

STI-Bench: Are MLLMs Ready for Precise Spatial-Temporal World Understanding?

Yun Li^{1,2}, Yiming Zhang^{1,3}, Tao Lin¹, XiangRui Liu^{1,4}, Wenxiao Cai⁵, Zheng Liu⁴, Bo Zhao¹

¹School of AI, Shanghai Jiao Tong University

²China University of Geosciences, ³Nanyang Technological University, ⁴BAAI, ⁵Stanford University

Corresponding to <bo.zhao@sjtu.edu.cn>

Abstract

The use of Multimodal Large Language Models (MLLMs) as an end-to-end solution for Embodied AI and Autonomous Driving has become a prevailing trend. While MLLMs have been extensively studied for visual semantic understanding tasks, their ability to perform precise and quantitative spatial-temporal understanding in real-world applications remains largely unexamined, leading to uncertain prospects. To evaluate models' Spatial-Temporal Intelligence, we introduce STI-Bench, a benchmark designed to evaluate MLLMs' spatial-temporal understanding through challenging tasks such as estimating and predicting the appearance, pose, displacement, and motion of objects. Our benchmark encompasses a wide range of robot and vehicle operations across desktop, indoor, and outdoor scenarios. The extensive experiments reveal that the state-of-the-art MLLMs still struggle in real-world spatial-temporal understanding, especially in tasks requiring precise distance estimation and motion analysis. Paper Page: <https://mirasjtu.github.io/STI-Bench.io/>

1. Introduction

The rapid development of Multimodal Large Language Models (MLLMs) [1, 4, 12, 26, 32, 34–36, 41, 46] has propelled them to the research forefront as a versatile tool to deal with numerous vision and multimodal tasks. Impressive performances have been achieved by MLLMs in general Visual Question Answering tasks [3], which mainly focus on the 2D visual perception and semantic question answering [18–20, 24, 37, 38, 48].

Beyond 2D visual perception, it has become a prevailing trend to employ MLLMs as an end-to-end solution for Embodied AI [7–9, 15, 22, 25, 29, 43] and Autonomous Driving [21, 39, 40, 44]. Such tasks require MLLMs to understand the 3D space and time, and then predict optimal manipulation strategies for robotic and vehic-

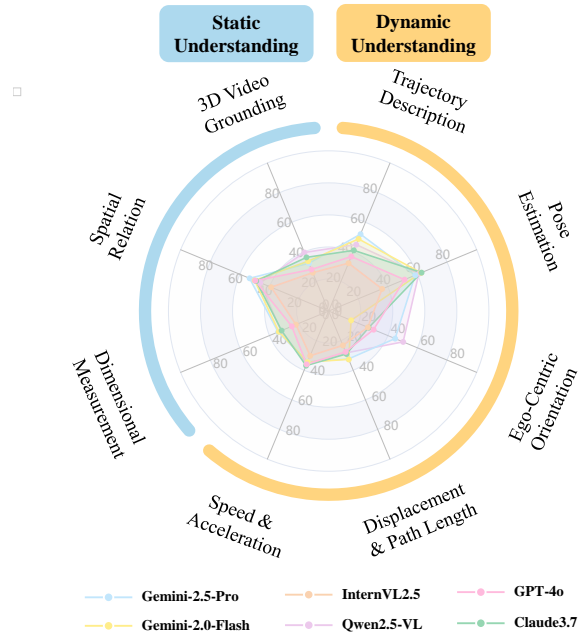


Figure 1. We evaluate state-of-the-art MLLMs on STI-Bench for precise and quantitative spatial-temporal understanding using video inputs. Results indicate the significant challenge in all tasks.

ular systems. Although many explorations have been conducted, the question remains: Are MLLMs ready for precise spatial-temporal world understanding?

To answer this question, we propose a **Spatial-Temporal Intelligence Benchmark (STI-Bench)**, designed to evaluate MLLMs' spatial-temporal world understanding capability. We evaluate MLLMs using single video or multiple images as input instead of 3D point clouds. The main reasons are: 1) the majority of state-of-the-art models, e.g., GPT-4o [30] and Gemini [35], can accept images or video as input rather than 3D point clouds; 2) Videos are more frequently used in human's daily life and they usually contain suffi-

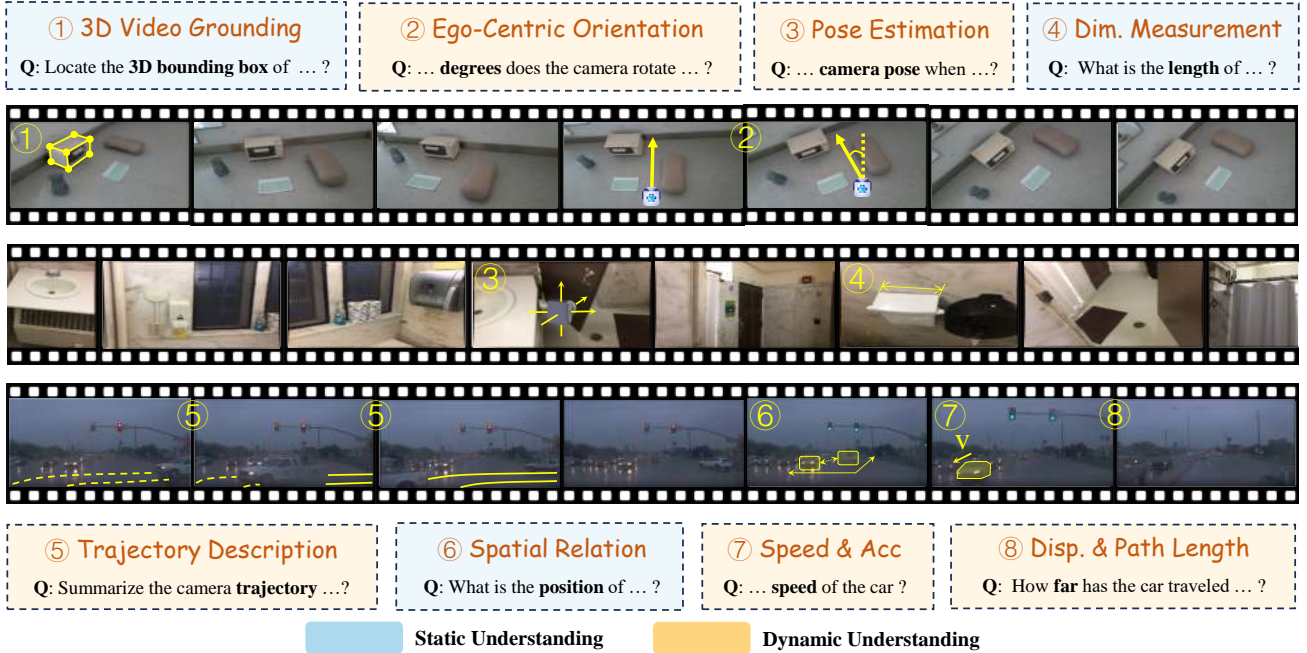


Figure 2. **Overview of STI-Bench.** We selected the most representative videos from each dataset scene and provided a few simple questions for demonstration.

cient information to infer the spatial-temporal environment. STI-Bench contains 300 videos and more than 2,000 QA pairs, covering three major scenarios: *Desktop*, *Indoor*, and *Outdoor*. The videos are sourced from Omni6DPose [47], ScanNet [16] and Waymo [33] respectively, thus encompassing a broad spectrum of real-world environments. As illustrated in Figure 2, we design eight distinct tasks to evaluate models’ ability of static spatial measurement and grounding, and dynamic tasks including speed, acceleration and trajectory estimation.

Through extensive experiments as illustrated in Figure 1, we observe that even the most advanced MLLMs struggle with real-world spatial-temporal understanding, especially in tasks requiring precise distance estimation and motion analysis. Our error analysis reveals three fundamental limitations: inaccurate spatial quantification, flawed temporal dynamics understanding, and weak cross-modal grounding and integration.

