.0	17.4
Macro bal. acc.	**63.7**	34.9	17.5
Trajectory input	**N-ADE ↓ / N-FDE ↓**		
Latent state	**0.083 / 0.118**		
Matched text	0.142 / 0.197		
Video-shuffled	0.214 / 0.286		

Table 8: **100-way overlay retrieval** from latent blocks versus matched controls.

Query	**R@1**	**R@5**
Latent block	**38.6**	**64.2**
Matched text	8.4	21.7
Video-shuf. lat.	1.2	5.3
Random chance	1.0	5.0

Table 9: **Causal interventions** on DSR-Bench with a DIFT-only checkpoint. Only latent content is modified; video, question, interleaved text, and answer positions remain unchanged.

Setting	**Acc.**
Normal	31.1
Temporal-block shuffle	23.9 (−7.2)
Cross-video replacement	21.8 (−9.3)
Preceding latents masked	26.9 (−4.2)
Corpus-mean latents	25.7 (−5.4)
Equal text states masked	30.4 (−0.7)

For attribute probing (Tab. 7), a linear head takes as input the first latent block, the last latent block, their difference, and the mean of all blocks. Matched text controls use an equal number of adjacent text CoT states from the same model and layer; shuffled labels provide an additional baseline. Latent states achieve 63.7% macro balanced accuracy, exceeding matched text states by 28.8 points (63.7 vs. 34.9). A linear coordinate regressor predicts the mask-centroid sequence from the latent block at each overlay timestamp. *Latent states yield lower normalized average and final displacement errors (N-ADE / N-FDE* $= 0.083/0.118$*) than matched text states (*$0.142/0.197$*) or video-shuffled pairings (*$0.214/0.286$*).*

For retrieval (Tab. 8), each latent block retrieves its ground-truth overlay among 99 hard negatives drawn from incorrect videos or timestamps and matched by task type, temporal position, and mask size. Candidate embeddings are produced by the same frozen visual encoder used in DIFT. *Latent blocks attain R@1 / R@5 of 38.6 / 64.2, exceeding matched text states by 30.2 / 42.5 points.*

For causal intervention (Tab. 9), we cache normally generated latent blocks and modify only the continuous states before reinsertion in a second autoregressive pass; video, question, interleaved text, and answer positions remain unchanged. Temporal-block shuffling and cross-video replacement (matched by DSR subtask and block count) reduce DIFT accuracy by 7.2 and 9.3 points, respectively. Masking attention to the content positions of all preceding latent blocks—while leaving video,

question, interleaved text, delimiters, and answer positions intact—reduces accuracy from 31.1 to 26.9 (-4.2), close to last-frame DIFT (26.7); filling latent positions with corpus-mean embeddings yields 25.7 (-5.4). Masking an equal number of preceding text states costs only 0.7 points (30.4). *The block-to-block dependency is therefore specific to the latent trajectory.*

B Data Quality and Evaluation Protocol

Human audit of generated data. We randomly sample 100 training examples for an independent human audit. Two annotators independently judge each example as correct or incorrect along three dimensions, and a third annotator adjudicates disagreements (Tab. 10). Adjudicated accuracies are 92% for mask identity and temporal consistency, 98% for motion-label correctness, and 96% for CoT factual validity, with Cohen's κ between 0.80 and 0.86. Mask consistency is the relatively weaker dimension because of occlusion, boundary ambiguity, and cross-frame association, yet remains above 90% after adjudication. Because 4DRL is driven by outcome rewards rather than direct supervision on generated reasoning traces, it can further mitigate residual trace-label noise.

Table 10: **Human audit of automatically generated training data** (N=100). Two annotators judge each example independently; a third adjudicates disagreements.

Audit dimension	Ann. A	Ann. B	Adjudicated	κ
Mask identity and temporal consistency	93/100	91/100	92/100	0.86
Motion-label correctness	98/100	97/100	98/100	0.80
CoT factual validity	97/100	96/100	96/100	0.85

Option-free evaluation. To assess dependence on multiple-choice selection, we sample 100 DSR-Bench questions, remove all answer choices, and require free-form answers under identical inference settings (Tab. 11). Gemini-3-Pro performs binary semantic judgments by comparing generated answers with ground truth. Accuracy increases from 22.5 (base) to 34.0 (DIFT) and 38.5 (full *4DThinker*), *showing that the method also benefits open-ended answering.*

Table 11: **Option-free evaluation** on 100 DSR-Bench questions with answer choices removed. Free-form answers are scored by Gemini-3-Pro via binary semantic judgment against ground truth.

Model	Option-free Acc.
Base	22.5
CoT SFT	25.0
DIFT	34.0
Full *4DThinker*	**38.5**

C Additional Comparisons

Latent and imagery baselines. Tab. 12 compares publicly available latent/imagery and geometry-augmented methods under Qwen2.5-VL-7B. Full *4DThinker* attains 47.1 / 65.9 on DSR-Bench / Dyn-Bench without external geometric features or human-annotated imagery targets, improving over VG-LLM Zheng et al. (2025) by 22.7% on DSR-Bench (47.1 vs. 38.4) and by 9.3% on Dyn-Bench (65.9 vs. 60.3), and outperforming LVR Li et al. (2025a), 3DThinker Chen et al. (2025), and AbductiveMLLM Chang et al. (2026).

Table 12: **Comparison with latent/imagery and geometry-augmented baselines** under Qwen2.5-VL-7B. "Ext." denotes external geometric features or human-annotated imagery targets required at training or inference.

