CVPR #12

4D-RGPT: Toward Region-level 4D Understanding via Perceptual Distillation

Anonymous CVPR submission

Paper ID 12

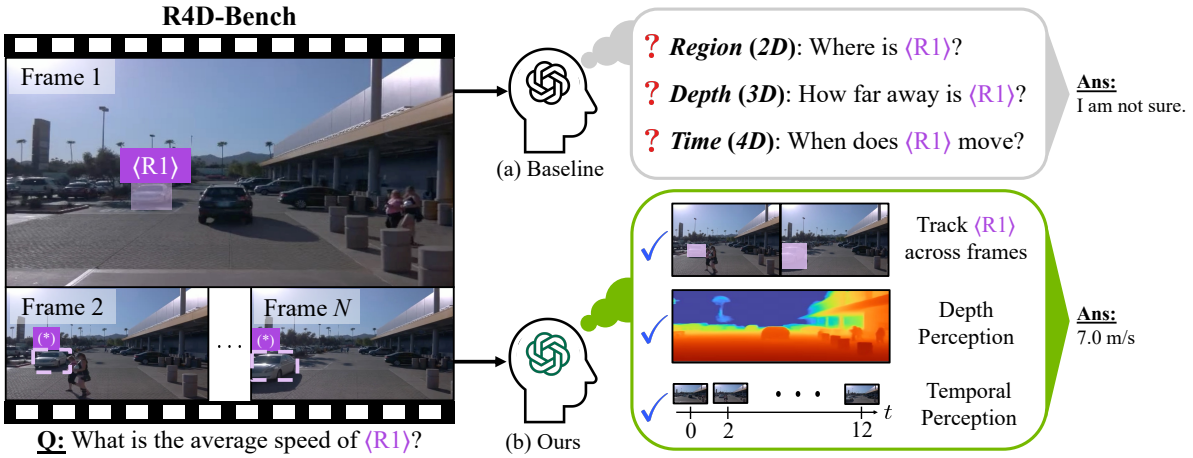

Figure 1. **Overview of Region-level 4D Understanding.** 4D region-level VQA, *e.g.*, our R4D-Bench, requires MLLMs to be able to track regions (2D), perceive depth (3D), and temporal progression (4D). Baseline MLLMs cannot recognize one or more of these aspects and thus fail to answer questions correctly. With our distillation framework, our 4D-RGPT better perceives these aspects and answers accurately. We note that the regions labeled with (*) are not provided in R4D-Bench; they are visualized for readability.

Abstract

Despite advances in Multimodal LLMs (MLLMs), their ability to reason over 3D structures and temporal dynamics remains limited, constrained by weak 4D perception and temporal understanding. Existing 3D and 4D Video Question Answering (VQA) benchmarks also emphasize static scenes and lack region-level prompting. We tackle these issues by introducing: **(a)** *4D-RGPT, a specialized MLLM designed to capture 4D representations from video inputs with enhanced temporal perception;* **(b)** *Perceptual 4D Distillation (P4D), a training framework that transfers 4D representations from a frozen expert model into 4D-RGPT for comprehensive 4D perception; and* **(c)** *R4D-Bench, a benchmark for depth-aware dynamic scenes with region-level prompting, built via a hybrid automated and human-verified pipeline. Our 4D-RGPT achieves notable improvements on both existing 4D VQA benchmarks and the proposed R4D-Bench benchmark.*

1. Introduction

By integrating visual inputs with Large Language Models (LLMs) [1, 14, 45, 73], Multimodal LLMs (MLLMs) demonstrate remarkable capabilities in complex understanding across vision and language modalities. However, current MLLMs, even proprietary models such as GPT-4o [44], often struggle with highly specialized tasks that require fine-grained spatial[1] and temporal visual understanding.

In this paper, we advance MLLMs for one such challenging task: *Region-level 4D Understanding*. This unique problem combines two critical aspects: (1) **4D understanding**, which demands answering questions regarding depth information, temporal dynamics, or object interactions in 3D space over time; and (2) **region-level understanding**, which requires grounding language queries to specific visual regions for controllable input. Region-level 4D VQA is essential for demanding real-world applications, such as autonomous driving and industrial inspection, where 4D information is critical and user queries must precisely target specific regions rather than rely on ambiguous descriptions. As an example, in Fig. 1, the 4D question "*What is the average speed of* ⟨R1⟩*?*" specifically targets the speed of the car marked by the purple bounding box ⟨R1⟩.

To achieve 4D understanding, previous works mainly rely

[1]We use "spatial" in this paper to refer to 3D (*i.e.*, 2D + depth), rather than 2D as in several general video understanding works.

on conventional Supervised Fine-Tuning (SFT) [22, 42, 72, 81] or Reinforcement Learning (RL) [27, 41, 46, 56, 71] paradigms, optimizing primarily over the final text output using self-curated data. However, due to the difficulty of curating large-scale, well-annotated dynamic video data, these works often struggle with dynamic scenarios. In region-level 4D VQA, having strong 4D understanding is even more critical, as it requires tracking region movement over time. More recently, several works [8, 9, 11, 15, 70, 85, 86] exploit external models to inject 3D knowledge into MLLMs to improve spatial understanding capabilities. However, external 3D knowledge mainly helps understand static videos, without fully achieving 4D understanding. Moreover, these approaches often integrate additional modules into the architecture, introducing additional inference burdens.

To address these challenges, we propose ***4D-RGPT***, a specialized MLLM with effective *4D perception* and thus better 4D understanding capabilities. 4D perception refers to the ability to extract low-level 4D perceptual knowledge, *e.g.*, depth and optical flow. Specifically, 4D-RGPT perceives 4D knowledge via our proposed **P**erceptual **4D** **D**istillation (**P4D**) *training-only* framework. P4D adopts both latent and explicit distillation processes to effectively distill 4D perceptual knowledge from an expert 4D teacher model into the student 4D-RGPT. Notably, unlike previous works, P4D contains only *training-only* modules, incurring no additional inference cost. Finally, we introduce Timestamp Positional Encoding (TPE) to provide explicit temporal cues, enhancing MLLMs' temporal perception capability.

Finally, while various 3D/4D VQA benchmarks have been proposed recently [15, 21, 29, 54, 76, 88], they often lack either region-prompted questions or sufficient 4D understanding challenges. As demonstrated in Fig. 1, this limitation prevents comprehensive evaluation of region-based 4D VQA capabilities, namely, answering questions about specific regions (*e.g.*, $\langle R1 \rangle$) in a 4D context. To bridge this gap, we construct ***R4D-Bench***, a new benchmark containing both static and dynamic scene understanding tasks with region-based 4D questions.

Our experiments show that 4D-RGPT improves over the baseline on both non-region-based 3D/4D benchmarks (+5.3% on average across 6 benchmarks) and our region-based R4D-Bench benchmark (+4.3%), while effectively capturing explicit 4D signals.

Our main contributions are as follows:

- We propose *4D-RGPT* (Sec. 4.1), a specialized MLLM that perceives 4D information for enhanced understanding.
- We propose the *P4D* (Sec. 4.2) training framework to distill 4D perceptual knowledge into 4D-RGPT without introducing additional inference cost.
- We introduce *R4D-Bench* (Sec. 5), a region-based 4D VQA benchmark that requires region-level 4D understanding.

Table 1. **Comparison among 3D / 4D VQA Benchmarks.** Existing benchmarks either lack dynamic video data or region prompts, while our R4D-Bench is the first to provide both at scale. All benchmarks are downloaded from official sources as of August 2025, and the numbers of VQA might differ from the original papers. Static videos contain only camera movement, while dynamic videos contain both camera and object movement. †We only adopt real-world videos from the VLM4D benchmark.