These insights highlight the significant challenges MLLMs face in precisely understanding spatial-temporal information from videos. We believe STI-Bench will serve as an important touchstone that guides the community to distinguish and develop better MLLMs for Embodied AI, Autonomous Driving tasks and beyond.

In summary, our main contributions include:

- We present STI-Bench, comprising over 300 videos and more than 2,000 tailored questions across desktop, in-

door, and outdoor scenarios, providing a systematic quantitative assessment of MLLMs’ spatial-temporal understanding capabilities.

- We conduct an in-depth study of state-of-the-art video-based MLLMs on STI-Bench, identify key error patterns in spatial-temporal reasoning, and provide empirical insights that can help the community develop more reliable MLLMs for embodied applications.

2. Related Work

2.1. Multimodal Large Language Models

Multimodal large language models (MLLMs) have achieved groundbreaking performance in visual understanding [1, 4, 12, 35], leveraging large language models (LLMs) [34, 36, 41] and visual encoders. Beyond image-based MLLMs, recent advancements have extended multimodal learning to video understanding. Classical works include models like VideoChat[23], which enable interactive video-based dialogue by integrating multimodal understanding. Subsequent models like Subsequent models like Video-LLaVA[27] enhance visual-language alignment through large-scale vision-language pretraining and fine-tuned adaptation, extending LLaVA[28]’s capability to process video inputs effectively. Recent works, Qwen2.5-VL [41] excels in long-video understanding and temporal localization by incorporating absolute temporal

Benchmark	QA Pairs	Data	Env.	Scene			View		Evaluation		Spatial-Temporal			
				D	I	O	Ego	Allo.	Num.	Desc.	Dist.	Dir.	Vel.	Traj.
SAT [31]	218k	I	S	✗	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗
VSI-Bench [42]	5,156	V	R	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✓
EmbSpatial-Bench [17]	3,640	I	R	✗	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗
EmbodiedAgentInterface [25]	448	-	S	✗	✓	✗	✓	✗	-	-	✗	✗	✗	✗
EmbodiedEval [15]	328	I/V	S	✗	✓	✓	✓	✗	-	-	✗	✗	✗	✗
EmbodiedBench [43]	1,128	I	S	✗	✓	✓	✓	✗	-	-	✗	✗	✗	✗
WorldSense [6]	3,172	V	R	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
MLVU [48]	3,102	V	R	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
Video-MMMU [20]	300	V	S	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
STI-Bench	2,064	V	R	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1. **Comparison of STI-Bench with existing benchmarks.** **Data** represents the source of our QA data, where **V** stands for Video and **I** stands for Image. **Env.** indicates the environment in which the data is generated, where **S** represents Simulation and **R** represents Real. The two columns under **View** indicate whether the dataset includes Ego-centric and Allocentric perspectives. The two columns under **Evaluation** specify whether the ground truth is presented in numerical or textual form. The four columns under **Spatial-Temporal** indicate whether the benchmark evaluates spatial distance, direction (with angular precision), velocity, or a precise and comprehensive trajectory description.

encoding, enabling the model to capture relationships among video frames more effectively. Additionally, its advancement in dynamic resolution modeling allows for seamless adaptation to videos with varying sampling rates, enhancing its versatility in processing diverse video inputs.

2.2. Spatial Understanding with MLLMs

Video MLLMs have attached great importance on semantic understanding. However, spatial understanding has always been a significant challenge, inspiring recent contribution [9, 11, 14]. This progress represents a significant step toward developing world models and embodied agents. Recent advancements in embodied intelligence have explored integrating large-scale MLLMs into robotic control, enabling better generalization and semantic reasoning. RT-2 [8] introduces a vision-language-action framework that transfers web-scale knowledge to robotic control by representing actions as tokens alongside visual and language data, allowing robots to generalize to novel objects and infer multi-step reasoning tasks. Building on this idea, GR-2 [10] extends generalist robot control across diverse embodiments using a Transformer-based architecture trained on a wide range of robotic tasks, demonstrating adaptability across different platforms. Further refining this approach, π_0 [7] incorporates a flow-matching mechanism to generate continuous, precise action trajectories, enhancing fine-grained manipulation skills. By integrating pretrained MLLMs with an independent action module, π_0 achieves zero-shot task execution and flexible adaptation through fine-tuning. Together, these models highlight the potential of leveraging large-scale learning for robotic control, pushing the boundaries of generalization, task adaptability, and multi-modal reasoning in embodied AI.

2.3. Video Benchmarks for MLLM

Recently, multiple benchmarks [19, 37, 38, 48] have emerged for comprehensively evaluating MLLMs’ ability of (long) video understanding, especially about visual perception and semantic reasoning in the form of Video Question Answering. LongVideoBench [38] and LVBench [37] focus on the understanding of long videos. Recent published benchmarks like Video-MME [19] and MMBench-Video [18] comprehensively evaluates MLLMs across various video-related tasks. Existing benchmarks primarily focus on high-level semantic understanding, such as entity recognition and event understanding. In addition, they are largely confined to a temporal extension of 2D image understanding, lacking precise 3D spatial and temporal reasoning of physical quantities. Recent works such as VSI-Bench [42], have shed light on a deeper understanding of the natural world by introducing visual-spatial intelligence tasks for MLLMs, where models are required to provide numerical answers in certain scenarios. However, as illustrated in Table 1, the limited inclusion of scenes and spatial-temporal tasks restricts their ability to capture the complexities of the real physical world. In contrast, STI-Bench comprehensively evaluate models’ ability of precise spatial-temporal understanding in tasks of static spatial measurement and physically motion understanding in Desktop, Indoor and Outdoor scenarios.

3. STI-Bench

In this section, we present the detailed design and construction of STI-Bench. The construction pipeline is depicted in Figure 4.

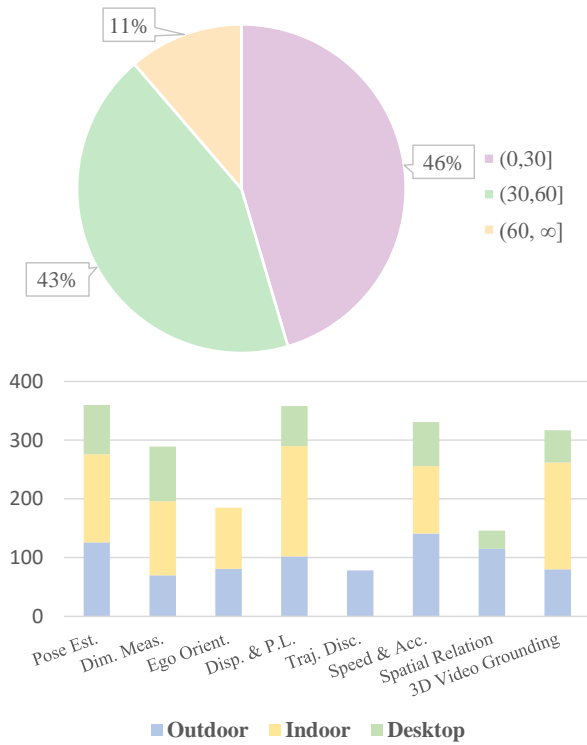


Figure 3. **Benchmark Statistics.** Top: Video length distribution across different categories and datasets. Bottom: The number of questions contributed by each dataset for evaluating different capabilities.

3.1. Task Definition

We propose eight tasks in total, each one systematically examining a distinct aspect of MLLMs’ spatial-temporal understanding. We divide these tasks into two main categories: Static Understanding and Dynamic Understanding.