Method	Ext. feature / annotation	DSR	Dyn
Qwen2.5-VL-7B Base	None	23.5	54.3
VLM-3R Fan et al. (2025)	3D reconstruction feature	31.4	56.9
VG-LLM Zheng et al. (2025)	Geometry selection module	38.4	60.3
3DThinker Chen et al. (2025)	3D foundation-model feature	27.8	52.6
LVR Li et al. (2025a)	Annotated image	33.5	58.4
AbductiveMLLM Chang et al. (2026)	Diffusion imaginer	27.9	56.2
4DThinker DIFT-only	None	34.8	59.5
4DThinker DIFT+4DRL	None	**47.1**	**65.9**

Non-DSR benchmarks and recent 4D models. Tab. 13(a) reports results on Dyn-Bench Huang et al. (2026), VLM4D Zhou et al. (2025c), DSI-Bench Zhang et al. (2025), and STI-Bench Li et al. (2025c) with Qwen2.5-VL-7B. Full *4DThinker* improves over the base by 10.3–11.6 points on every benchmark. Under Qwen3-VL-8B (Tab. 13(b)), MLLM-4D RFT Yin et al. (2026) and World2VLM Zhang et al. (2026) reach 41.6 / 66.7 and 40.3 / 66.1 on DSR-Bench / Dyn-Bench, whereas full *4DThinker* reaches 54.6 / 74.5, corresponding to relative improvements of 31.3% and 11.7% over MLLM-4D RFT.

Table 13: **Additional evaluations beyond DSR-Bench.** (a) Non-DSR benchmarks with Qwen2.5-VL-7B. (b) Publicly available 4D-oriented models with Qwen3-VL-8B.

(a) Non-DSR benchmarks (Qwen2.5-VL-7B).

Stage	Dyn-Bench	VLM4D	DSI-Bench	STI-Bench
Base	54.3	43.8	35.7	34.5
DIFT	59.5	49.6	41.9	40.3
DIFT+4DRL	**65.9**	**54.2**	**46.8**	**44.8**

(b) 4D-oriented models (Qwen3-VL-8B).

Method	DSR-Bench	Dyn-Bench
Base Qwen3-VL-8B	28.7	63.7
MLLM-4D RFT Yin et al. (2026)	41.6	66.7
World2VLM Zhang et al. (2026)	40.3	66.1
4DThinker DIFT-only	42.6	68.0
4DThinker DIFT+4DRL	**54.6**	**74.5**

D Object Selection Rules

As described in Sec. 3.1, we define a set of predefined rules $\mathcal{R}$ to guide M_{high} in selecting a representative static object o^s and a dynamic object o^d from each video. Specifically, we instruct M_{high} with the following criteria:

Static object selection.

- The object must be *stationary* throughout the entire video (e.g., the traffic sign).
- It should be *visually salient* and occupy a reasonable area, avoiding small/occluded objects.
- The object should *persist* in most frames without prolonged absence.
- Select *central* objects to maximize camera-induced apparent displacement.

Dynamic object selection.

- The object must exhibit *clear, non-trivial* motion (e.g., the walking person wearing a red shirt).
- It should be visually *distinguishable* from the background and other moving entities.

- The object must be *trackable* across a sufficient number of frames ($\geq 50\%$ of the video duration).
- Prefer the most *prominent* moving object if multiple candidates exist.

Both selections are formatted as structured outputs containing the object name and a brief visual description (e.g., "the red car on the left lane"), which are subsequently used as text prompts for SAM3 Carion et al. (2025) to generate mask sequences.

E Prompt Details

Our data generation pipeline employs a series of carefully designed prompts at each stage. We present them below, organized by pipeline stage described in Sec. 3.1.

Video preprocessing prompts. The preprocessing stage uses three prompts. The *landmark identification* prompt (Fig. 3) with rules in Sec. D instructs M_{high} to select one static and one dynamic object from uniformly sampled frames, producing structured JSON labels for SAM3 mask extraction. The *static mask consistency verification* prompt (Fig. 4) implements the consistency filter $\Phi_{M_{\text{high}}}$ in Eq. equation 2 by evaluating object identity, mask consistency, mask quality, and visibility across all overlay frames. For dynamic objects, the *dynamic mask verification* prompt (Fig. 5) performs a more nuanced per-frame evaluation, returning *validity indices to allow partial frame acceptance.*

Landmark Identification

Analyze this video sequence and identify:
1) One object that is clearly moving/changing throughout the frames (dynamic object).
2) One object that remains static/unchanged throughout the frames (static object).

Requirements:
Both objects must be visible and prominent in ALL frames. Use SHORT and CLEAR labels (3-8 words max) with key features like color or shape, for example: 'a blue-handled knife', 'a red car', 'a person in white shirt'. If no clear dynamic or static object exists, respond with 'null' for that category.

Respond in JSON format:
{"dynamic_object": "short clear label or null", "static_object": "short clear label or null"}

Figure 3: Prompt for **landmark identification**. M_{high} identifies one static and one dynamic object with short visual descriptions, which are subsequently used as text prompts for SAM3 mask extraction.

Static Mask Consistency Verification

You are given **{num_frames}** frames extracted from a video sequence. In each frame, a region has been highlighted with a red semi-transparent overlay and a red contour outline. The intended target object is '**{obj}**'.

