Metric-Bench: Exploring In-context Spatial Metric Reasoning in VLMs for Indoor Scenes

Yuling Xi[1†], Haokai Zhang[1†], Muzhi Zhu[1], Hao Zhong[1], Zongze Du[1], Hengyu Zhao[1], Chenchen Jing[2], Yufei Yin[3], Bin Qin[4], Yongjie Yang[4], Zhenbo Luo[4], Hao Chen[1⋆], and Chunhua Shen[12]

[1] State Key Lab of CAD & CG, Zhejiang University, China
[2] Zhejiang University of Technology
[3] Hangzhou Dianzi University, China
[4] Xiaomi Corporation, China
https://huggingface.co/datasets/yulingxi/Metric-Bench

Abstract. Metric reasoning is a critical and challenging task for Vision Language Models (VLMs), playing a pivotal role in embodied AI tasks such as robotic manipulation and autonomous navigation. However, current spatial reasoning remains bottlenecked by rigid pixel-level supervision; such localized optimization often compromises general multimodal intelligence, triggering performance degradation or catastrophic forgetting of broad reasoning capabilities. To address these limitations, we introduce Metric-Bench, a focused benchmark designed to guide metric-spatial reasoning using contextual information. By incorporating in-image reference objects with known physical dimensions, Metric-Bench guides models to implicitly learn the 2D-to-3D mapping without camera intrinsics. We further present MetricReasoner, a task-adapted reinforcement fine-tuning recipe for reference-grounded metric reasoning, using structured prompts and verifiable numerical rewards. Extensive experiments on Metric-Bench demonstrate that our approach significantly enhances spatial metric understanding, outperforming existing and even larger proprietary models by 43.1%, while improving downstream embodied performance over a spatial-specialized counterpart by 30.4% on RoboSpatial overall accuracy and 9.3% on ERQA, and additionally delivering consistent gains on general benchmarks (15.9% on V⋆Bench, 88.9% on BLINK), indicating that the proposed adaptation does not necessarily compromise general VLM capabilities.

Keywords: Vision language models · Spatial reasoning · Metric measurement · ScanNet

1 Introduction

Understanding metric scale, the ability to infer real-world sizes and distances from images, is a fundamental aspect of visual intelligence, yet remains challenging for current Vision Language Models (VLMs). While recent VLMs [1, 10,

⋆ Corresponding author

32, 39, 40] excel in semantic reasoning and instruction following, they often fail to make physically consistent judgments about spatial scale, such as estimating object size or distance [3, 4, 12, 28, 48]. This gap reveals a missing bridge between semantic perception and metric reasoning, limiting their utility in downstream tasks such as navigation and manipulation.

Humans, by contrast, can estimate metric quantities by referencing known objects and logically inferring spatial relationships [7, 16, 22, 23, 30]. For instance, when viewing a cup beside a laptop, we can quickly infer the cup's approximate height using the known scale of the laptop and perspective cues such as relative position and occlusion. This reasoning process does not rely on memorized numeric facts, but on contextual comparisons and geometric reasoning. Such observations raise a key question: **do VLMs possess a comparable ability to infer metric scale from contextual references, rather than relying solely on rote memorization?**

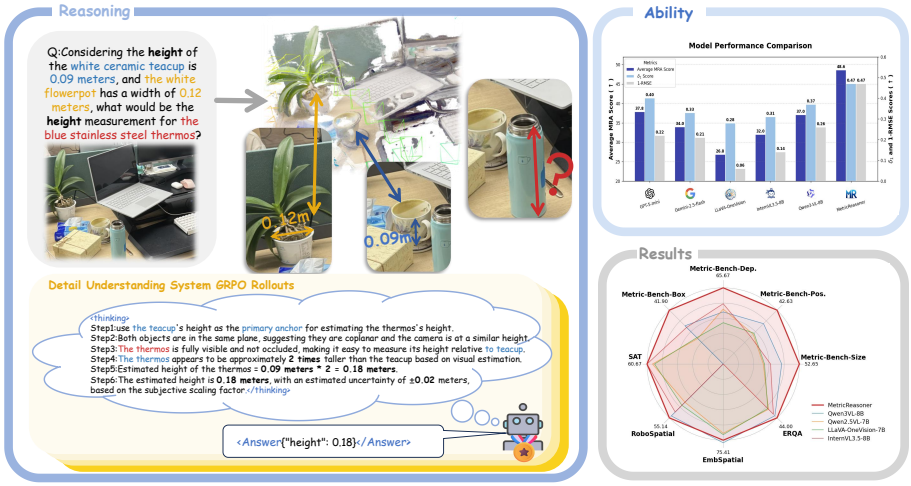


Fig. 1: We propose Metric-Bench, a benchmark inspired by human perception of 3D space: given the true physical measurements of a few reference objects, the model learns to reason about relative positions and perspective relationships in 3D space, thereby inferring precise metric measurements of other objects in the scene. Furthermore, building upon this benchmark, we employ a spatial-guided reinforcement fine-tuning strategy, without compromising the model's generalizable reasoning capability, and achieves state-of-the-art performance in metric scale perception, surpassing existing multimodal large language models.