Static Understanding

- a. **Dimensional Measurement.** Concerns estimates of an object’s geometric size, such as length, width, and height, as well as the distance between objects or between the camera and an object. This requires the ability to transform 2D pixel observations into physical world measurements and accurately perceive depth from monocular inputs.
Example: “What is the height of this box?” or “How close is the camera to the table?”
- b. **Spatial Relation.** Focuses on identifying spatial relationships among objects or between the camera and an object, including front and back, left and right, up and below. This task tests models’ ability to understand relative positioning across different reference frames and maintain spatial relationship judgment consistency across varying viewpoints.

Example: “Is the chair on the left or right side of the table?” or “What is the position of the red bag relative to the fur sofa?”

- c. **3D Video Grounding.** Given a semantic description such as “the red backpack on the brown sofa,” the goal is to retrieve the object’s 3D bounding box in the camera coordinate system at a specific point in the video. This requires seamlessly aligning linguistic descriptions with visual features and accurately parameterizing 3D positional information.

Example: “Locate the 3D bounding box of the red suitcase near the bed.”

Dynamic Understanding

- d. **Displacement and Path Length.** Focuses on how far an object or the camera travels between two given time points. This requires tracking consistent reference points across frames and integrating motion information from discrete frames into continuous paths.
Example: “How far has the car traveled from 1s to 18s?”
- e. **Speed and Acceleration.** Investigates motion parameters by integrating spatial displacement with time intervals. This tests models’ ability to compute spatial derivatives with respect to time and maintain scale consistency across varying distances and perspectives.
Example: “What is the average speed of the camera?” or “How quickly is the ball accelerating?”
- f. **Ego-Centric Orientation.** Examines how the camera’s azimuth orientation, parallel to the ground plane, changes over the duration of the video. This requires understanding rotation representations and utilizing fixed scene elements as angular reference points.
Example: “How many degrees does the camera’s horizontal orientation shift from the start of the video to its end?”
- g. **Trajectory Description.** Describes or infers the camera’s or an object’s motion path throughout the entire video, potentially involving multiple segments of travel and turns. This tests the ability to segment complex trajectories into meaningful components and abstract spatial motion patterns into concise language descriptions.
Example: “Summarize the camera trajectory, including distances moved and turns made.”
- h. **Pose Estimation.** Given the camera’s initial 3D pose, including position and orientation, estimates its pose at a specified point in the video using only the observed RGB data. This requires visual odometry capabilities and the ability to manage cumulative error in long sequences.
Example: “Given the initial pose of the camera, what is the camera’s pose at the requested time?”

Each of these tasks presents unique challenges that collectively evaluate models’ comprehensive spatial-temporal intelligence across different scales, from millimeter-

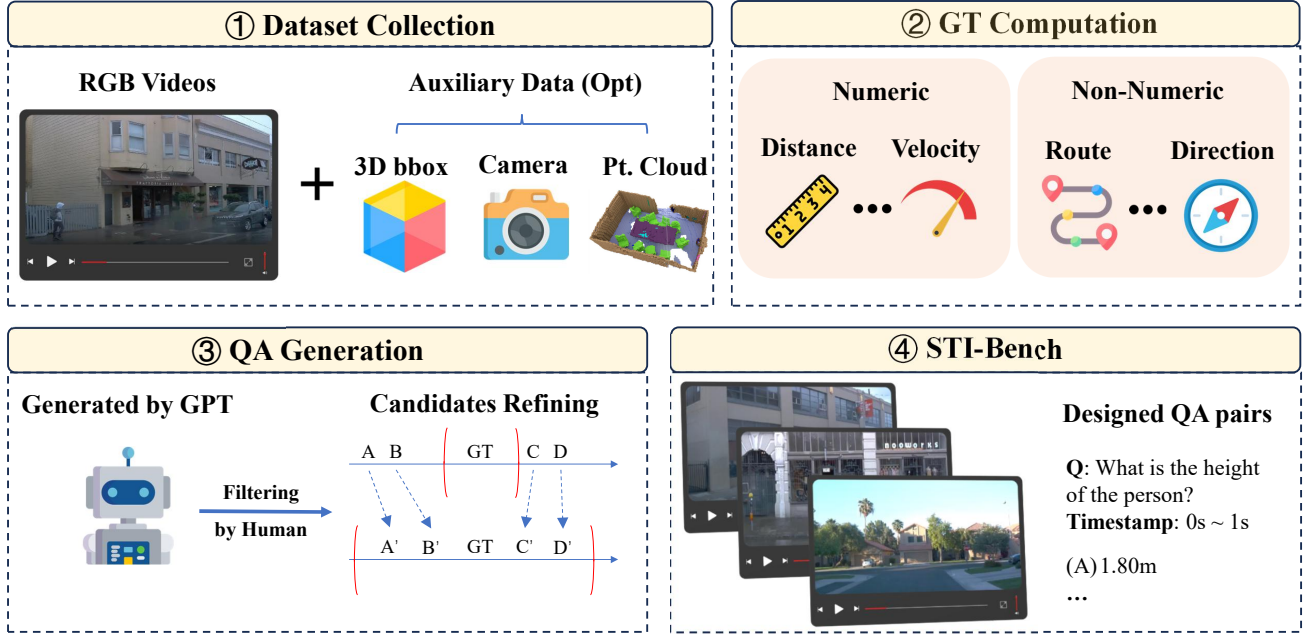


Figure 4. **Benchmark curation pipeline.** The pipeline first aggregates multi-scene RGB datasets that contain 3D bounding box annotations, camera parameters, and point cloud data, which serve as the basis for computing ground truth. From these datasets, we extract numerical ground truth such as distance and velocity, as well as textual descriptions of trajectories and directions. Subsequently, we leverage GPT to assist in generating QA pairs and design a website for rigorous human verification and filtering.

precision desktop manipulation to meter-scale indoor navigation and beyond. Success in these tasks requires not only fundamental 3D spatial reasoning but also physical common sense and the ability to integrate information across different modalities and reference frames over time.

3.2. Benchmark Construction

Data Collection. To encompass a broad spectrum of real-world environments, **STI-Bench** covers three major scenarios: *Desktop*, *Indoor*, and *Outdoor*. Accordingly, we draw from three publicly available datasets—**Omni6DPose**[47] for desktop-scale 6D object pose estimation, **ScanNet**[16] for indoor 3D scene reconstruction, and **Waymo** [33] for autonomous driving. These datasets provide frame-by-frame camera intrinsic and extrinsic parameters, as well as point clouds for each object, which we map to two-dimensional bounding boxes in each frame.

Automatic QA Pair Generation. We used MLLMs to produce detailed semantic descriptions for each object, such as “A beige minivan with a roof rack,” “A refrigerator with emoji magnets, photos, and a to-do list,” or “A red backpack on a brown leather sofa.” Next, leveraging the frame-by-frame annotations, we computed the ground-truth information required for each task. We then provided the ground-truth data, object descriptions, and task-specific QA

requirements to MLLMs to generate a diverse set of questions and challenging answer options.

Human Quality Control. During QA pair generation, several issues arose:

1. LLM-generated descriptions could be inaccurate or fail to uniquely identify the target object.
2. Some questions and options remained unreasonable or incorrect, even with detailed guidelines.
3. In certain cases, the video alone did not provide sufficient information. For example, the camera was occluded but lidar data were available.

To address these challenges, we developed a website for multiple rounds of manual filtering and sampling-based review, ensuring high-quality questions. We also randomly shuffled the answer options to enhance evaluation robustness. Ultimately, we curated more than 2,000 high-quality QA pairs from over 300 videos. Details are shown in Figure 3.