Dataset	Regions	Input Type	FPS	# Visual	# QA
SAT-real [54]	✗	Images	-	196	150
MMSI-Bench [76]	✗	Images	-	2.5k	1.0k
OmniSpatial [21]	✗	Images	-	561	1.5k
VSTI-Bench [15]	✗	Static Video	24	312	6k
STI-Bench [29]	✗	Dynamic Video	$10 \sim 30$	369	2k
VLM4D-real† [88]	✗	Dynamic Video	$12 \sim 24$	600	1k
R4D-Bench (Ours)	✓	Dynamic Video	$10 \sim 30$	780	1.5k

2. Related Work

2.1. Multimodal LLMs (MLLMs)

The success of LLMs [1, 3, 14, 45, 63, 64, 73] has inspired various MLLMs [12, 30, 32, 33, 37, 44, 51, 59] for multi-modal understanding or generation. While several MLLMs [36, 55, 57, 78, 87] excel at video understanding, they lack specialization in region-level or 3D/4D tasks.

Region-Level MLLMs understand specified regions within visual inputs. Earlier works [6, 7, 23, 38, 47, 50, 61, 66, 84, 89] use bounding box coordinates as text prompts, while others [11, 31, 40, 43, 67, 82] extract Region of Interest (RoI) visual features. Visual markers [5, 24, 69, 74] provide intuitive region indication. However, region-level video understanding remains challenging, especially for dynamic scenes where user queries provide sparse region annotations without temporal tracking (Fig. 1). While recent works [11, 18] address this, they do not fully explore 4D dynamic scenarios. We propose *4D-RGPT* (Sec. 4.1) to interpret 4D spatio-temporal knowledge without 4D annotations during training.

3D/4D MLLMs focus on spatial and temporal understanding. Previous works [8, 11, 15, 20, 28, 46, 54, 58, 72, 85] enhance MLLMs with depth or 3D reconstruction models but require additional modules, introducing inference costs. Others use SFT [22, 42, 72, 81] or RL [27, 41, 46, 56, 71] with text-based supervision, which is insufficient for 4D perception. We propose *P4D* (Sec. 4.2) to enhance 4D perception without modifying the architecture. 3DRS [20] employs distillation for static 3D scenes, while P4D addresses dynamic scenes with dual distillation on latent and explicit representations to achieve 4D understanding.

2.2. 3D/4D VQA Benchmarks

Several benchmarks evaluate MLLMs' 3D and 4D understanding. OmniSpatial [21], VSTI-Bench [15], SAT [54],

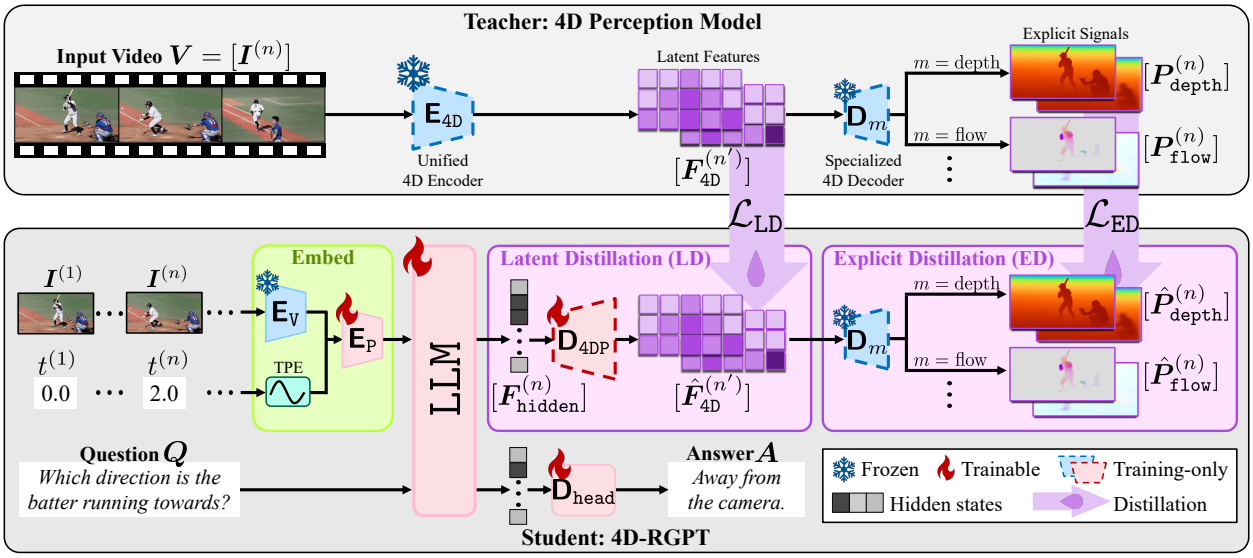

Figure 2. **Perceptual 4D Distillation (P4D) framework for 4D-RGPT.** For each frame $I^{(i)}$ in V, 4D-RGPT extracts 4D representations through training-only modules, *i.e.*, $\mathbf{D}_{4DP}$ and $\mathbf{D}_m$ for $m \in \mathcal{M}$. This includes both latent features, *i.e.*, $\hat{F}_{4D}$, and explicit signals, *e.g.*, depth $\hat{P}_{depth}$ or optical flow maps $\hat{P}_{flow}$. We also incorporate timestamp positional encodings (TPE) to provide temporal cues for 4D-RGPT to be temporally aware. In the P4D framework, the frozen teacher, *i.e.*, 4D perception model, captures 4D expert knowledge from V. It is then distilled to the student 4D-RGPT via two strategies. (a) *Latent Distillation (LD)*: We align the latent $\hat{F}_{4D}$ with the teacher's intermediate 4D embeddings F_{4D}. (b) *Explicit Distillation (ED)*: We align the explicit $\hat{P}_m$ with the teacher's final 4D signals P_m. 4D-RGPT is optimized end-to-end using both SFT loss and the distillation losses, *i.e.*, $\mathcal{L}_{LD}$ and $\mathcal{L}_{ED}$.

and MMSI-Bench [76] focus on 3D spatial understanding in images. STI-Bench [29] is a pioneering work that introduces 4D VQA on both static and dynamic videos, while VLM4D [88] focuses on semantic understanding in dynamic videos. However, these benchmarks lack region-level prompting or sufficient dynamic video data (Tab. 1). We introduce *R4D-Bench* (Sec. 5) with region-level prompts and diverse 4D understanding tasks.

3. Preliminaries

We briefly review the background and introduce notation for an MLLM and a 4D perception model.

Multimodal LLMs extend the understanding capabilities of LLMs to visual inputs such as images and videos. The architecture typically consists of: (a) $\mathbf{E}_V$: a vision encoder for input visuals, *e.g.*, images or videos; (b) $\mathbf{E}_P$: a multi-modal projector that aligns the visual and textual features within a shared space; (c) LLM: an auto-regressive model that takes in both features and generates output hidden states or tokens in a step-by-step manner; (d) $\mathbf{D}_{head}$: a linear head layer that maps the hidden states to the final vocabulary space for text generation.

4D Perception Models, *e.g.*, L4P [2], encode a latent feature from input visuals for multiple 4D low-level representations. They consist of a unified encoder $\mathbf{E}_{4D}$ and specialized decoders $\mathbf{D}_m$ for each 4D modality $m \in \mathcal{M}$. Each 4D modality $m \in \mathcal{M}$ describes some per-pixel 4D properties of the input video. For example, m can be either "depth," which describes the per-pixel depth values, or "flow," which describes the per-pixel optical flow between adjacent frames.