Please carefully evaluate ALL of the following criteria:
1. **Object Identity**: Does the red-highlighted region in EVERY frame correspond to the same '**{obj}**' object? (Not a different object or background.)
2. **Mask Consistency**: Is the mask region reasonably consistent across frames? The highlighted area should cover roughly the same portion of the object in each frame, without dramatically shifting to a completely different part of the scene.
3. **Mask Quality**: Does the mask reasonably cover the target object (not just a tiny fragment or an excessively large area that includes unrelated background)?
4. **Object Visibility**: Is the '**{obj}**' clearly visible and identifiable in all frames?

If ALL criteria are satisfied, answer 'YES'. If ANY criterion fails, answer 'NO'. Answer with ONLY 'YES' or 'NO', nothing else.

Figure 4: Prompt for **static mask consistency verification**. M_{high} evaluates four criteria across all overlay frames to implement the consistency filter (Eq. equation 2).

Camera motion QA prompts. For camera motion data, the *question generation* prompt (Fig. 6) produces natural-language MCQs subject to predefined constraints.

Object motion QA prompts. Object motion data construction involves three prompt types. First, the trajectory analysis determines ground-truth motion attributes: the *direction analysis* prompt

Dynamic Mask Consistency Verification

You are given **{num_frames}** frames extracted from a video sequence. In each frame, a region has been highlighted with a red semi-transparent overlay and a red contour outline. The intended target is a dynamic (moving) object, possibly described as '**{obj}**'.

Please carefully evaluate each frame:
1. **Is it a dynamic object?** The highlighted object must be something that is actually moving/in motion in the video (e.g., a person walking, a car driving, an animal running). If the highlighted region is a static/stationary object (like a building, tree, or fixed furniture), mark that frame as INVALID.
2. **Same object identity**: Among the frames with valid dynamic objects, do they all show the SAME moving object (same identity, not just same category)?
3. **Mask quality**: Does the mask reasonably cover the target object?

Output a JSON object with exactly this format (no other text): {"is_any_dynamic": true/false, "valid_frame_indices": [list of 0-based indices], "object_description": "brief description of the moving object"}

Figure 5: Prompt for **dynamic object mask verification**. Unlike the binary static check (Fig. 4), this prompt returns per-frame validity indices, allowing partial acceptance of frames.

Camera Motion Question Generation

You are writing multiple-choice questions about camera movements in videos.

Context: A video is given as input. We want to ask about the camera movement during the time segment from **{start_frame}**s to **{end_frame}**s.

The answer options are: **{options_str}**

Please write a single question (in English) that asks the viewer to identify the camera movement during this time segment. Use **{chosen_style}**.

Rules: - The question should naturally reference the time range (**{start_frame}**s to **{end_frame}**s). - Do NOT mention any specific objects, masks, or highlights. - Do NOT include the options in your output — just the question text itself. - Keep it to 1-2 sentences. - Output ONLY the question text, nothing else.

Figure 6: Prompt for **camera motion question generation**. Given a time segment and answer options, M_{high} produces a natural-language MCQ.

(Fig. 7) identifies the primary movement direction from 11 candidates (Tab. 15) while separating camera ego-motion, and the *speed change analysis* prompt (Fig. 8) classifies the speed pattern. Second, the *question generation* prompt (Fig. 9) covers four complementary question types: direction, bounding-box grounded description, distance change, and speed variation detailed in Tab. 14.

CoT synthesis prompts. The core "think with 4D" reasoning traces are produced by the *camera motion CoT synthesis* prompt (Fig. 10) and the *object motion CoT synthesis* prompt (Fig. 11). Both enforce the *4DThinker* reasoning flow and frame the overlay images as the model's own mental imagination via placeholders. The object motion variant additionally includes a *camera compensation* step that disentangles camera-induced apparent motion from the object's own displacement.

Training and inference instruction. Finally, the *system prompt* (Fig. 12) is appended before every question during DIFT training, 4DRL, and inference, specifying the output format that the model must follow. During 4DRL, the format reward R_{fmt} checks adherence to this think-answer structure.

Table 14: Candidate question types, target objects, and descriptions in our data generation pipeline.

Category	Type	Target Object	Description
Camera Motion	Direction	Static object	Identifying how the camera moves
Object Motion	Direction	Dynamic object	Detecting motion direction from camera view
	BBox-Grounded	Dynamic object	Identifying direction with bbox-based grounding
	Distance	Dynamic object	Determining the object's distance to the camera
	Speed	Dynamic object	Assessing how the object's speed varies over time

Object Movement Direction Analysis

You are given a video as a sequence of **{num_video}** frames (at 1 fps, so frame N = Ns). You are also given **{num_masked}** key frames (at times: **{frame_desc}**) where a dynamic (moving) object '**{obj}**' is highlighted with a red overlay and contour.

Scene context: **{scene_summary}**
Camera motion instructions (for reference, NOT the object movement): **{instructions}**

Your task: Analyze the position change of the highlighted dynamic object '**{obj}**' across the masked key frames and determine its PRIMARY movement direction **as observed from the camera's viewpoint** (i.e., how the object moves within the camera frame).

IMPORTANT: - All directions are defined **from the camera's perspective**: "Moving Left" means the object moves toward the left side of the camera frame, "Moving Up" means toward the top of the frame, etc. - "Moving Toward Camera" means the object is getting closer to the camera (appears larger over time). "Moving Away from Camera" means getting farther (appears smaller). - The camera may also be moving, so you need to separate the camera's own motion from the object's actual motion in the 3D scene. Focus on how the object's position changes **relative to the scene/world**, then express that as seen from the camera. - Choose the single best direction that describes the object's dominant movement.