Fine-Grained Adjustment. After generating and refining the QA pairs, we recognized that real-world applications differ significantly in terms of error tolerance. For instance, a desktop robotic arm may require millimeter-level precision, whereas autonomous driving can function effectively with meter-scale accuracy. To accommodate these

varied needs, we applied a scaling factor to the numerical differences between correct answers and distractors, aligning them with the precision requirements of specific scenarios. Consequently, the smallest margin of error ranges from millimeters to centimeters in desktop settings, centimeters to decimeters indoors, and decimeters to meters outdoors. We also adopted a logarithmic sampling approach to avoid clustering most differences at the higher bounds of each range. This fine-grained adjustment preserves the semantic value of each question while maintaining suitable gradients across different precision levels, enabling more effective training and evaluation of MLLMs in diverse environments and industries.

4. Experiments

4.1. Settings

We conduct a thorough evaluation of leading MLLMs from diverse model families, focusing on both proprietary and open-source solutions. Specifically, we assess the performance of four proprietary models, GPT-4o[30], Gemini-2.0-Flash[35], Gemini-2.5-Pro[35], and Claude-3.7-Sonnet[2], as well as several representative open-source MLLMs that have undergone specialized video-related training, including Qwen2.5-VL-72B[5], InternVL2.5-78B[13], MiniCPM-V-2.6[?], VideoChat-Flash[?] and VideoLLaMA3-7B[45].

Considering the stability of open-source models and the API limitations of proprietary models, we uniformly sample 30 frames from the video for each record and explicitly indicate the sampling FPS (Frames Per Second) for the current video within the prompt. An exception is made for Claude-3.7-Sonnet, for which only 20 frames are sampled due to its API constraints. Our benchmark tasks are presented in a multiple-choice format with five possible answers, hence a random guess baseline yields a 20% accuracy. We measure each model’s accuracy by directly comparing the model’s selected answer with the ground truth, without employing any additional external models or annotations for performance evaluation.

4.2. Main Results

As shown in Table 2, we present a comprehensive evaluation of various MLLMs on STI-Bench. Overall, Qwen2.5-VL-72B achieves the highest average accuracy of 41.3% among all tested models, slightly outperforming Gemini-2.5-Pro (40.9%). While these results significantly exceed the random guess baseline (20%), they still highlight substantial room for improvement in spatial-temporal understanding.

When analyzing performance across different scene types, we observe varied patterns. Qwen2.5-VL demonstrates significantly better performance in outdoor scenarios

(49.2%) compared to its indoor (35.6%) and desktop performance (36.2%). Conversely, Gemini-2.5-Pro shows its strongest performance in indoor environments (38.6%), followed by outdoor (47.5%) and then desktop scenes (33.1%). Other models like Claude-3.7-Sonnet and Gemini-2.0-Flash show strength in desktop scenarios (both 37.4%). This suggests varying model specializations, possibly influenced by training data distribution or architectural differences in handling different environmental cues.

Task-specific performance reveals particularly challenging areas. Models generally struggle with tasks requiring precise quantitative estimation. For instance, the best performance on Displacement & Path Length is achieved by Gemini-2.0-Flash (32.7%), and on Dimensional Measurement by Gemini-2.5-Pro (34.2%). Speed & Acceleration also presents difficulties, with Claude-3.7-Sonnet leading at 36.9%. In contrast, models demonstrate stronger capabilities in Pose Estimation (best: 62.7% by Claude-3.7-Sonnet) and Spatial Relation tasks (best: 53.4% by Gemini-2.5-Pro). Notably, VideoChat-Flash achieves the highest score in Ego Orientation (64.7%) despite its lower overall average.

Among open-source models, Qwen2.5-VL stands out, demonstrating highly competitive performance and achieving the top overall rank. However, other open-source models like InternVL2.5, VideoLLaMA3-7B, and MiniCPM-V-2.6 still lag significantly behind the top performers and larger counterparts.

It is important to emphasize that even the best-performing model, Qwen2.5-VL, achieves only 41.3% average accuracy on our benchmark. While this is considerably better than random guessing, it remains far from the reliability required for demanding real-world applications like embodied AI or autonomous driving. These results indicate that current MLLMs, despite their impressive general visual understanding capabilities, still require significant advancements in precise spatial-temporal intelligence for embodied tasks.

4.3. Experimental Analysis

Given that Gemini-2.5-Pro exhibits strong multi-modal reasoning capabilities, provides detailed thinking processes (as exemplified in Figure 6), and ranks second-best overall (best among proprietary models), we select it as a representative for in-depth analysis.

Our analysis of Gemini-2.5-Pro reveals several key characteristics of current state-of-the-art MLLMs’ spatial-temporal understanding capabilities. Overall, the model achieves 40.9% average accuracy. Performance varies across scene types, with the strongest results in indoor scenarios (38.6%), followed by outdoor (47.5%) and desktop environments (33.1%). This performance disparity suggests that the model’s training data likely emphasized outdoor

Methods	Rank	Avg.	Static Understanding			Dynamic Understanding					
			Dim. Meas.	Spatial Relation	3D Video Grounding	Disp. & P.L.	Speed & Acc.	Ego Orient.	Traj. Desc.	Pose Est.	
Proprietary Models (API)											
GPT-4o[30]	5	34.8	24.9	49.6	28.1	27.6	36.0	30.3	36.8	51.3	
Claude-3.7-Sonnet[2]	3	39.4	31.8	49.0	36.3	29.0	36.9	27.0	41.0	62.7	
Gemini-2.0-Flash[35]	4	38.7	33.7	50.0	33.7	32.7	34.4	15.1	48.7	62.4	
Gemini-2.5-Pro[35]	2	40.9	34.2	53.4	32.3	32.4	34.3	44.9	52.0	58.4	
Open-source Models											
VideoLLaMA3-7B[45]	8	26.9	25.6	36.3	24.3	22.4	27.5	23.8	28.2	31.7	
MiniCPM-V-2.6[?]]	8	26.9	27.7	44.5	29.0	19.0	25.7	7.0	30.8	35.6	
InternVL2.5-78B[13]	7	28.4	22.1	38.9	26.0	23.2	30.2	26.5	32.4	36.1	
VideoChat-Flash[?]]	6	34.1	27.9	21.2	29.7	27.4	22.3	64.7	45.8	46.4	
Qwen2.5-VL-72B[5]	1	41.3	32.2	46.2	39.6	28.3	35.6	50.3	44.9	61.0	

Table 2. Evaluation on Ourbench. Orange marks the best result, and Light Orange marks the second best.

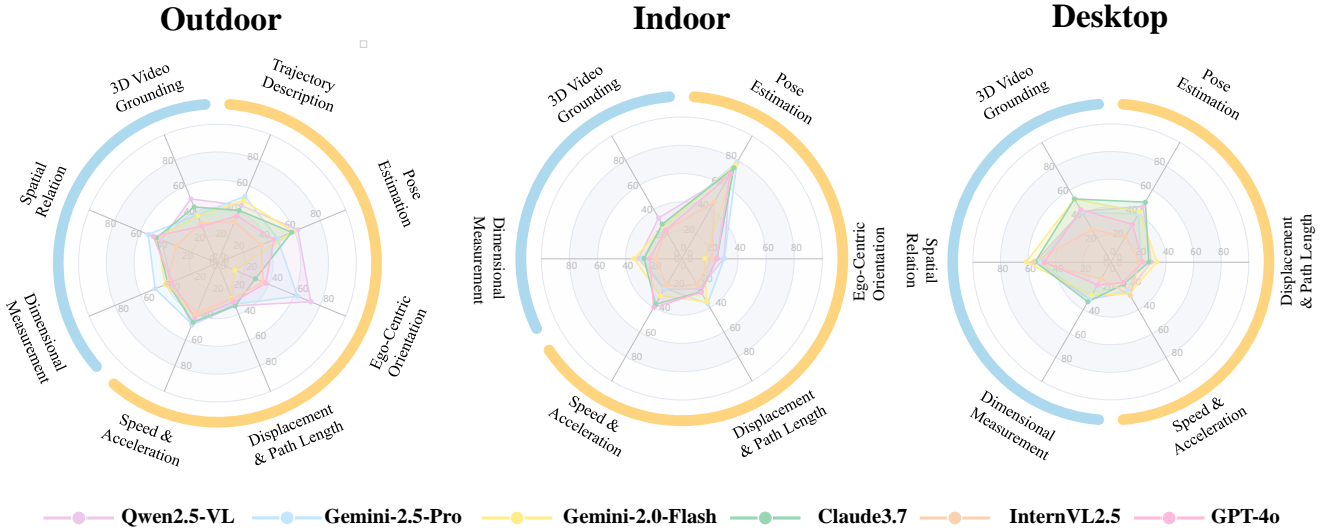


Figure 5. Evaluation results across different scenes and tasks.