We denote the input video as $V = [I^{(n)}]_{n=1:N}$ with each image frame $I^{(n)} \in \mathbb{R}^{H \times W \times 3}$. Here, N is the number of input frames and (H, W) is the spatial size. Given V, we can acquire its 4D latent representation as follows,

$$F_{4D} = \mathbf{E}_{4D}(V) \in \mathbb{R}^{N' \times h' \times w' \times c'}, \quad (1)$$

where N', h', w' are the down-sampled number of frames, height, and width of $\mathbf{E}_{4D}$'s outputs and c' is the number of output channels.

For each m, the decoder $\mathbf{D}_m$ decodes F_{4D} to its corresponding low-level representation, *i.e.*,

$$P_m = \mathbf{D}_m(F_{4D}). \quad (2)$$

We use the following 4D modalities $\mathcal{M}$ in this work: (a) $m =$ depth where $P_{depth}^{(n)} \in \mathbb{R}^{H \times W \times 1}$ describes the per-pixel depth values; (b) $m =$ flow where $P_{flow}^{(n)} \in \mathbb{R}^{H \times W \times 2}$ describes the per-pixel optical flow between adjacent frames; (c) $m =$ motion where $P_{motion}^{(n)} \in \mathbb{R}^{H \times W \times 1}$ describes whether a pixel is moving or static in 3D space; (d) $m =$ camray where $P_{camray}^{(n)} \in \mathbb{R}^{H \times W \times 6}$ describes the per-pixel Plucker ray maps.

4. Approach

Overview. Given a video $\boldsymbol{V}$ and a question $\boldsymbol{Q}$, an MLLM responds with an answer $\boldsymbol{A}$ autoregressively. To tackle the complex, dynamic scenes presented in 4D VQA benchmarks, we develop an MLLM that can better answer questions by incorporating 4D knowledge from a teacher model and leveraging low-level representations, *e.g.*, depth and flow, over time. To this end, we design ***4D-RGPT*** to capture both *latent* 4D features and *explicit* 4D signals from $\boldsymbol{V}$ with **training-only** modules. These 4D representations enable the model to better perceive 4D knowledge during training, without introducing additional inference cost. Additionally, to accurately capture temporal progression for answering 4D questions, we introduce Timestamp Positional Encoding (TPE) to provide explicit temporal cues to the MLLM.

To circumvent the extreme training cost and instability of training MLLMs from scratch, we introduce our **P**erceptual **4**D **D**istillation (**P4D**) framework to distill 4D knowledge into 4D-RGPT during training. As shown in Fig. 2, our framework leverages a frozen expert 4D perception model (teacher) to supervise both latent and explicit 4D representations of 4D-RGPT (student). The latent distillation provides intermediate guidance on abstract 4D features, while the explicit distillation ensures accurate extraction of interpretable low-level 4D signals. We describe the 4D-RGPT architecture in Sec. 4.1 and the P4D framework in Sec. 4.2.

4.1. 4D-RGPT

Given an input video $\boldsymbol{V}$ with N sampled frames $[\boldsymbol{I}^{(n)}]_{n=1}^{N}$, and the timestamps $\{t^{(n)}\}_{n=1}^{N}$ of each frame, our 4D-RGPT consists of training-only 4D perception modules that can extract 4D representations for distillation in P4D (Sec. 4.2). Moreover, 4D-RGPT can perceive temporal progression by incorporating timestamp positional encodings into input visual features. In short, we use a 4D perception decoder $\mathbf{D}_{\texttt{4DP}}$ to extract latent 4D features and prediction heads $\mathbf{D}_m$ for $m \in \mathcal{M}$ to extract explicit 4D signals.

Latent 4D Representations. To capture latent 4D representations for P4D, we extract $\hat{\boldsymbol{F}}_{\texttt{4D}}$ from the input video. Through the video encoder $\mathbf{E}_{\texttt{V}}$, multi-modal projector $\mathbf{E}_{\texttt{P}}$, and LLM, each frame $\boldsymbol{I}^{(n)}$ is encoded as hidden state features $\boldsymbol{F}^{(n)}_{\texttt{hidden}} \in \mathbb{R}^{h\times w\times c}$, where $l = hw$ is the number of per-image tokens, (h, w) is the spatial size of visual features, and c is the hidden dimension. We introduce a *training-only* MLP as a 4D perception decoder $\mathbf{D}_{\texttt{4DP}}$ on top of the MLLM to decode latent 4D representations $\hat{\boldsymbol{F}}^{(n)}_{\texttt{4D}}$. Specifically, we first sample and resize (`Rearrange`) the hidden $\boldsymbol{F}^{(n)}_{\texttt{hidden}}$ to match the target shape of (N', h', w') in Eq. 1. Thus, for each down-sampled frame $n' \in [1, N']$, we have

$$\hat{\boldsymbol{F}}^{(n')}_{\texttt{4D}} = \mathbf{D}_{\texttt{4DP}}\left(\texttt{Rearrange}(\boldsymbol{F}^{(n)}_{\texttt{hidden}})\right). \tag{3}$$

Explicit 4D Representations. Although $\hat{\boldsymbol{F}}_{\texttt{4D}}$ can capture rich 4D features, explicit 4D signals, *e.g.*, depth maps, are more interpretable and provide unambiguous supervision. To capture explicit 4D representations for P4D, we extract explicit 4D signals $\hat{\boldsymbol{P}}_m$ given $\hat{\boldsymbol{F}}_{\texttt{4D}}$ via the *training-only* prediction heads $\mathbf{D}_m$ from the frozen 4D perception model. Specifically, for each $m \in \mathcal{M}$, we have

$$\hat{\boldsymbol{P}}_m = \mathbf{D}_m(\hat{\boldsymbol{F}}_{\texttt{4D}}). \tag{4}$$

Timestamp Positional Encoding (TPE). Accurate temporal perception, such as "when" an event occurred and "how long" an action took, is fundamental to 4D VQA. For example, to answer "*What is the average speed of the car?*," even if the MLLM can perceive depth and knows its displacement, it still needs to understand the time duration of the video to compute speed. Incorrect temporal perception can lead to significant errors in acquiring the displacement over the correct time duration, *i.e.*, speed.

We observe that MLLMs struggle with temporal perception when there are no explicit time cues (see the experiments in Sec. 6.3 and Tab. 6). To provide temporal cues, we encode timestamps directly into the MLLM's visual input as positional encodings. That is, for each input frame $\boldsymbol{I}^{(n)}$ from video $\boldsymbol{V}$ that is sampled at time $t^{(n)}$, we add a sinusoidal timestamp positional encoding $\boldsymbol{p}^{(n)} \in \mathbb{R}^D$ to the visual features $\mathbf{E}_{\texttt{V}}(\boldsymbol{I}^{(n)})$ before feeding them into the $\mathbf{E}_{\texttt{P}}$, where

$$\boldsymbol{p}^{(n)}[2i] = \sin\left(\frac{t^{(n)}}{T^{\frac{2i}{D}}}\right) \text{ and } \boldsymbol{p}^{(n)}[2i+1] = \cos\left(\frac{t^{(n)}}{T^{\frac{2i}{D}}}\right). \tag{5}$$

Here T is the maximum timescale and i is the index.

4.2. Perceptual 4D Distillation (P4D)

To answer 4D questions, MLLMs must understand not only semantic content but also various aspects of 4D knowledge, such as sub-pixel movements and numeric depth values. For example, to answer "*Is the person moving closer to the camera?*", the MLLM must compare the depth values of the *person* across frames. Recent 3D/4D specialized MLLMs either rely on self-curated training datasets or exploit external models to enhance 3D knowledge. However, both are insufficient for MLLMs to fully achieve 4D understanding. Moreover, introducing external modules results in additional inference costs. Therefore, a mechanism that provides direct supervision on the MLLM's internal 4D perception capabilities without introducing additional modules is desirable.