Available directions: - Moving Left - Moving Right - Moving Up - Moving Down - Moving Left-Up - Moving Left-Down - Moving Right-Up - Moving Right-Down - Moving Toward Camera - Moving Away from Camera - Stationary

Output ONLY the direction name from the list above, nothing else.

Figure 7: Prompt for **object movement direction analysis**. M_{high} analyzes centroid displacement and apparent scale variation across masked key frames to determine the primary movement direction, while explicitly separating camera ego-motion from the object's own motion.

Object Speed Change Analysis

You are given a video as a sequence of **{num_video}** frames (at 1 fps, so frame N = Ns). You are also given **{num_masked}** key frames (at times: **{frame_desc}**) where a dynamic (moving) object '**{obj}**' is highlighted with a red overlay and contour.

Scene context: **{scene_summary}**

Your task: Analyze how the **speed** of the highlighted dynamic object '**{obj}**' changes across the masked key frames.

IMPORTANT: - Compare the displacement of the object between consecutive masked frames. If the displacement per unit time increases, the object is speeding up; if it decreases, it is slowing down. - Also consider the apparent size change: an object getting rapidly larger may be accelerating toward the camera. - The camera may be moving too — try to isolate the object's own speed change from camera motion. - Choose the single best description of how the speed varies.

Available speed descriptions: - Speeding Up - Slowing Down - Maintaining a Constant Speed - First Speeding Up, Then Slowing Down - First Slowing Down, Then Speeding Up - Moving at Varying Speeds

Output ONLY one speed description from the list above, nothing else.

Figure 8: Prompt for **object speed change analysis**. Complementary to the Fig. 7, M_{high} classifies the speed pattern by comparing per-frame centroid displacements and apparent size changes.

F Candidate Question Types and Answer Choices

Tab. 14 summarizes the five *candidate question types* in our data generation pipeline, and Tab. 15 lists the *corresponding answer choices*. To ensure rigorous evaluation, the camera motion MCQs are

Table 15: Answer choices for each question type.

Type	Answer Choices
Camera Direction	Move Forward, Move Backward, Move Left, Move Right, Move Up, Move Down, Pan Left, Pan Right, Tilt Up, Tilt Down, Roll Clockwise, Roll Counter-clockwise
Object Direction	Moving Left, Right, Up, Down, Left-Up, Left-Down, Right-Up, Right-Down, Toward Camera, Away from Camera, Stationary
Object Distance	Moving Toward Camera, Moving Away from Camera, Maintaining Roughly the Same Distance
Object Speed	Speeding Up, Slowing Down, Maintaining a Constant Speed, First Speeding Up Then Slowing Down, First Slowing Down Then Speeding Up, Moving at Varying Speeds

Object Motion Question Generation (4 Types)

% ---- (a) Direction Question ----
% Asks which direction the object moves in the camera frame.
You are writing a multiple-choice question about a moving object in a video.
Context: A video is given as input (1 fps). We want to ask about the movement direction of '**{obj}**' **as seen from the camera's perspective** during **{start}**s to **{end}**s.
The answer options are: **{options_str}**
Please write a single question (in English) that asks the viewer to identify the movement direction of '**{obj}**' from the camera's viewpoint. Use **{style}**.
Rules: - Make it clear the direction is as observed from the camera. - Naturally reference the time range and the object name. - Do NOT mention any masks, highlights, or overlays. - Do NOT include the options — just the question text. - Keep it to 1-2 sentences. - Output ONLY the question text, nothing else.

% ---- (b) BBox-Grounded Question ----
% References the object by its bounding box coordinates, testing spatial grounding alongside motion direction recognition.
You are writing a multiple-choice question about a moving object in a video.
Context: A video is given as input (1 fps). We want to ask about the movement direction of a dynamic object **as seen from the camera's perspective** during **{start}**s to **{end}**s.
The object can be described as: '**{obj}**'
The object's initial bounding box coordinates (in the model's coordinate space) are: **{bbox}**
The answer options are: **{options_str}**
Please write a single question (in English) that: - Asks the viewer to identify the movement direction from the camera's viewpoint - References the bounding box coordinates to specify which object - Makes clear that the direction is as seen through the camera
Rules: - Reference the time range and bbox coordinates naturally. - Keep it to 1-2 sentences. - Do NOT include the options — just the question text. - Output ONLY the question text, nothing else.

% ---- (c) Distance Question ----
% Asks how the object's distance to the camera changes, probing depth-axis understanding via apparent scale variation.
You are writing a multiple-choice question about a moving object's distance change relative to the camera in a video.
Context: A video is given as input (1 fps). We want to ask about how the distance between '**{obj}**' and the camera changes during **{start}**s to **{end}**s.
The answer options are: **{options_str}**
Please write a single question (in English) that asks the viewer to determine how the distance/proximity between the '**{obj}**' and the camera changes during this segment.
Rules: - Focus on distance/proximity change, not lateral direction. - Naturally reference the time range and object name. - Do NOT mention any masks, highlights, or overlays. - Do NOT include the options — just the question text. - Keep it to 1-2 sentences. - Output ONLY the question text, nothing else.

% ---- (d) Speed Question ----
% Asks how the object's speed varies over time, probing temporal dynamics of motion.
You are writing a multiple-choice question about how a moving object's speed varies over time in a video.
Context: A video is given as input (1 fps). We want to ask about how the speed of '**{obj}**' changes during **{start}**s to **{end}**s.
The answer options are: **{options_str}**
Please write a single question (in English) that asks the viewer to assess how the speed of the '**{obj}**' varies over the given time period.
Rules: - Focus on speed/velocity change (acceleration, deceleration, constant, varying). - Naturally reference the time range and object name. - Do NOT mention any masks, highlights, or overlays. - Do NOT include the options — just the question text. - Keep it to 1-2 sentences. - Output ONLY the question text, nothing else.