Model	Outdoor	Indoor	Desktop	Overall
VideoLLaMA3-7B	26.61	27.63	25.86	26.89
MiniCPM-V-2.6	27.87	26.13	26.60	26.89
InternVL2.5	34.11	25.89	23.81	28.39
VideoChat-Flash	36.12	35.32	27.53	34.08
GPT-4o	39.12	34.45	27.41	34.82
Claude-3.7-Sonnet	43.43	36.53	37.41	39.35
Gemini-2.0-Flash	40.71	37.57	37.41	38.74
Gemini-2.5-Pro	47.48	38.55	33.09	40.88
Qwen2.5-VL	49.24	35.63	36.23	41.32

Table 3. Evaluation results across different scenes.

scenes and larger-scale understanding, or that outdoor environments typically provide clearer visual cues for spatial reasoning.

When examining task-specific performance for Gemini-2.5-Pro (Table 2), we observe its strongest capabilities in tasks involving relative understanding and object state. It achieves its highest score in Pose Estimation (58.4%), followed by Spatial Relation (53.4%, also the best overall score for this task among all models). Its performance in Ego Orientation is 44.9%. Interestingly, while tasks requiring precise quantitative estimation generally show lower absolute scores across models, Gemini-2.5-Pro performs very competitively or even leads in these areas: it achieves the top score in Dimensional Measurement (34.2%) and the second-best score in Displacement & Path Length (32.4%).



Example 1

From 21.0 seconds to 24.0 seconds. What is the average speed of the camera between 21s and 24s?

A.0.13m/s B.0.44m/s C.0.15m/s D.0.42m/s E.0.24m/s

Gemini 2.5 pro:

*Output code estimates the speed of the 'hole_chili' feature between 21s and 24.6s. It uses pre-defined normalized coordinates to calculate displacement. Assuming a board width of 2m for scale, it estimates the real speed (~ 0.141 m/s). Finally, it finds the multiple-choice option (A-E) with the closest speed value and prints its letter ('C').)

Correct!

Example 2

What is the camera's instantaneous speed around $t=12s$?

A.0.14m/s B.0.09m/s C.0.24m/s D.0.23m/s E.0.27m/s

Gemini 2.5 pro:

*The code estimates angular speed based on visual inspection (~ 18 deg/s). Assuming a camera orbital radius of 0.75m, it calculates the corresponding linear speed (~ 0.235 m/s) using $\text{speed} = \text{angular_speed} * \text{radius}$. It then compares this speed to options A-E and selects the closest one ('D').

Not accurate enough.

Wrong!

Figure 6. The simplified thought process examples of Gemini 2.5 Pro.

Its score for Speed & Acceleration is 34.3%. This indicates that while the absolute accuracy in precise estimation tasks remains a challenge, Gemini-2.5-Pro shows comparatively strong capabilities in these domains relative to other models.

By leveraging the model's reasoning process and uniformly sampling approximately 200 error records across each task type and scenario, we categorize its errors into three representative patterns. Figure 7 shows the distribution of error categories.

Inaccurate Spatial Quantification The model faces significant challenges in accurately estimating static spatial properties and relationships from visual inputs. These difficulties manifest when estimating object dimensions (size), distances between objects or between camera and objects, and absolute 3D coordinates/bounding boxes at specific time points. These errors stem from a lack of clear visual size references, difficulty distinguishing between numerically close options, and the inherent challenges of inferring metric scale from 2D pixels and estimating depth

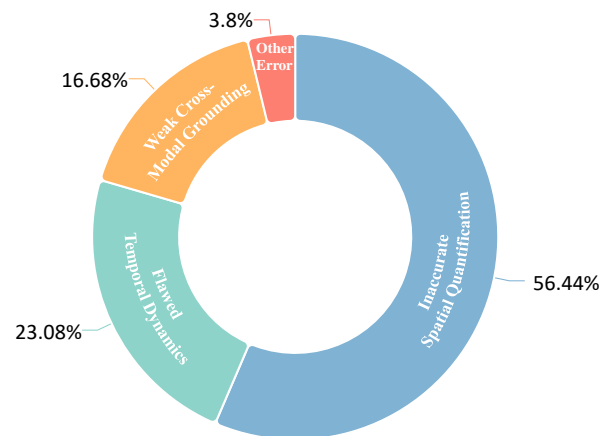


Figure 7. Distribution of error categories in Gemini-2.5-Pro across our sampled error cases.

with monocular cameras. Such limitations directly impact performance in dimensional measurement, spatial relation, and 3D video grounding tasks.

Flawed Temporal Dynamics Understanding The model struggles to perceive, track, and interpret cross-frame information that changes over time, such as motion and its dynamics. This results in erroneous calculations or descriptions of displacement, path length, speed, acceleration, directional changes (ego-centric or object pose), and overall trajectory shapes. The model particularly struggles with relative motion (distinguishing object motion from camera motion), a problem exacerbated by sparse temporal sampling. These difficulties arise from challenges in integrating information across frames, lack of internal models for physics/kinematics, inability to separate ego-motion from object motion, and information loss due to sparse sampling. These issues manifest in tasks involving displacement and path length, speed and acceleration, ego-centric orientation, trajectory description, and pose estimation (as it changes over time).

Weak Cross-Modal Grounding and Integration The model fails to properly connect textual queries/instructions with relevant spatial-temporal visual content, or to integrate provided non-visual data (such as initial poses) with visual information. This includes misinterpreting temporal constraints (like "from 1s to 18s," "at the end," "the moment of last co-occurrence"), failing to correctly utilize provided initial conditions (e.g., initial camera pose in pose estimation tasks), and incorrectly associating structured data (coordinates, timestamps) with visual elements. These errors stem from deficiencies in parsing structured/natural language instructions and difficulty integrating information from different modalities (text prompts, initial state data, video frames) into a unified reasoning process. This affects all tasks that rely on specific instructions or initial data.

These error patterns highlight that, despite Gemini-2.5-Pro's strong performance relative to other models, it still faces significant challenges in precise spatial-temporal understanding. Its limitations in quantitative estimation and complex spatial-temporal reasoning indicate that current MLLMs remain far from achieving the reliability required for embodied AI or autonomous driving applications.

5. Conclusion

We introduced STI-Bench, a comprehensive benchmark to assess MLLMs' spatial-temporal understanding through over 300 real-world videos and 2,000 QA pairs of robot desktop, indoor, and outdoor scenarios, which reveals significant limitations in current MLLMs' spatial-temporal understanding capabilities, with even the top-performing models achieving no more than 50% accuracy. Models particularly struggle with precise quantitative tasks like dimensional measurement. Our analysis identifies three key weaknesses: inaccurate spatial quantification, flawed temporal dynamics understanding, and weak cross-modal in-

tegration. These findings emphasize the substantial gap between current capabilities and the reliability needed for embodied AI and autonomous driving applications. STI-Bench provides a valuable framework for evaluating and improving MLLMs' ability to understand the physical world—essential for developing the next generation of embodied intelligent systems.