To this end, we propose our P4D framework. We leverage an existing 4D perception model as a teacher to transfer its expert representations to our student, 4D-RGPT. To ensure comprehensive knowledge transfer, we propose dual-branch distillation: latent distillation and explicit distillation.

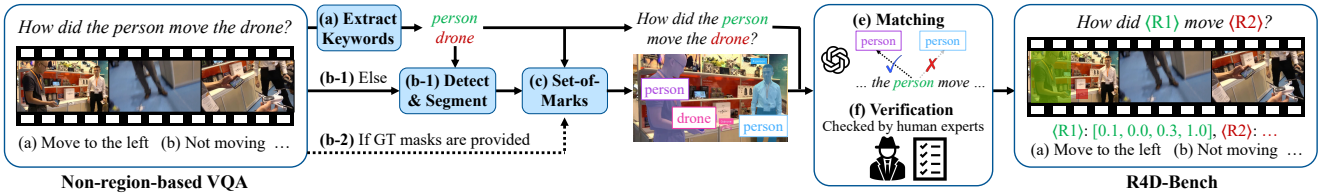

Figure 3. **Curation pipeline of our R4D-Bench.** Given existing non-region 4D VQA benchmarks, we (a) first extract the noun keywords from the question as candidates for objects of interest. (b) Next, if ground truth segmentation masks are provided, we use them for step (d). Otherwise, we use off-the-shelf GroundingDINO [35] and SAM2 [53] to extract segmentation masks for each object of interest. (c) We generate a SoM [74] image for the first frame. (d) We prompt Qwen-2.5VL [51] with the SoM image and the processed question to match the objects referred to in the question with the regions. (e) Finally, the generated matching results are verified by human experts.

Latent Distillation. We start by introducing latent distillation to supervise the MLLM's latent 4D representations, *i.e.*, $\hat{\boldsymbol{F}}_{\text{4D}}$, on the latent space. Latent distillation serves as intermediate 4D guidance to the MLLM on the latent space. Specifically, our latent distillation loss $\mathcal{L}_{\text{LD}}$ is defined to pull the margin Δ_{LD} between the latent 4D features from the teacher model $\boldsymbol{F}_{\text{4D}}$ and those from the student model $\hat{\boldsymbol{F}}_{\text{4D}}$:

$$\mathcal{L}_{\text{LD}} = \sum_{n'=1}^{N'} \Delta_{\text{LD}}(\boldsymbol{F}_{\text{4D}}^{(n')}, \hat{\boldsymbol{F}}_{\text{4D}}^{(n')}). \tag{6}$$

Explicit Distillation. On the other hand, we introduce explicit distillation to supervise the MLLM's explicit 4D representations, *i.e.*, $\hat{\boldsymbol{P}}_m$, on the signal space. Explicit distillation provides direct, interpretable supervision to ensure the MLLM captures accurate 4D signals in $\mathcal{M}$. Specifically, our explicit distillation loss $\mathcal{L}_{\text{ED}}$ is defined to pull the margin Δ_m between the explicit 4D signals from the teacher model $\boldsymbol{P}_m$ and those from the student model $\hat{\boldsymbol{P}}_m$:

$$\mathcal{L}_{\text{ED}} = \sum_{n=1}^{N} \sum_{m \in \mathcal{M}} \lambda_m \Delta_m(\boldsymbol{P}_m^{(n)}, \hat{\boldsymbol{P}}_m^{(n)}), \tag{7}$$

where λ_m describes the loss weights of each m.

Training. We optimize our 4D-RGPT using both SFT and P4D. The overall loss function is a combination of the standard cross-entropy SFT loss $\mathcal{L}_{\text{SFT}}$, latent distillation loss $\mathcal{L}_{\text{LD}}$, and explicit distillation loss $\mathcal{L}_{\text{ED}}$. We train on various 3D / 4D conversation datasets, including RoboFAC [39], SAT [54], VSTI-Bench [15] (the training split), and Wolf [26]. Please refer to the supplementary material for more training details.

5. R4D-Bench

Recently, there has been significant progress in 3D/4D VQA [15, 21, 29, 54, 76, 88]. Several new benchmarks require MLLMs to have depth perception or understand 3D interactions among objects. However, existing benchmarks do not evaluate MLLMs on 4D region-based understanding in complex, real-world scenarios. As shown in Tab. 1, they lack the following critical properties:

- **Lack of Dynamic Scenes**: Most focus on indoor scenes with minimal object interaction or constrained movement, which do not fully capture the complexity of real-world object manipulation and dynamic changes.
- **Lack of Region Prompting**: Region prompts allow controlled and intuitive user queries in VQA. Without this ability, an MLLM's interpretability and usability in practical applications are hindered.

To address these gaps, we introduce ***R4D-Bench*** (see the rightmost example in Fig. 3), a novel benchmark that challenges MLLMs with region-level 4D VQA, where depth and temporal perception are critical.

Task Formulation. Given an input video $\boldsymbol{V} = [\boldsymbol{I}^{(n)}]_{n=1:N}$ of N frames, a region-prompted 4D question $\boldsymbol{Q}$, and a set of region masks $\boldsymbol{M}$ describing the objects of interest in $\boldsymbol{Q}$ in $\boldsymbol{I}^{(1)}$, the task is to respond with the correct or most suitable answer from a set of options.

Benchmark. We curate R4D-Bench based on existing non-region-based 4D VQA benchmarks, *i.e.*, STI-Bench [29] and VLM4D [88]. Our pipeline (Fig. 3) employs a hybrid automated and human-verified process to transform conventional VQ pairs into highly specific region-prompted questions.

The process begins with a non-region-prompted 4D VQA. In the example of Fig. 3, we are given a video of two persons and a drone with the query question "How did the person move the drone?" First, we use Qwen2.5-VL [51] to perform keyword extraction (**Extract**) and identify objects of interest from the query question, *e.g.*, the *person* and the *drone*. While videos from some sources, *e.g.*, DAVIS [49], provide annotations of object masks, other real-world videos lack such detailed annotations. Hence, we leverage state-of-the-art object detection and segmentation models, *i.e.*, GroundingDINO [35] and SAM2 [53], to generate accurate object masks (**Detect & Segment**) for the identified objects of interest. We then apply the segmentation masks with their corresponding keywords onto the video frame to generate an image with **Set-of-Marks** [74]. This serves as an intermediate and potential portrayal of the region-prompted QA before the final step of checking correctness.

Since the objects of interest can be non-unique (*e.g.*, multiple persons) and segmentation masks can be noisy, ensuring

correct association between extracted keywords and found regions is critical. We check correctness with both automated and human-in-the-loop processes. We use Qwen2.5-VL [51] to automatically match the generated region marks to the entities in the question (**Matching**). Finally, human annotators verify and correct any mismatches (**Verification**). We also trim videos to ensure all RoIs are visible in the first frame.

This concludes our region prompting process. The original VQA is transformed into R4D-Bench format, where entities are replaced by region tokens, *e.g.*, "How did $\langle R1 \rangle$ move $\langle R2 \rangle$?" with their corresponding region masks.

Statistics. Our R4D-Bench benchmark consists of 1,517 region-prompted VQAs. Each question is a multiple-choice problem with four to five answer options. The benchmark provides region-prompted challenges to semantic and numerical 4D understanding in both static and dynamic scenes. The static split (418 VQAs) includes 3 categories: (1) Dimension Measurement; (2) 3D Video Grounding; and (3) Spatial Relation. The dynamic split (1,098 VQAs) includes 6 categories: (1) Counting objects; (2) Translational movement; (3) Rotational movement; (4) False Positive detection; (5) Speed & Acceleration estimation; and (6) Displacement & Path Length measurement. We provide more details for each question type in the supplementary material.