Figure 9: Prompts for **object motion question generation** (four types). Each variant probes a different aspect of dynamic understanding: (a) movement direction, (b) 4D question with bounding-box grounding, (c) distance change relative to the camera, and (d) speed variation over time.

Camera Motion CoT Synthesis

You are an advanced multimodal reasoning assistant. Your task is to generate a comprehensive, step-by-step Chain of Thought (CoT) reasoning process that leads to the correct option for a given multiple-choice question.

Input Data Provided to You: 1. **Video Frames:** The complete video as a sequence of **{num_video}** frames (from 0s to **{total_frames}**s). 2. **Masked Frames:** **{num_masked}** key frames (at **{masked_frame_desc}**) where the static object '**{obj}**' is highlighted in red overlay and red contour. 3. **Multiple Choice Question:** **{question_text}** 4. **Correct Answer:** **{correct_label}**. **{correct_answer_text}**

Your Goal: You must engineer the reasoning process. Output a thought process that logically uses the visual inputs to derive the Correct Answer.

Critical Rules for `<thinking>` and `<output_image>`:
1. **Mental Imagery as Output (NOT Input):** The Masked Frames represent the **output of your own mental imagination**. Describe a process where you *mentally visualize* highlighting the static object '**{obj}**', imagining where it appears across frames. Each `<output_image>` tag represents one such mental image.
2. **Strict Usage of `<output_image>`:** - You **must** use exactly **{num_masked}** `<output_image>` tags, strictly in chronological order. - Present these tags **all together in one sentence** to show the mental progression. - *Correct:* "...I mentally visualize the change from <output_image> to <output_image> to <output_image>..."
3. **Negative Constraint:** Do NOT use `<output_image>` again in subsequent analysis text. Refer to them using natural language like "In the first frame," "In the final view," etc.
4. **Reasoning Flow:** - **Temporal Localization:** Narrow down to the segment ({**start_frame**}s–{**end_frame**}s). - **Scene Context:** Briefly describe the target segment. - **Strategy:** Introduce the static reference object. - **Visualization:** Present the `<output_image>` sequence. - **Analysis:** Describe observed pixel displacement. - **Conclusion:** Map to the correct option.

Output Format:
<thinking> [Detailed reasoning ...] </thinking> <answer> [The Correct Option, e.g., "A. Pan Right"] </answer>

Figure 10: Prompt for **camera motion CoT synthesis**. Given the video, static mask overlays, and the correct answer, M_{high} produces reasoning trace where `<output_image>` placeholders represent the model's own "mental imagery," which are later replaced by latent visual tokens during DIFT training.

structured with four options, comprising the ground-truth label and three plausible distractors. These distractors are preferentially drawn from semantically related motion types to prevent trivial guessing. Following a parallel design, the object motion MCQs maintain the same four-option format, with distractors systematically sampled from the valid candidate set of the respective question category.

G Training and Evaluation Datasets

Training data. The DIFT stage uses ∼38K samples synthesized by our pipeline from SpatialVID Wang et al. (2025a), a large-scale video collection of 2.7M clips (7,089 hours) with MegaSaM-estimated camera poses. We use only its raw videos and geometric annotations, no human labels are involved. The 4DRL stage uses DSR-Train Zhou et al. (2025a), which provides ∼37K QA pairs covering compound camera-object motions from in-the-wild videos, without explicit reasoning traces.

Benchmarks. We evaluate on benchmarks spanning dynamic spatial reasoning. **DSR-Bench** Zhou et al. (2025a) covers 13 fine-grained subtasks including absolute/relative direction, distance, speed, and orientation. **Dyn-Bench** Huang et al. (2026) evaluates perception, tracking, and reasoning of dynamic content in 4D scenes with 1K videos and 7K VQA pairs. We do not use its mask data.

H Benchmark Subtask Descriptions

We provide detailed descriptions of the subtask abbreviations used in Tab. 1 and Tab. 2.

DSR-Bench subtasks. DSR-Bench Zhou et al. (2025a) organizes its 13 subtasks along two axes: *viewpoint mobility* (Absolute vs. Relative) and *spatial attribute type*. "Absolute" (A.) denotes that the viewpoint is fixed at a specific timestamp, while "Relative" (R.) denotes that the viewpoint moves with the observing agent over time. The attribute types are:

- **Dis** (Distance): How the distance between objects changes over time.
- **Dir** (Direction): The movement direction of a target object.
- **Ori** (Orientation): How the orientation of an object evolves.
- **Spd** (Speed): The speed change pattern of a target object.
- **SpdC** (Speed Comparison): Comparing the speeds of two objects.
- **DirP** (Direction Prediction): Predicting the future movement direction.

The 13th subtask, **N-Temp** (Non-Template Based), consists of free-form questions auto-generated by a language model to probe more general spatial-temporal understanding beyond fixed templates.

Object Motion CoT Synthesis

You are an advanced multimodal reasoning assistant. Your task is to generate a comprehensive, step-by-step Chain of Thought (CoT) reasoning process that leads to the correct option for a given multiple-choice question about a dynamic object's movement (as observed from the camera's viewpoint).