Acknowledgements

We thank the data annotation team at MolarData for their support, as well as the volunteer contributors Zixin Liu, Junjie Ruan, Lingxi Xia, Zixuan Huang, Tianrui Wan, Qingcheng Wei, Yujie Yao, Shangyang Dong, Zhuofan Zeng, and Yiming Li for their valuable work as human annotators and evaluators.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 1, 2
- [2] Anthropic. Claude 3.7 Sonnet and Claude Code. Anthropic News Announcement, 2025. Announcement on Anthropic blog, Feb 24, 2025. 6, 7
- [3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1
- [4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1, 2
- [5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yanzhi Zhu, Mingkun Yang, Zhao-hai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. 6, 7
- [6] Youssef Benchechrout, Megi Dervishi, Mark Ibrahim, Jean-Baptiste Gaya, Xavier Martinet, Grégoire Mialon, Thomas Scialom, Emmanuel Dupoux, Dieuwke Hupkes, and Pascal Vincent. Worldsense: A synthetic benchmark for grounded reasoning in large language models. *arXiv preprint arXiv:2311.15930*, 2023. 3
- [7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1, 3

- [8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 3
- [9] 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. 1, 3
- [10] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 3
- [11] 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*, pages 14455–14465, 2024. 3
- [12] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 1, 2
- [13] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025. 6, 7
- [14] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language model. *arXiv preprint arXiv:2406.01584*, 2024. 3
- [15] Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, et al. Embodiedeval: Evaluate multimodal llms as embodied agents. *arXiv preprint arXiv:2501.11858*, 2025. 1, 3
- [16] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 5
- [17] Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. *arXiv preprint arXiv:2406.05756*, 2024. 3
- [18] Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *arXiv preprint arXiv:2406.14515*, 2024. 1, 3
- [19] Chaoyou Fu, Yuhao Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 3
- [20] Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025. 1, 3
- [21] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, Yin Zhou, James Guo, Dragomir Anguelov, and Mingxing Tan. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024. 1
- [22] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabriela Levine, Wensi Ai, Benjamin Martinez, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024. 1
- [23] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2
- [24] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 1
- [25] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. In *NeurIPS 2024*, 2024. 1, 3
- [26] Bin Lin, Yang Ye, Bin Zhu, Jiayi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 1
- [27] Bin Lin, Yang Ye, Bin Zhu, Jiayi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection, 2024. 2
- [28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 2
- [29] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022. 1
- [30] OpenAI. Gpt-4o system card, 2024. 1, 6, 7
- [31] 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. *arXiv preprint arXiv:2412.07755*, 2024. 3

- [32] Yan Shu, Peitian Zhang, Zheng Liu, Minghao Qin, Junjie Zhou, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. *arXiv preprint arXiv:2409.14485*, 2024. 1
- [33] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5
- [34] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1, 2
- [35] 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. 1, 2, 6, 7
- [36] 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. 1, 2
- [37] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024. 1, 3
- [38] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2025. 1, 3
- [39] Yi Xu, Yuxin Hu, Zaiwei Zhang, Gregory P. Meyer, Siva Karthik Mustikovela, Siddhartha Srinivasa, Eric M. Wolff, and Xin Huang. Vlm-ad: End-to-end autonomous driving through vision-language model supervision. *arXiv preprint arXiv:2412.14446*, 2024. 1
- [40] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024. 1
- [41] 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
- [42] 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. *arXiv preprint arXiv:2412.14171*, 2024. 3
- [43] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025. 1, 3
- [44] Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. Rag-driver: Generalisable driving explanations with retrieval-augmented in-context learning in multi-modal large language model. *arXiv preprint arXiv:2402.10828*, 2024. 1
- [45] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 6, 7
- [46] 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. 1
- [47] Jiyao Zhang, Weiyao Huang, Bo Peng, Mingdong Wu, Fei Hu, Zijian Chen, Bo Zhao, and Hao Dong. Omnidpose: A benchmark and model for universal 6d object pose estimation and tracking. In *European Conference on Computer Vision*, pages 199–216. Springer, 2024. 2, 5
- [48] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024. 1, 3

ST-Bench: Are MLLMs Ready for Precise Spatial-Temporal World Understanding?

Supplementary Material

A. Technical Details for Benchmark Construction and Analysis

A.1. Dataset Interface

When searching for datasets, we found that apart from RGB frame data, other data formats are inconsistent. For example, we noticed that it does not provide 3D bounding box information, so we could only use it to compute camera displacement, velocity, and other related physical quantities. In contrast, Waymo includes such information, which is why we aim to align different datasets into a unified format. Therefore, we unify different types of data by converting them into frame-by-frame instance-level 3D point clouds or bounding box annotations, along with per-frame camera parameters, enabling us to obtain all the required information and process it in a consistent manner.

B. Ground Truth Computation

In ST-Bench, the ground truth annotations are derived from multi-modal data, including frame-by-frame camera intrinsics/extrinsics and precise 3D point cloud annotations. We utilize three publicly available datasets (Omni6DPose, ScanNet, Waymo) to cover different scenarios (desktop, indoor, outdoor). Below, we provide eight benchmark tasks with unified notations and formulas.

B.1. Static Understanding

B.1.1. Dimensional Measurement

Let l_x, l_y, l_z denote the dimensions (length, width, height) of an object along the x, y , and z axes:

$$\begin{aligned} l_x &= x_{\max} - x_{\min}, \\ l_y &= y_{\max} - y_{\min}, \\ l_z &= z_{\max} - z_{\min}. \end{aligned} \quad (1)$$

Here, l_x, l_y, l_z represent the object size along each coordinate axis.

If we need the distance between two objects (or between the camera and an object), let d_{12} be the Euclidean distance between their center points:

$$d_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}. \quad (2)$$

Here, (x_1, y_1, z_1) and (x_2, y_2, z_2) are the center coordinates of the two objects.

B.1.2. Spatial Relation

When the difference along one coordinate axis is significantly larger than along others, the sign of that difference determines the spatial relation:

$$\begin{aligned} r_{xy} &= \text{sign}(x_A - x_B), \\ r_{yz} &= \text{sign}(y_A - y_B), \\ r_{zx} &= \text{sign}(z_A - z_B). \end{aligned} \quad (3)$$

Here, r_{xy}, r_{yz}, r_{zx} indicate relative positioning along each axis (e.g., front/back, left/right, above/below). We choose the axis with the greatest difference to label the dominant relation.

B.1.3. 3D Video Grounding

For frame t in the camera coordinate system, the 3D bounding box of an object can be described with dimensions, center position, and optional rotations:

$$\text{BBox}_t = (l_t, w_t, h_t, x_t, y_t, z_t, \text{yaw}_t, \text{pitch}_t, \text{roll}_t). \quad (4)$$

Here, (l_t, w_t, h_t) are the object dimensions, (x_t, y_t, z_t) is the center position, and $(\text{yaw}_t, \text{pitch}_t, \text{roll}_t)$ are optional rotation angles if available.

B.2. Dynamic Understanding

B.2.1. Pose Estimation

Given the camera's initial pose (p_0, o_0) , the pose (p_t, o_t) at time t can be obtained using the extrinsic-derived matrices R_t (rotation) and T_t (translation):

$$\begin{aligned} p_t &= R_t p_0 + T_t, \\ o_t &= o_0 + \Delta o_t. \end{aligned} \quad (5)$$

Here, p_t is the position, o_t is the orientation.

B.2.2. Displacement and Path Length

Let $p_i = (x_i, y_i, z_i)$ be the position at time i . The displacement d_{0n} and path length L_{traj} are computed as:

$$d_{0n} = \sqrt{(x_n - x_0)^2 + (y_n - y_0)^2 + (z_n - z_0)^2}, \quad (6)$$

$$L_{\text{traj}} = \sum_{i=1}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}. \quad (7)$$

Here, d_{0n} is the straight-line distance from the initial to the final position; L_{traj} sums consecutive segment lengths for the entire path.