6. Experiments

6.1. Experiment Setup

Benchmarks. We evaluate our 4D-RGPT on various 4D VQA benchmarks, including our R4D-Bench and existing ones, *i.e.*, STI-Bench [29], VLM4D-real [88], OmniSpatial [21], MMSI-Bench [76], SAT [54], and VSTI-Bench [15]. Please note that the first four benchmarks are testing-only benchmarks and are disjoint from our training data. Apart from the numerical questions in VSTI-Bench, where we report relative accuracy, we report the multiple-choice accuracy for all other benchmarks.

Comparison Models. We compare our 4D-RGPT with various proprietary MLLMs, *e.g.*, GPT-4o [44], GPT-5 [45], Gemini-2.5-Pro [12]; open-source generalized MLLMs, *e.g.*, Qwen2.5-VL [51]; and recent 3D/4D specialized MLLMs, *e.g.*, SpatialReasoner [41], ViLaSR [71], and SpaceR [46].

Architecture. We select a SOTA open-source generalized MLLM, NVILA-Lite-8B [37], as our MLLM backbone, which uses SigLIP [79] as the $\mathbf{E}_{\texttt{V}}$ and Qwen2 [60] as the `LLM`. For the 4D perception model $\mathbf{E}_{\texttt{4D}}$ and $\mathbf{D}_m$, we follow the exact architecture and weights of L4P [2]. We document training setups in the supplementary material.

Table 2. **Evaluation on non-region-level 3D / 4D benchmarks.** We report the average multiple-choice accuracy (↑) on each benchmark. For simplicity, we use the following abbreviations: STI (STI-Bench [29]), V4D (VLM4D-real [88]), MMSI (MMSI-Bench [76]), OS (OmniSpatial [21]), and VSTI (VSTI-Bench [15]).

Methods	STI	V4D	MMSI	OS	SAT	VSTI
GPT-4o [44]	34.8	60.0	30.3	47.8	57.5	38.2
GPT-5 [45]	39.3	-	40.7	59.9	-	-
Gemini-2.5-Pro [12]	41.4	63.5	36.9	55.4	-	-
Gemini-1.5-Pro [59]	-	-	-	-	64.8	-
InternVL2.5-8B [10]	-	42.4	28.7	-	-	-
Qwen2.5-VL-7B [51]	32.1	43.3	25.9	39.2	-	-
VideoLLaMA3-7B [80]	35.2	46.5	-	-	-	-
LLaVA-Video-7B [83]	-	-	-	-	53.5	-
LLaVA-OneVision-7B [25]	29.0	36.0	24.5	35.7	41.7	-
LLaVA-NeXT-Video-7B [34]	29.9	-	26.8	-	-	40.0
VLM-3R-7B [15]	-	-	-	-	-	58.8
LLaVA-Video-7B + SAT [54]	-	-	-	-	63.4	-
ViLaSR-7B [71]	33.4	46.9	30.2	-	-	-
SpatialReasoner-7B [41]	31.0	43.4	22.7	-	-	-
SpaceR-7B [46]	37.0	51.3	28.8	-	47.8	-
NVILA-Lite-8B [37]	33.8	46.5	31.3	37.2	62.0	45.2
4D-RGPT-8B (Ours)	**37.6** +3.8	**52.7** +6.2	**33.3** +2.0	**40.4** +3.2	**64.7** +2.7	**59.1** +13.9

6.2. Main Results

We present the effectiveness of 4D-RGPT in Tab. 2 and Tab. 3, showing improvements over baseline MLLMs.

Non-region-based 4D VQA. In Tab. 2, we evaluate 4D-RGPT on several non-region-level 3D/4D VQA benchmarks, including input modalities of both images and videos. We compare with various state-of-the-art proprietary MLLMs, open-source general MLLMs, and recent 3D/4D MLLMs. 4D-RGPT consistently improves over the baseline NVILA-Lite-8B by a large margin across all benchmarks, especially on VLM4D [88] and VSTI-Bench [15]. Compared to other MLLMs with similar model sizes, 4D-RGPT achieves SOTA performance over open-source MLLMs and competitive performance with GPT-4o [44]. Please note that SpatialReasoner [41], ViLaSR [71], and SpaceR [46] are all trained with RL to further boost accuracy.

R4D-Bench. In Tab. 3, we present quantitative comparisons of our 4D-RGPT on R4D-Bench against other MLLMs. For fair comparison, we use SoM [74] to indicate the regions of interest for all MLLMs. Additionally, for all open-source MLLMs and 4D-RGPT, we use the same number of sampled frames, *i.e.*, 16 frames. We observe that although SpaceR [46] outperforms Qwen2.5-VL [51] in Tab. 2, it falls behind on R4D-Bench, suggesting that SpaceR is highly tuned for non-region VQA and its region understanding is weakened. Overall, 4D-RGPT achieves the best performance among all open-source MLLMs by at least 1.6% on average and 2.6% on the dynamic split.

Table 3. **Evaluation on R4D-Bench.** We report performance on the static split (Sta), the dynamic split (Dyn), and all 9 tasks of R4D-Bench. For simplicity, we abbreviate them as follows: 3D **V**ideo **G**rounding (VG); **D**imension **M**easurement (DM); **S**patial **R**elationship (SR); **R**otational (R); **C**ounting (C); **T**ranslational (T); **F**alse **P**ositive (FP); **S**peed & **A**cceleration (SA); and **D**isplacement & **P**ath Length (DP).

Methods	Avg	Sta	Dyn	VG	DM	SR	R	C	T	FP	SA	DP
Random	23.4	20.0	24.7	20.0	20.0	20.0	25.0	25.0	25.0	25.0	20.0	20.0
GPT-4o [44]	42.8	30.3	47.5	30.7	26.8	43.9	49.1	35.2	51.8	54.1	27.0	10.7
Qwen2.5-VL-7B [51]	40.6	**34.1**	43.1	39.1	25.7	48.8	50.0	38.4	46.6	28.9	45.9	28.6
LLaVA-Video-7B [83]	39.7	26.9	44.6	23.4	28.4	36.6	46.2	30.2	50.4	33.6	48.6	35.7
ViLaSR-7B [71]	39.6	31.5	42.6	34.4	24.6	48.8	46.2	42.8	51.3	3.7	43.2	17.9
SpatialReasoner-7B [41]	38.3	31.2	41.0	35.4	25.7	36.6	43.4	37.1	49.3	11.9	32.4	17.9
SpaceR-7B [46]	37.0	26.2	41.1	30.7	18.0	41.5	47.2	40.3	43.8	25.9	51.4	21.4
NVILA-Lite-8B [37]	37.9	29.1	41.3	33.9	20.2	46.3	41.5	39.6	41.9	40.7	45.9	32.1
4D-RGPT-8B (Ours)	**42.2**	32.9	**45.7**	35.1	26.3	52.2	43.1	40.1	48.7	40.2	50.9	38.9
	+4.3	**+3.8**	**+4.4**	**+1.2**	**+6.1**	**+5.9**	**+1.6**	**+0.5**	**+6.8**	**-0.5**	**+5.0**	**+6.8**

Table 4. **Alternative strategies for 4D VQA.** We compare P4D with direct SFT (*4D-SFT*) and straightforward designs of incorporating $\boldsymbol{F}_{4D}$ from the 4D perception model, *i.e.*, *4D-Concat* and *4D-PE*. For simplicity, we use the same abbreviations as in Tab. 3 and STI for STI-Bench [29].