Input Data Provided to You: 1. **Video Frames:** The complete video as a sequence of **{num_video}** frames (from 0s to **{total_frames}**s, at 1 fps). 2. **Masked Frames:** **{num_masked}** key frames (at **{masked_frame_desc}**) where the dynamic object '**{obj}**' is highlighted in red overlay and red contour. 3. **Multiple Choice Question:** **{question_text}** 4. **Correct Answer:** **{correct_label}**. **{correct_direction}**

Your Goal: Output a thought process that logically derives the Correct Answer.

Critical Rules for `<thinking>` and `<output_image>`:
1. **Mental Imagery as Output (NOT Input):** The Masked Frames represent the **output of your own mental imagination**. Describe mentally visualizing the dynamic object '**{obj}**' highlighted across frames. Each `<output_image>` is one such mental image.
2. **Strict Usage of `<output_image>`:** - Exactly **{num_masked}** tags, in chronological order. - All grouped together in one sentence. - *Correct:* "...I mentally visualize tracking the object from <output_image> to <output_image> to <output_image>..."
3. **Negative Constraint:** Do NOT reuse `<output_image>` in subsequent analysis. Use natural language references instead.
4. **Reasoning Flow:** - **Scene Overview:** Describe the video scene and identify the dynamic object. - **Temporal Localization:** Identify the target time segment. - **Strategy:** State the need to track the object's position. - **Visualization:** Present the `<output_image>` sequence. - **Position Analysis:** Describe position change and/or apparent size change (text only, no tags). - **Camera Compensation:** If the camera is also moving, separate camera motion from object motion. - **Conclusion:** Determine movement direction and match to the correct option.

Output Format:
<thinking> [Detailed reasoning ...] </thinking> <answer> [The Correct Option, e.g., "A. Moving Left"] </answer>

Figure 11: Prompt for **object motion CoT synthesis**. Analogous to the camera motion variant (Fig. 10), the model tracks the dynamic object's position across frames.

Dyn-Bench subtasks. Dyn-Bench Huang et al. (2026) structures evaluation along three complementary semantic axes:

Inference / Training System Instruction

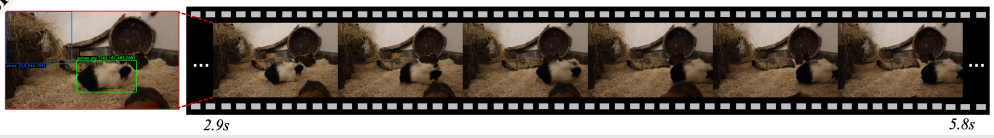
First, think about the reasoning process with mental imagery of the relevant object or region, then provide the user with the answer. Put the reasoning in <think>...</think> and only the final answer in <answer>...</answer>, e.g. <think> reasoning process (with mental imagery as needed) </think><answer> answer here </answer>.

Figure 12: The **system instruction** appended before every question during DIFT training, 4DRL training, and inference. It specifies the output format that the model must follow.

- **Inter-Object**: How multiple dynamic objects interact with each other, including spatial relations, approach/separation, occlusion, and action descriptions.
- **Object-Scene**: How individual objects move within their environment, covering movement patterns, trajectories, and scene-level dynamics.
- **Camera-Object**: How camera motion affects the perceived geometry and temporal consistency of dynamic objects, including camera motion orientation, camera-object interaction, as well as the temporal visual changes.

DSR-Bench

Visual Input



Query

Between 2.9s and 5.8s, following the perspective of camera, how does the distance between guinea pig with initial bounding box coordinates [262,181,483,298] and straw with initial bounding box coordinates [0,0,244,193] change?
A. (1) Becomes smaller. (2) Becomes larger.
B. (1) Becomes larger.
C. (1) Becomes smaller.
D. (1) Becomes larger. (2) Keep nearly constant.

4DThinker-Qwen2.5-VL-3B

<think>\nThe question asks for an analysis of how the distance between the 'guinea pig' (initially identified by the bounding box [262,181,483,298]) and the 'straw' (initially identified by the bounding box [0,0,244,193]) changes during the interval from 2.9s to 5.8s.\n\nLooking at the scene, the straw is located on the left side of the frame. The camera is positioned low, looking slightly up at the enclosure's interior. To determine the distance change, I need to track the position of the black-and-white guinea pig relative to the straw over the specified time period.\n\nIn my mind, I imagine highlighting the black-and-white 'guinea pig' with a red overlay and contour across the key frames to track its movement. I mentally visualize this progression: from <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|>, where the 'guinea pig' is highlighted in each imagined frame.\n\nAnalyzing this mental sequence, I observe two distinct phases of movement. First, starting from 2.9s, the black-and-white guinea pig moves towards the right side of the enclosure. Since the straw is positioned on the far left, this movement to the right causes the distance between the guinea pig and the straw to become larger.\n\nNext, as the video progresses towards 5.8s, the guinea pig reaches the right side of the frame and then stops. It stays relatively still for the remainder of the interval. Because the guinea pig is no longer moving, its distance to the straw keeps nearly constant during this latter part of the sequence.\n\nTherefore, the distance first becomes larger, and then keeps nearly constant. \n\nThis observation aligns perfectly with Option D. \n</think>\n<answer>**D. (1) Becomes larger. (2) Keep nearly constant.**</answer>

Gemini 3

Distance Change Analysis\nThe distance between the guinea pig and the straw evolves in two distinct phases within this timeframe:\n\n1. **Phase 1 (approx. 2.9s to 4.5s):** The guinea pig moves toward the straw pile, making the distance **smaller**.\n2. **Phase 2 (approx. 4.5s to 5.8s):** The guinea pig turns and moves away from the straw, causing the distance to become **larger**.\n\n---\n\n### Conclusion\nThe correct option is **A**:\n* **(1) Becomes smaller.**\n* **(2) Becomes larger.**

Qwen2.5-VL-3B

Answer: **A. (1) Becomes smaller. (2) Becomes larger.**

Figure 13: **Qualitative example on DSR-Bench (fine-grained).** *4DThinker* correctly identifies a two-phase pattern (first becomes larger, then keeps constant) by mentally simulating the guinea pig's trajectory via latent 4D imagery. Both Gemini-3 and the base Qwen2.5-VL-3B fail.