B.2.3. Speed and Acceleration

Let Δt be the time interval between consecutive frames. Then the speed v_i and acceleration a_i are:

$$\begin{aligned} v_i &= \frac{d_i}{\Delta t}, \\ a_i &= \frac{v_i - v_{i-1}}{\Delta t}. \end{aligned} \quad (8)$$

Here, d_i is the displacement between adjacent frames, v_i is the speed at time i , and a_i is the acceleration.

B.2.4. Ego-Centric Orientation

If θ_t denotes the camera orientation (azimuth) at time t , then the orientation change $\Delta\theta_t$ is:

$$\Delta\theta_t = \theta_t - \theta_0. \quad (9)$$

This indicates how much the camera has rotated relative to its initial azimuth.

B.2.5. Trajectory Description

We apply the Ramer-Douglas-Peucker (RDP) algorithm to simplify the sequence of positions into key line segments. The resulting polyline is described in a piecewise manner (e.g., “go straight for 30m, turn left 85°, then go straight for 20m, ...”), providing a concise representation of complex motion trajectories.

B.3. Human Involved Quality Control

Despite leveraging automated methods for generating large quantities of question-answer pairs, several critical issues necessitated human intervention for quality assurance. These included inaccuracies or ambiguities in automatically generated object descriptions, logical inconsistencies or irrelevant content within questions and options, and insufficient visual information in video data due to occlusions or limited resolution. To address these challenges, we established a dedicated online platform for systematic multi-round manual review and random sampling checks. Human reviewers evaluated each QA pair based on several criteria: clarity and accuracy of object descriptions, logical coherence between questions and answer options, sufficiency of visual information to accurately respond to questions, and effectiveness of distractors in posing genuine challenges. After passing the rigorous review process, answer options were randomized to prevent bias arising from fixed option ordering. Ultimately, this meticulous procedure enabled us to select over 2,000 high-quality QA pairs from more than 300 video sequences, significantly enhancing the robustness and reliability of the ST-Bench benchmark dataset.

C. Evaluation Details

The evaluation framework of ST-Bench is carefully designed to assess multimodal models across diverse real-

world scenarios, including Desktop, Indoor, and Outdoor environments. Specifically, the benchmark incorporates tasks derived from three publicly available datasets: Omni6DPose for desktop-scale, ScanNet for indoor, and Waymo for outdoor. Each task comprises question-answer pairs generated using Multimodal Large Language Models, which provide detailed semantic descriptions for objects based on video frames and corresponding multimodal data.

To ensure task diversity and challenge, each question is accompanied by carefully constructed distractors, differing significantly yet plausibly from the ground truth. Recognizing the variance in precision requirements among different scenarios—millimeter-level for Desktop, centimeter-to-decimeter-level for Indoor, and decimeter-to-meter-level for Outdoor—we apply scenario-specific scaling factors to generate distractors. Logarithmic sampling techniques are also utilized to ensure an even distribution of numerical differences, thus maintaining evaluation robustness.

Furthermore, rigorous human quality control was conducted, involving multiple rounds of manual filtering and random sampling reviews. This step ensured the logical consistency, clarity, and accuracy of all QA pairs. Finally, answer options were randomized post-review to prevent ordering bias. As a result, the final benchmark contains more than 2,000 meticulously curated QA pairs, providing a comprehensive and reliable evaluation framework across diverse real-world scenarios.

D. Visualization Results

To qualitatively illustrate the effectiveness and interpretability of ST-Bench, visualization examples from representative Desktop, Indoor, and Outdoor scenarios are presented. Visualizations include video frames with annotated bounding boxes and corresponding object descriptions generated by Multimodal Large Language Models (MLLMs). For instance, objects are annotated with clear descriptions such as “a beige minivan with a roof rack” or “a red backpack on a brown leather sofa,” which are directly visible in the annotated frames. Additionally, visual representations of object trajectories, camera movements, and bounding box projections are provided, clearly demonstrating the spatial relationships and dynamic interactions within each scenario. These visualizations not only validate the accuracy and robustness of generated annotations but also provide intuitive insights into the model’s interpretability and performance in diverse real-world settings.



Ego-Centric Orientation

From 0.0 seconds to 19.6 seconds. What is the most appropriate change in orientation of the camera from the start to the end of the video?

A:15° **B:0°** C:40° D:60° E:30°

Pose Estimation

Given the initial pose:

```
{"tx": 3389.92, "ty": 2178.7, "tz": 1.4,
  "r11": 1.0, "r12": -0.0, "r13": -0.01,
  "r21": 0.0, "r22": 1.0, "r23": -0.03,
  "r31": 0.01, "r32": 0.03, "r33": 1.0}
```

estimate and select the most likely camera pose in 3D space using only the RGB video when time=3s, ensuring it most closely matches the ground truth.

Answer:
{"tx": 3430.6, "ty": 2178.15, "tz": 1.29,
"r11": 1.0, "r12": 0.02, "r13": -0.01,
"r21": -0.02, "r22": 1.0, "r23": -0.03,
"r31": 0.01, "r32": 0.03, "r33": 1.0}

Spatial Relation

From 0.0 seconds to 19.6 seconds. What is the orientation of the camera mounted on the moving car relative to the car's forward direction?

A:Front B:Front-Left C:Front-Right
D:Left E:Right

3D Video Grounding

Given a randomly selected frame around t=17.0s, localize the 3D bounding box of a silver Honda minivan in camera coordinates.

Answer:
{"C_lwh": [4.95, 2.36, 1.81],
"C_central_point": [4.37, 3.35, -1.35],
"C_heading": -0.17}"

Displacement & Path Length

From 0.0 seconds to 19.6 seconds. What is the most likely displacement (straight-line distance) of the camera or object between two frames?

A:189.41m **B:185.53m** C:193.18m
D:177.88m E:170.24m

Dimensional Measurement

From 7.4 seconds to 18.3 seconds. What is the most appropriate length of a silver Honda minivan?

A:5.05m B:4.45m **C:4.95m** D:2.46m E:1.71m

Speed & Acceleration

From 7.3 seconds to 18.2 seconds. What is the most appropriate average speed of the silver Honda minivan over the specified time interval?

A:0.0m/s B:0.1m/s C:0.2m/s
D:0.3m/s E:0.5m/s

Trajectory Description

From 0.0 seconds to 19.6 seconds. Which of the following options most accurately describes the camera's movement trajectory?

A:move forward 185.53m
B:move forward 180.23m, then turn right 75°
C:move forward 185.53m, then turn left 75°
D:move forward 180.23m, then turn left 75°
E:move forward 190.45m, then turn right 75°

Figure 1. STI-Bench Examples (Outdoor)



Displacement & Path Length

From 0.0 seconds to 32.5 seconds. What is the most likely displacement (straight-line distance) of the camera between two frames?

A:1.16m B:0.91m C:1.03m D:0.75m E:1.31m

Pose Estimation

Given the initial pose:

```
{ "tx": 1.36 , "ty": 1.17 , "tz": 1.36 ,
  "r11": -0.4 , "r12": -0.51, "r13": 0.76,
  "r21": -0.92, "r22": 0.25 , "r23": -0.32,
  "r31": -0.03, "r32": -0.82, "r33": -0.56}
```

estimate and select the most likely camera pose in 3D space using only the RGB video when time=3s, ensuring it most closely matches the ground truth.

Answer:
 { "tx": 1.23 , "ty": 0.96 , "tz": 1.36 ,
 "r11": -0.46, "r12": -0.43, "r13": 0.78,
 "r21": -0.89, "r22": 0.23 , "r23": -0.4,
 "r31": -0.01, "r32": -0.87, "r33": -0.49}

Speed & Acceleration

From 25.5 seconds to 30.3 seconds. What is the most appropriate average speed of the camera over the specified time interval?