Methods	$\boldsymbol{F}_{4D}$	STI	R4D-Bench		
			Avg	Sta	Dyn
Zero-shot	✗	33.8	37.9	29.1	41.3
4D-SFT	✗	34.7	40.1	32.2	43.8
4D-Concat	✓	34.8	39.5	30.6	42.9
4D-PE	✓	31.3	36.0	26.6	39.5
Ours (*P4D*)	✓	**37.6**	**42.2**	**32.9**	**45.7**

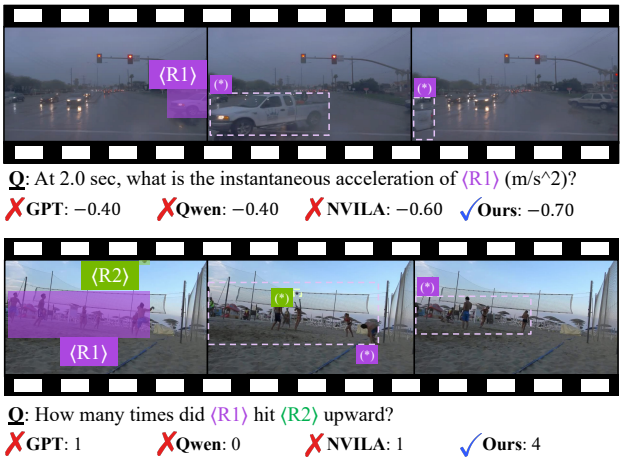

Figure 4. **VQA comparison among baseline MLLMs and 4D-RGPT on R4D-Bench.** For the baseline MLLMs, we use GPT-4o-20241120 [44], Qwen-2.5VL-7B-Instruct [51], and NVILA-Lite-8B [37]. We note that the regions labeled with (*) or (*) are not provided in R4D-Bench; they are visualized for readability.

In Fig. 4, we showcase two cases of 4D-RGPT against other MLLMs on R4D-Bench. In both cases, the regions of interest are constantly moving. Only 4D-RGPT effectively perceives the 4D dynamics and provides the correct answers.

6.3. Ablation Studies

To justify our various designs, we conduct extensive ablation studies and analysis. For most experiments in this subsection, we report results on STI-Bench [29] and the static and dynamic question subsets of R4D-Bench. Without specific notes, we use the same training data, and all other components are kept identical unless specified.

Alternative Strategies. Besides P4D, there are other strategies to utilize 4D conversation data or the latent feature $\boldsymbol{F}_{4D}$ from the 4D perception models to enhance MLLMs' 4D understanding. First, denoted as *4D-SFT*, we apply solely SFT to the entire MLLM without access to $\boldsymbol{F}_{4D}$. Additionally, there are two straightforward ways to leverage $\boldsymbol{F}_{4D}$. Denoted as *4D-Concat*, we directly concatenate $\boldsymbol{F}_{4D}$ with the 2D visual features $\boldsymbol{E}_{V}(\boldsymbol{V})$. We note that this requires additional training on $\mathbf{E}_{P}$ as the dimension differs from the original visual features. On the other hand, denoted as *4D-PE*, we project $\boldsymbol{F}_{4D}$ to positional encodings (PE) for the visual features, similar to the spatial PE proposed in SR-3D [11].

As shown in Tab. 4, apart from *4D-PE*, both *4D-SFT* and *4D-Concat* improve over the *Zero-shot* baseline. However, they all fall short compared to P4D. Moreover, *4D-Concat* and *4D-PE* require additional inference costs as they need to compute $\boldsymbol{F}_{4D}$ for each input during inference. In comparison, P4D requires solely training-only 4D perception modules, making 4D-RGPT as efficient as *Zero-shot* during inference.

Perceptual 4D Distillation. To validate the effectiveness of P4D, we experiment with various distillation strategies used in latent distillation ($\mathcal{L}_{LD}$ in Eq. (6)) and explicit distillation ($\mathcal{L}_{ED}$ in Eq. (7)). In Tab. 5, we ablate different combinations of distillation on $\hat{\boldsymbol{F}}_{4D}$ and $\hat{\boldsymbol{P}}_{m}$.

We first observe that applying $\mathcal{L}_{LD}$ alone (*LD-only*) improves the performance over the *Zero-shot* baseline by 2.3% on R4D-Bench. For $\mathcal{L}_{ED}$, adding more $m \in \mathcal{M}$ incrementally improves the performance steadily, with $m = \texttt{depth}$ and $m = \texttt{flow}$ being the most effective ones (see *LD+D* and *LD+D+F*). While $\mathcal{L}_{ED}$ alone (*ED-only*) also improves the performance on R4D-Bench by 1.9%, combining both (*LD+ED*) achieves the best average performance, showing

Table 5. **Analysis of 4D modalities in P4D.** We ablate the effectiveness of different combinations of distillation in latent distillation (LD) on $\hat{\boldsymbol{F}}_{4D}$ and explicit distillation (ED) on $\hat{\boldsymbol{P}}_m$. For simplicity, we use the same abbreviations as Tab. 4 and **D**epth (D), **F**low (F), **M**otion (M), and **C**amray (C) for each $m \in \mathcal{M}$.

Methods	$\hat{\boldsymbol{F}}_{4D}$	$\hat{\boldsymbol{P}}_m$				STI	R4D-Bench		
		D	F	M	C		Avg	Sta	Dyn
Zero-shot	✗	✗	✗	✗	✗	33.8	37.9	29.1	41.3
LD-Only	✓	✗	✗	✗	✗	34.2	40.2	32.0	43.3
LD+D	✓	✓	✗	✗	✗	33.4	40.8	32.5	44.0
LD+D+F	✓	✓	✓	✗	✗	36.2	41.9	**33.1**	45.3
LD+D+F+M	✓	✓	✓	✓	✗	36.5	42.0	**33.1**	45.4
ED-Only	✗	✓	✓	✓	✓	35.4	39.8	31.5	42.9
Ours (*LD+ED*)	✓	✓	✓	✓	✓	**37.6**	**42.2**	32.9	**45.7**

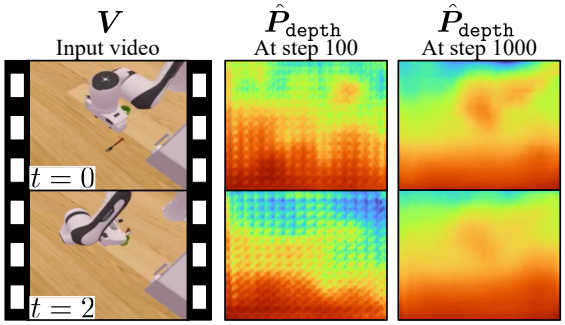

Figure 5. **Predicted depth maps at different training steps.** We visualize the progress of $\hat{\boldsymbol{P}}_{\texttt{depth}}$ throughout training.

Table 6. **Ablation studies on explicit temporal cues.** We experiment without and with different choices of explicit time cues. For simplicity, we use the same abbreviations as Tab. 4.

Methods	Time cues	STI	R4D-Bench		
			Avg	Sta	Dyn
Zero-shot	✗	33.8	37.9	29.1	41.3
P4D	✗	34.8	41.0	31.8	44.5
P4D+mark	marks	35.1	41.1	31.5	44.7
P4D+prompt	prompts	36.1	41.5	32.1	45.0
Ours (*P4D+TPE*)	TPE	**37.6**	**42.2**	**32.9**	**45.7**

the complementary benefits of both LD and ED.

4D Perception Visualization. In Fig. 5, we visualize the progress of how 4D-RGPT learns to extract 4D signals through P4D. We show a video from our training set [39] with extracted $\hat{\boldsymbol{P}}_{\texttt{depth}}$ at various steps. $\hat{\boldsymbol{P}}_{\texttt{depth}}$ is barely meaningful at first but gradually captures the 3D structure of the scene as training proceeds. This indicates that P4D successfully distills 4D perception capabilities into 4D-RGPT.