I AUTOMATED COT VALIDATION

To ensure data quality, we apply a rule-based validator mentioned in Sec. 3.1 that checks:

Dyn-Bench

Visual Input



Query

What is the overall pattern of the main player's movement across the court?
A. He moves in a straight line from one end of the court to the other.
B. He moves back and forth diagonally, covering the entire court area.
C. He stays mostly in one half of the court, shifting slightly side to side.
D. He runs in circles around the center of the court.

4DThinker-Qwen2.5-VL-3B

<think>\nThe video depicts an outdoor tennis court where a man is playing. The question asks for the overall movement pattern of the \"main player\" across the court.\n\nTo analyze his movement, I need to track his position at various points throughout the video. I will focus on his location relative to the court's boundaries and his change in depth within the scene.\n\nIn my mind, I mentally visualize highlighting the main player with a red overlay to track his progression through the game. I picture this sequence: from <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|>, where the 'the main player' is highlighted in each imagined frame.\n\nAnalyzing this mental sequence, I observe that in the first imagined frame, the man is positioned on the left side of the court. As the sequence progresses through the middle frames, he moves towards the right side of the screen while simultaneously moving further away from the camera, heading toward the net. By the final imagined frames, he has reached the far right side of the court and is beginning to move toward the left again.\n\nWhile there are minor lateral shifts as he plays the shot, the most prominent movement is a continuous traversal across the width of the court from the left side toward the right side. He does not stay in one specific area but moves across the court's length.\n\nTherefore, the overall pattern of the main player's movement across the court is a diagonal path from one end to the other.\n\nThis corresponds to Option B.\n</think>\n<answer>**B. He moves back and forth diagonally, covering the entire court area.**</answer>

Gemini 3

Based on the sequence of images provided, the player's movement is confined to a specific area of the tennis court rather than covering its full length or width.\n\nAnalysis of Movement\n* Initial Positioning: The player begins on the green surface (the "out" area or baseline region), positioned behind the blue court area.\n* Lateral Shifting: Across the images, he moves horizontally (side to side) to adjust his position for incoming shots.\n* Depth Control: While he steps forward toward the blue court line to strike the ball (as seen in the final frames), he remains primarily on one side of the net.\n* Consistency: He does not cross to the opposite end of the court, nor does he move in wide diagonal patterns or circles.\n\nConclusion\nThe correct choice is **C. He stays mostly in one half of the court, shifting slightly side to side**.

Qwen2.5-VL-3B

C. He stays mostly in one half of the court, shifting slightly side to side.

Figure 14: **Qualitative example on Dyn-Bench (holistic).** *4DThinker* correctly identifies the player's diagonal movement pattern across the full court by mentally tracking his position through 4D latents, while both Gemini-3 and the base Qwen2.5-VL-3B incorrectly conclude that the player stays in one half of the court, relying on local frame-level heuristics.

Visual Input



Query

Over time, does the main object visually appear larger or smaller? (Considering combined effects of zoom and physical movement)
A. Larger: The proportion of the screen occupied by the object significantly increases.
B. Smaller: The proportion of the screen occupied by the object significantly decreases.
C. Unchanged: The size ratio remains stable.
D. Fluctuating: The size undergoes repeated changes.

4DThinker-Qwen2.5-VL-3B

<think>\nThe video presents a scene featuring a panda bear in its enclosure, situated next to a water trough. To determine if the main object (the panda) appears larger or smaller over time, I need to observe its scale relative to the camera frame across several key moments.\n\nMy strategy involves tracking the panda's size from the beginning to the end of the specified interval. I will mentally visualize highlighting the panda in red at each stage to precisely gauge any changes in its apparent dimensions on screen.\n\nIn my mind, I picture the panda highlighted across the key moments, forming a progression from <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|>, where the object is clearly marked in each imagined frame.\n\nAnalyzing this mental sequence, I notice that the panda is the central focus of the shot. In the first few frames, the panda is seen sitting down. As the sequence progresses toward the final view, the panda shifts into an upright position while remaining near the same water trough. Throughout this entire transition, the overall size of the panda within the camera frame remains remarkably consistent. There is no discernible increase in size that would suggest zooming in, nor is there a decrease in size that would indicate the object moving away or the camera moving closer.\n\nWhile the panda changes its posture slightly, the stability of its screen footprint confirms that its size remains constant from the perspective of the camera lens. \n\nBased on this observation, the primary visual change is due to the panda's own movement rather than a change in camera perspective or focal length. Therefore, the object appears unchanged in terms of its size on screen.\n\nThis conclusion aligns with Option C.\n</think>\n<answer>**C. Unchanged: The size ratio remains stable.**</answer>

Figure 15: **Qualitative example.** *4DThinker* tracks the panda's apparent size across frames through latent visual tokens and correctly determines that the size ratio remains stable, distinguishing the panda's posture change from actual camera zoom or physical depth movement.