A:0.25m/s B:0.42m/s C:0.14m/s
 D:0.11m/s E:0.09m/s

Ego-Centric Orientation

From 0.0 seconds to 32.5 seconds. What is the most appropriate change in orientation of the camera from the start to the end of the video?

A:-83.37° B:-113.37° C:-98.37°
 D:-143.37° E:-128.37°

Dimensional Measurement

What is the most likely minimum relative distance between the black mesh office chair near a round wooden table and the round table with grey top and wooden edge around t=3.3s?

A:5.05m B:4.45m C:4.95m D:2.46m E:1.71m

3D Video Grounding

In a randomly selected frame, what is the 3D bounding box in camera coordinates for the black office chair with mesh backrests in the conference room?

Answer:
 { "C_lwh": [0.74, 0.57, 0.93],
 "C_central_point": [-1.78, 0.44, 0.84],
 "C_heading": 27.28 }

Figure 2. STI-Bench Examples (Indoor)



Pose Estimation ①

Given the initial pose at $t=0s$:
 $\{ "Tx": 0.0, "Ty": 0.0, "Tz": 0.0, "r11": 1.0, "r12": 0.0, "r13": 0.0, "r21": 0.0, "r22": 1.0, "r23": 0.0, "r31": 0.0, "r32": 0.0, "r33": 1.0 \}$
 , what is the camera pose at $t=23s$?

Answer:
 $\{ "Tx": 0.02, "Ty": 0.61, "Tz": -0.34, "r11": -0.96, "r12": 0.22, "r13": -0.15, "r21": -0.26, "r22": -0.72, "r23": 0.64, "r31": 0.03, "r32": 0.65, "r33": 0.76 \}$

Speed & Acceleration ①

What is the average speed of the camera between 4s and 13s?

A:0.13m/s B:0.39m/s C:0.29m/s
 D:0.41m/s **E:0.16m/s**

Spatial Relation ①

What is the positional relationship of the red suitcase relative to the teddy bear? ($t=0s$)

A:Left B:Right C:Front
 D:Back E:Up

3D Video Grounding ②

Given a single frame, identify the 3D bounding box localization for a small beige handbag with gold zippers. Consider dimensions, central point, and orientation in yaw, pitch, and roll. ($t=30s$)

Prompt: A small beige handbag with gold zippers.

Answer:
 $\{ "dimensions": [19.401, 20.67, 19.4], "central_point": [33.99, 17.09, 115.45], "orientation": \{ "yaw": 56.83, "pitch": 37.74, "roll": 112.36 \} \}$

Dimensional Measurement ②

What is the distance between the camera and the silver laptop with a black keyboard at the given frame? ($t=0s$)

Prompt: A silver Laptop with a black keyboard.

A:104.97cm B:110.22cm C:106.28cm
 D:103.66cm E:107.59cm

Displacement & Path Length ②

What is the camera's displacement between 15s and 24s?

A:0.45m B:0.23m C:0.37m
 D:0.68m E:0.31m

Figure 3. STI-Bench Examples (Desktop)



Speed & Acceleration

From 21.0 seconds to 24.0 seconds. What is the average speed of the camera between 21s and 24s? (unit: m/s)

MODEL OUTPUT:

```
import numpy as np
# Function to estimate feature position (normalized coordinates 0-1)
def estimate_position(frame_index, feature_name):
    # Placeholder estimations based on visual inspection
    positions = {
        21: (0.4, 0.5), # Frame at 21s
        22: (0.43, 0.47), # Frame at 22s
        23: (0.47, 0.43), # Frame at 23s
        24: (0.5, 0.4) # Frame at 24s
    }
    if feature_name == 'blue_disc_center':
        return positions.get(frame_index)
    positions_hole = {
        21: (0.8, 0.3),
        24: (0.95, 0.15)
    }
    if feature_name == 'hole_chili':
        return positions_hole.get(frame_index)
    return None

# Time points
t1 = 21.0
t2 = 24.0
delta_t = t2 - t1 # 3.0 seconds

# Estimate positions at t1 and t2 using a feature (e.g., hole below chili)
pos1_norm = estimate_position(t1, 'hole_chili')
pos2_norm = estimate_position(t2, 'hole_chili')

if pos1_norm is None or pos2_norm is None:
    chosen_option = "C" # Guessing based on rough calculations (~0.14 m/s)
else:
    # Calculate normalized displacement
    delta_x_norm = pos2_norm[0] - pos1_norm[0]
    delta_y_norm = pos2_norm[1] - pos1_norm[1]
    displacement_norm = np.sqrt(delta_x_norm**2 + delta_y_norm**2)

    # Assume the width of the board visible in the frame is W meters
    # Speed = (displacement_norm * W) / delta_t
    W = 2.0 # Assumed board width in meters
    speed_estimate = (displacement_norm * W) / delta_t

    # Find the closest speed option
    options = {'A': 0.13, 'B': 0.44, 'C': 0.15, 'D': 0.42, 'E': 0.24}
    closest_option = min(options.keys(), key=lambda option: abs(options[option] - speed_estimate))
    chosen_option = closest_option
```

Output the chosen option
print(chosen_option)

C

A 0.13m/s

B 0.44m/s

C 0.15m/s

D 0.42m/s

E 0.24m/s

Figure 4. Thinking example (Correct)



Speed & Acceleration

What is the camera's instantaneous speed around $t=12s$? (unit: m/s)

MODEL OUTPUT:

```
import numpy as np

# Function to estimate rotation angle between two frames based on object positions
# This is a simplified estimation based on visual inspection
def estimate_angle(frame1_objects, frame2_objects):
    # Example: Use positions of two objects (e.g., glasses and blue object)
    # This requires actual coordinates which we don't have, so we estimate visually.
    #
    # Visual estimation:
    # Frame 11: Glasses left, Blue object right (roughly horizontal line)
    # Frame 13: Glasses bottom-left, Blue object center-right (line rotated clockwise)
    #
    # Angle estimation: ~30-40 degrees over 2 seconds.
    # Angular speed estimate: (30 to 40) / 2 = 15 to 20 degrees/second

    angular_speed_deg_per_sec = 18 # Taking a value in the middle range
    angular_speed_rad_per_sec = np.radians(angular_speed_deg_per_sec)
    return angular_speed_rad_per_sec

# Assume table diameter and estimate camera radius
table_diameter_estimate = 0.8 # meters
table_radius_estimate = table_diameter_estimate / 2 # 0.4m

# Estimate camera distance from the center of the table based on perspective
camera_radius_estimate = 0.75 # meters (Assumption based on how much of the table is visible and perspective)

# Calculate speed = angular_speed * radius
angular_speed = estimate_angle(None, None) # Using visual estimate
speed = angular_speed * camera_radius_estimate

# speed = np.radians(18) * 0.75
# speed = 0.314 * 0.75 = 0.2355 m/s

# Compare with options
options = {'A': 0.14, 'B': 0.09, 'C': 0.24, 'D': 0.23, 'E': 0.27}
closest_option = min(options.keys(), key=lambda option: abs(options[option] - speed))

# Based on visual estimation and reasonable assumptions:
# Angular speed ~ 15-20 deg/s (0.26-0.35 rad/s)
# Radius ~ 0.7-0.9 m
# Speed range ~ (0.26*0.7) to (0.35*0.9) = 0.182 to 0.315 m/s
# Options C, D, E fall within or close to this range.
# Using 18 deg/s and 0.75m radius gives ~0.235 m/s, which is very close to D.

# Output the closest option
print(closest_option)
D
A 0.14m/s      B 0.09m/s      C 0.24m/s      D 0.23m/s      E 0.27m/s
```

Figure 5. Thinking example (Incorrect)