Timestamp Positional Encoding (TPE). MLLMs often struggle with temporal perception when no explicit time cues are provided. We conduct a controlled toy experiment to validate this observation by curating a simple benchmark with VQAs that require temporal perception, such as "*How many seconds have passed in the input video?*" We observe that NVILA-Lite-8B [37] is naively guessing the answers, resulting in accuracy close to random guessing. This problem is further exacerbated by the inconsistency among multiple sources of data with different frame rates. We detail the toy experiment in the supplementary material.

Without introducing additional modules, we test two simple solutions to provide explicit temporal cues to MLLMs. First, denoted as *P4D+mark*, we add explicit time marks similar to SoM [74] on each $\boldsymbol{I}^{(n)}$, such as burned-in text showing the timestamp, *e.g.*, "$t^{(n)}$ s" Second, denoted as *P4D+prompt*, we add explicit time information in $\boldsymbol{Q}$, such as "*The following video frames are sampled from a video 19 seconds long and recorded at 30 frames per second.*"

Both *P4D+mark* and *P4D+prompt*, as shown in Tab. 6, can improve 4D VQA performance. However, they require additional data preprocessing, distract MLLMs from the main visual and textual content, and do not generalize well to region-level settings, *i.e.*, R4D-Bench. Our *P4D+TPE* consistently improves performance across both benchmarks, as shown in the last row of Tab. 6.

Table 7. **Ablation studies on different training designs in 4D-RGPT.** We ablate different training designs on whether each module is trainable and whether to use LoRA [19]. For simplicity, we use the same abbreviations as Tab. 4.

Methods	Trainable			STI	R4D-Bench		
	$\mathbf{E}_{\texttt{V}}$	$\mathbf{E}_{\texttt{P}}$	LLM		Avg	Sta	Dyn
Zero-shot	✗	✗	✗	33.8	37.9	29.1	41.3
Tune-All	✓	✓	✓	34.7	38.8	30.1	42.1
Tune-V	✓	✗	✗	32.3	35.8	27.3	39.0
Tune-P	✗	✓	✗	34.3	38.6	29.8	42.0
Tune-LLM	✗	✗	✓	35.4	40.5	32.2	43.7
Tune-LLM-LoRA	✗	✗	LoRA	37.0	41.1	**33.0**	44.2
Tune-P+LLM-LoRA	✗	✓	LoRA	36.5	41.4	32.8	44.7
Ours (*Tune-P+LLM*)	✗	✓	✓	**37.6**	**42.2**	32.9	**45.7**

Architecture Design. In Tab. 7, we ablate different designs on whether $\mathbf{E}_{\texttt{V}}$, $\mathbf{E}_{\texttt{P}}$, or LLM is trainable or frozen. Our *Tune-P+LLM* achieves the best performance by tuning both $\mathbf{E}_{\texttt{P}}$ and LLM, while keeping $\mathbf{E}_{\texttt{V}}$ frozen. This is likely because $\mathbf{E}_{\texttt{P}}$ requires finetuning for TPE and P4D works best on LLM.

7. Conclusion

We show that existing MLLMs struggle with region-level 4D VQA due to not fully perceiving 4D information. Without incurring additional inference cost, our 4D-RGPT effectively improves MLLMs' 4D perception by learning from a 4D perception model via a novel distillation framework, P4D. Additionally, we introduce a proper benchmark, R4D-Bench, for this domain, contributing to region-level 4D VQA. Extensive experiments confirm the effectiveness of our approach on both non-region-level and region-level 4D VQA.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 2

[2] Abhishek Badki, Hang Su, Bowen Wen, and Orazio Gallo. L4P: Low-level 4D vision perception unified. *arXiv preprint arXiv:2502.13078*, 2025. 3, 6

[3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 2

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proc. CVPR*, 2020. 1

[5] Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. ViP-LLaVA: Making large multimodal models understand arbitrary visual prompts. In *Proc. CVPR*, 2024. 2

[6] Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. LION: Empowering multimodal large language model with dual-level visual knowledge. In *Proc. CVPR*, 2024. 2

[7] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 2

[8] Pingyi Chen, Yujing Lou, Shen Cao, Jinhui Guo, Lubin Fan, Yue Wu, Lin Yang, Lizhuang Ma, and Jieping Ye. SD-VLM: Spatial measuring and understanding with depth-encoded vision-language models. In *Proc. NeurIPS*, 2025. 2

[9] Yiming Chen, Zekun Qi, Wenyao Zhang, Xin Jin, Li Zhang, and Peidong Liu. Reasoning in space via grounding in the world. *arXiv preprint arXiv:2510.13800*, 2025. 2

[10] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 6

[11] An-Chieh Cheng, Yang Fu, Yukang Chen, Zhijian Liu, Xiaolong Li, Subhashree Radhakrishnan, Song Han, Yao Lu, Jan Kautz, Pavlo Molchanov, et al. 3d aware region prompted vision language model. *arXiv preprint arXiv:2509.13317*, 2025. 2, 7

[12] 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. 2, 6

[13] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. CVPR*, 2017. 1

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1, 2

[15] 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. 2, 5, 6, 1, 7

[16] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. AISTATS*, 2010. 1

[17] D Hendrycks. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. 1

[18] Miran Heo, Min-Hung Chen, De-An Huang, Sifei Liu, Subhashree Radhakrishnan, Seon Joo Kim, Yu-Chiang Frank Wang, and Ryo Hachiuma. Omni-RGPT: Unifying image and video region-level understanding via token marks. In *Proc. CVPR*, 2025. 2

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*, 2022. 8

[20] Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. Mllms need 3d-aware representation supervision for scene understanding. In *Proc. NeurIPS*, 2025. 2

[21] Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. OmniSpatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*, 2025. 2, 5, 6

[22] Dohwan Ko, Sihyeon Kim, Yumin Suh, Minseo Yoon, Manmohan Chandraker, Hyunwoo J Kim, et al. ST-VLM: Kinematic instruction tuning for spatio-temporal reasoning in vision-language models. *arXiv preprint arXiv:2503.19355*, 2025. 2

[23] Byung-Kwan Lee, Beomchan Park, Chae Won Kim, and Yong Man Ro. CoLLaVO: Crayon large language and vision mOdel. In *Proc. ACL*, 2024. 2

[24] Xuanyu Lei, Zonghan Yang, Xinrui Chen, Peng Li, and Yang Liu. Scaffolding coordinates to promote vision-language coordination in large multi-modal models. In *Proc. ACL*, 2025. 2

[25] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *TMLR*, 2025. 6

[26] Boyi Li, Ligeng Zhu, Ran Tian, Shuhan Tan, Yuxiao Chen, Yao Lu, Yin Cui, Sushant Veer, Max Ehrlich, Jonah Philion, et al. Wolf: Dense video captioning with a world summarization framework. *TMLR*, 2025. 5, 1, 2, 7

[27] 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*, 2025. 2

[28] Pengteng Li, Pinhao Song, Wuyang Li, Weiyu Guo, Huizai Yao, Yijie Xu, Dugang Liu, and Hui Xiong. See&trek:

CVPR #12

Training-free spatial prompting for multimodal large language model. In *Proc. NeurIPS*, 2025. 2

[29] Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. STI-Bench: Are MLLMs ready for precise spatial-temporal world understanding? In *Proc. ICCV*, 2025. 2, 3, 5, 6, 7, 4

[30] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On pre-training for visual language models. In *Proc. CVPR*, 2024. 2

[31] Weifeng Lin, Xinyu Wei, Ruichuan An, Peng Gao, Bocheng Zou, Yulin Luo, Siyuan Huang, Shanghang Zhang, and Hongsheng Li. Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want. In *Proc. ICLR*, 2025. 2

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proc. NeurIPS*, 2023. 2

[33] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proc. CVPR*, 2024. 2

[34] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, ocr, and world knowledge, 2024. 6