Visual Input



Query

How does the environment around the road change as the vehicle continues its journey?
A. The road becomes wider and is lined with buildings.
B. The landscape shifts from open fields to dense forest.
C. The roadside vegetation becomes increasingly sparse and rocky.
D. The road transitions from paved to dirt and then back to pavement.

4DThinker-Qwen2.5-VL-3B

<think>\nThe question asks for an analysis of how the environment around the road changes as the vehicle travels forward. Looking at the video, we see a first-person perspective from a vehicle driving along a paved road through a hilly landscape.\n\nTo accurately track the environmental changes, I need to observe the road's geometry and the surrounding terrain as the vehicle progresses. The most reliable way to do this is to focus on a static feature of the environment. I will use the 'road' itself as my primary reference point.\n\nIn my mind, I imagine highlighting the road surface with a red overlay across key moments to track its position and features relative to the frame. I mentally visualize this progression from <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|> to <|latent_start|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_pad|><|latent_end|>, where the road is highlighted in each imagined frame.\n\nAnalyzing this mental sequence, I start by observing the initial view where the road is relatively wide and stretches out ahead toward a distant curve. As the vehicle moves forward, it approaches a bend in the road. By the middle of the segment, the road has clearly shifted toward the bottom-left corner of the frame, and the camera begins to capture more of the left-side embankment and trees.\n\nAs we move toward the end of the sequence, the road narrows significantly as the vehicle prepares to navigate the upcoming curve. The embankment on the right side of the road becomes much more prominent, showing a transition from open fields into a dense forest area. The texture of the road surface changes as it follows the path of the hillside, but the overall composition of the scene is dominated by the shift from open space to dense foliage.\n\nBecause the vehicle is traveling along a winding road that passes through a forested area, the visual environment is characterized by a gradual transition from open landscape to a more wooded, mountainous setting. The road doesn't simply widen or narrow; it evolves from a flat, paved surface into one that follows the contours of the terrain.\n\nThis observation aligns with Option B, which describes the landscape shifting from open fields to dense forest.\n</think>\n<answer>**B. The landscape shifts from open fields to dense forest.**</answer>

Figure 16: **Qualitative example.** Given a first-person driving video, *4DThinker* tracks gradual environmental transitions via 4D latents and predict the open fields to dense forest.

- **Format completeness**: The response must contain properly paired <think>...</think> and <answer>...</answer> tags.
- **Placeholder count**: The number of placeholders must match the number of overlay images.
- **Answer validity**: The generated answer must exactly match one of the predefined options.

Samples failing any check are regenerated up to three times; persistent failures are discarded.

J Qualitative Examples

We present qualitative examples in Figs. 13–16 to illustrate how *4DThinker* leverages its "think with 4D" reasoning process. In each example, the model first articulates the reasoning strategy with a sequence of latent visual tokens (displayed as <|latent_start|>...<|latent_end|>) representing its internal 4D mental imagery, and then derives the answer.

Fig. 13 demonstrates a DSR-Bench example requiring procedural distance tracking, where *4DThinker* correctly decomposes the motion into two temporal phases via latent imagery. Fig. 14 shows a Dyn-Bench example where *4DThinker* captures the global movement pattern (diagonal traversal) that confuses models (e.g., Gemini-3, Qwen2.5-VL-3B) relying on local frame-level heuristics. Fig. 15 illustrates *4DThinker*'s ability to *disentangle object posture changes* from camera-induced size variations through continuous tracking in latent space. Fig. 16 shows that *4DThinker* can anchor on static reference landmarks to *capture gradual scene-level environmental transitions and predict the future scene* in egocentric driving videos.

K Implementation Details

Base model. We build *4DThinker* on a list of base VLM (e.g., Qwen2.5-VL Bai et al. (2025), Qwen3-VL Team (2025), InternVL3.5 Wang et al. (2025b), etc.). The visual encoder is kept *frozen* throughout all training stages.

DIFT training. In terms of Qwen2.5-VL-3B, we train for 1 epochs with a batch size of 1. We use the AdamW optimizer with a learning rate of 1×10^{-5} and latent size 4. The loss weights are set to $\lambda_{\text{ce}} = 0.1$ and $\lambda_{\text{sim}} = 1.0$. Input videos are uniformly sampled to 1 FPS. Training is conducted on 8 NVIDIA H200 141GB GPUs using DeepSpeed ZeRO-2.

Note that training configurations vary slightly across different foundation models. We utilized up to 64 NVIDIA H200 141GB GPUs to train a single model (e.g., InternVL3.5-38B).

4DRL training. We apply modified GRPO with the DSR-Train dataset. In terms of Qwen2.5-VL-3B, the group size is G=8. The reward weights are $\lambda_{\text{acc}} = 1.0$ and $\lambda_{\text{fmt}} = 0.2$. We use a learning rate of 1×10^{-6} with the max completion length $= 8192$ and KL coefficient $\beta = 0.01$. Additionally, the batch size is 8 with 2 gradient accumulation steps.

Configurations also vary slightly across models, with training requiring up to 64 NVIDIA H200 141GB GPUs per model. Note that specific RL hyperparameters exert a disproportionately large influence on the overall training process.

Mask overlay parameters. For generating mask overlays (Eq. equation 1), we use opacity $\alpha = 0.6$ and highlight color $\mathbf{c} = [255, 0, 0]$ (red) for both static and dynamic objects.