[35] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Proc. ECCV*, 2024. 5, 2

[36] Zikang Liu, Longteng Guo, Yepeng Tang, Tongtian Yue, Junxian Cai, Kai Ma, Qingbin Liu, Xi Chen, and Jing Liu. Vrope: Rotary position embedding for video large language models. In *Proc. EMNLP*, 2025. 2

[37] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. NVILA: Efficient frontier visual language models. In *Proc. CVPR*, 2025. 2, 6, 7, 8, 1

[38] Jinghui Lu, Haiyang Yu, Yanjie Wang, Yongjie Ye, Jingqun Tang, Ziwei Yang, Binghong Wu, Qi Liu, Hao Feng, Han Wang, et al. A bounding box is worth one token-interleaving layout and text in a large language model for document understanding. In *ACL Findings*, 2025. 2

[39] Weifeng Lu, Minghao Ye, Zewei Ye, Ruihan Tao, Shuo Yang, and Bo Zhao. RoboFAC: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*, 2025. 5, 8, 1, 2, 7

[40] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *Proc. ECCV*, 2024. 2

[41] Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso M de Melo, Jianwen Xie, and Alan Yuille. SpatialReasoner: Towards explicit and generalizable 3d spatial reasoning. In *Proc. NeurIPS*, 2025. 2, 6, 7

[42] Wufei Ma, Luoxin Ye, Celso M de Melo, Alan Yuille, and Jieneng Chen. Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models. In *Proc. CVPR*, 2025. 2

[43] Yunze Man, De-An Huang, Guilin Liu, Shiwei Sheng, Shilong Liu, Liang-Yan Gui, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. ARGUS: Vision-centric reasoning with grounded chain-of-thought. In *Proc. CVPR*, 2025. 2

[44] OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1, 2, 6, 7, 8, 9

[45] OpenAI. Gpt-5. https://openai.com/chatgpt, 2025. Large language model. 1, 2, 6

[46] 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. 2, 6, 7

[47] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. Grounding multimodal large language models to the world. In *Proc. ICLR*, 2024. 2

[48] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proc. CVPR*, 2016. 2

[49] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv:1704.00675*, 2017. 5, 2

[50] Shraman Pramanick, Guangxing Han, Rui Hou, Sayan Nag, Ser-Nam Lim, Nicolas Ballas, Qifan Wang, Rama Chellappa, and Amjad Almahairi. Jack of all tasks master of many: Designing general-purpose coarse-to-fine vision-language model. In *Proc. CVPR*, 2024. 2

[51] Alibaba Group Qwen Team. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2, 5, 6, 7, 3

[52] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proc. ICCV*, 2021. 1

[53] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In *Proc. ICLR*, 2025. 5, 2

[54] Arijit Ray, Jiafei Duan, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, Kuo-Hao Zeng, et al. SAT: Spatial aptitude training for multimodal language models. In *Proc. COLM*, 2025. 2, 5, 6, 1, 7

[55] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proc. CVPR*, 2024. 2

[56] Yifan Shen, Yuanzhe Liu, Jingyuan Zhu, Xu Cao, Xiaofeng Zhang, Yixiao He, Wenming Ye, James Matthew Rehg, and Ismini Lourentzou. Fine-grained preference optimization improves spatial reasoning in vlms. In *Proc. NeurIPS*, 2025. 2

[57] Yumeng Shi, Quanyu Long, Yin Wu, and Wenya Wang. Causality matters: How temporal information emerges in video language models. *arXiv preprint arXiv:2508.11576*, 2025. 2

[58] Peiwen Sun, Shiqiang Lang, Dongming Wu, Yi Ding, Kaituo Feng, Huadai Liu, Zhen Ye, Rui Liu, Yun-Hui Liu, Jianan Wang, et al. Spacevista: All-scale visual spatial reasoning from mm to km. *arXiv preprint arXiv:2510.09606*, 2025. 2

[59] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 2, 6

[60] Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 6

[61] Yunjie Tian, Tianren Ma, Lingxi Xie, Jihao Qiu, Xi Tang, Yuan Zhang, Jianbin Jiao, Qi Tian, and Qixiang Ye. ChatterBox: Multi-round multimodal referring and grounding. In *Proc. AAAI*, 2025. 2

[62] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020-2025. Open source software available from https://github.com/HumanSignal/label-studio. 3

[63] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[64] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2

[65] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *Proc. CVPR*, 2023. 1

[66] Weiyun Wang, Yiming Ren, Haowen Luo, Tiantong Li, Chenxiang Yan, Zhe Chen, Wenhai Wang, Qingyun Li, Lewei Lu, Xizhou Zhu, et al. The All-Seeing project v2: Towards general relation comprehension of the open world. In *Proc. ECCV*, 2024. 2

[67] Weiyun Wang, Min Shi, Qingyun Li, Wenhai Wang, Zhenhang Huang, Linjie Xing, Zhe Chen, Hao Li, Xizhou Zhu, Zhiguo Cao, et al. The all-seeing project: Towards panoptic visual recognition and understanding of the open world. In *Proc. ICLR*, 2024. 2

[68] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 1

[69] Sangmin Woo, Kang Zhou, Yun Zhou, Shuai Wang, Sheng Guan, Haibo Ding, and Lin Lee Cheong. Black-box visual prompt engineering for mitigating object hallucination in large vision language models. In *Proc. NAACL*, 2025. 2

[70] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. In *Proc. NeurIPS*, 2025. 2

[71] Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. In *Proc. NeurIPS*, 2025. 2, 6, 7

[72] Runsen Xu, Weiyao Wang, Hao Tang, Xingyu Chen, Xiaodong Wang, Fu-Jen Chu, Dahua Lin, Matt Feiszli, and Kevin J Liang. Multi-SpatialMLLM: Multi-frame spatial understanding with multi-modal large language models. *arXiv preprint arXiv:2505.17015*, 2025. 2

[73] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 1, 2

[74] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 2, 5, 6, 8, 3

[75] 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 *Proc. CVPR*, 2025. 1

[76] Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, et al. MMSI-Bench: A benchmark for multi-image spatial intelligence. *arXiv preprint arXiv:2505.23764*, 2025. 2, 3, 5, 6

[77] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3d indoor scenes. In *Proc. ICCV*, 2023. 1

[78] Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, et al. Timesuite: Improving mllms for long video understanding via grounded tuning. In *Proc. ICLR*, 2025. 2

[79] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proc. ICCV*, 2023. 6, 1

[80] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 6

[81] Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. In *Proc. NeurIPS*, 2025. 2

[82] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Yu Liu, Kai Chen, and Ping Luo. GPT4RoI: Instruction tuning large language model on region-of-interest. In *Proc. ECCV Workshop*, 2024. 2

[83] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 6, 7

[84] Liang Zhao, En Yu, Zheng Ge, Jinrong Yang, Haoran Wei, Hongyu Zhou, Jianjian Sun, Yuang Peng, Runpei Dong, Chunrui Han, and Xiangyu Zhang. ChatSpot: Bootstrapping multimodal llms via precise referring instruction tuning. In *Proc. IJCAI*, 2024. 2

[85] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. In *Proc. NeurIPS*, 2025. 2

[86] Hanyu Zhou and Gim Hee Lee. LLaVA-4D: Embedding spatiotemporal prompt into lmms for 4d scene understanding. *arXiv preprint arXiv:2505.12253*, 2025. 2

[87] Honglu Zhou, Xiangyu Peng, Shrikant Kendre, Michael S Ryoo, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. Strefer: Empowering video llms with space-time referring and reasoning via synthetic instruction data. In *Proc. ICCV*, 2025. 2

[88] Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Eric Xin Wang, and Achuta Kadambi. VLM4D: Towards spatiotemporal awareness in vision language models. In *Proc. ICCV*, 2025. 2, 3, 5, 6

[89] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *Proc. ICLR*, 2024. 2