To answer this question, this work introduces **Metric-Bench**, a new benchmark specifically designed to evaluate metric-scale reasoning in multimodal models. Built upon the ScanNetV2 [11] dataset, Metric-Bench includes 1340 carefully curated question–answer pairs that span five major spatial categories—size, po-

sition, depth, distance and 3D bounding box. Each question provides multiple reference objects with a known metric and asks the model to infer another object's absolute metric quantity. Through this setup, Metric-Bench systematically tests whether a model can reason about real-world spatial metrics using in-image contextual cues rather than memorized priors.

Beyond evaluation, we further train the **MetricReasoner** model, using the proposed Reinforcement Fine-Tuning (RFT) framework which replaces supervised numerical regression with reward-based optimization over metric accuracy and structured outputs. Instead of optimizing for answer accuracy alone, we design reward functions that encourage logically coherent, physically consistent reasoning steps grounded in reference metrics. The supervision allows VLMs to learn structured spatial reasoning strategies—integrating geometric consistency, perspective cues, and contextual priors—thereby improving both interpretability and accuracy in metric reasoning tasks.

Our main contributions are as follows:

- We introduce Metric-Bench, the first benchmark tailored to evaluate *spatial metric reasoning* in multimodal large models—i.e., inferring absolute metric quantities from *sparse* object–metric references—without requiring access to camera intrinsics.
- We introduce a metric-guided RFT recipe tailored to reference-based spatial metric reasoning. By leveraging contextual reference metrics and Chain-of-Thought (CoT) reasoning, our method unlocks the latent spatial reasoning potential of existing vision–language models.
- Extensive experiments, including zero-shot and reinforcement fine-tuning (RFT) settings, demonstrate significant improvements in metric understanding while preserving general inference capabilities. We hope our analysis provides insights for advancing physically grounded and interpretable multimodal reasoning.

2 Related Work

2.1 Metric-Scale Reasoning

Estimating metric depth and recovering spatial object geometric attributes from monocular images are long-standing challenges in computer vision, primarily due to the inherent scale ambiguity. Traditional approaches address this by incorporating semantic priors [43, 46, 51], leveraging the rich prior knowledge embedded in extensively pre-trained 2D foundation models to compensate for this scarcity, while Metric3D series [17, 50] utilize the camera intrinsics to normalize images and labels so that the model could effectively learn a unified metric scale of the world even when mixing with data from different types of cameras. To remove the need of camera intrinsics, recent works [2, 41, 42] designed specific architectures to address camera ambiguity by aligning with affine-invariant point cloud, along with scale optimization to calculate final metric scale. Despite their progress, these methods often overlook the human capacity to infer scale via in-context

reference objects. In this work, we emulate this human-centric perceptive process, enabling models to resolve scale ambiguity by leveraging contextual scale priors rather than relying solely on rigid camera parameters.

2.2 Spatial and Metric Benchmarks for VLMs

Evaluating the spatial intelligence of Vision Language Models (VLMs) has gained significant traction. Recent benchmarks and methods [13, 20, 29, 38, 45, 48] have incorporated various aspects of metric reasoning into their evaluation protocols. For instance, 3DSR-Bench [29] and OmniSpatial [20] design image-based QA tasks that probe metric understanding, while VSI-Bench [48] extends this evaluation to video-based metric estimation scenarios. Concurrent efforts aim at real-world applications. ERQA [15] systematically investigates trajectory reasoning, action estimation and task reasoning grounded in robotics scenarios, and RoboSpatial [38] further assesses spatial awareness within the context of robotics and dynamic environments. However, since inferring 3D metrics from 2D observations is inherently ill-posed, these benchmarks predominantly adopt multiple-choice or relative comparison, thus such qualitative assessments allow VLMs to bypass rigorous geometric reasoning. Existing benchmarks only support coarse relative judgments, while precise 3D metric reasoning remains underexplored. In contrast, our benchmark enables direct and accurate metric prediction by leveraging in-context visual cues and implicit object scale priors, which poses a more challenging and practical test for the spatial intelligence.

2.3 Learning Paradigms for Spatial Intelligence

A common approach to endow VLMs with spatial capabilities is to introduce dense pixel-level supervision or explicit numerical regression objectives [5, 9, 25, 31, 52]. While effective on the targeted spatial tasks, such objective design can bias optimization toward local cues and task-specific outputs, which may undermine broader multimodal reasoning and sometimes manifests as transfer degradation or catastrophic forgetting. To reduce this trade-off, recent works explore Chain-of-Thought (CoT) prompting and reinforcement learning to elicit intermediate reasoning and internalize multi-step logic via reward-driven learning [8, 19]. Notably, RFT-based methods [18, 21, 27, 36] have shown improvements in multi-step reasoning, but they typically employ outcome-level rewards or coarse spatial constraints, providing limited supervision over *metric consistency* along the reasoning trajectory. Different from these efforts, we use task-specific verifiable rewards to evaluate output validity and numerical proximity, while structured prompts encourage the model to use reference metrics as contextual evidence.

3 Benchmark Construction Pipeline

We introduce **Metric-Bench**, a benchmark specifically designed to evaluate spatial metric understanding capabilities of vision-language models, and primarily

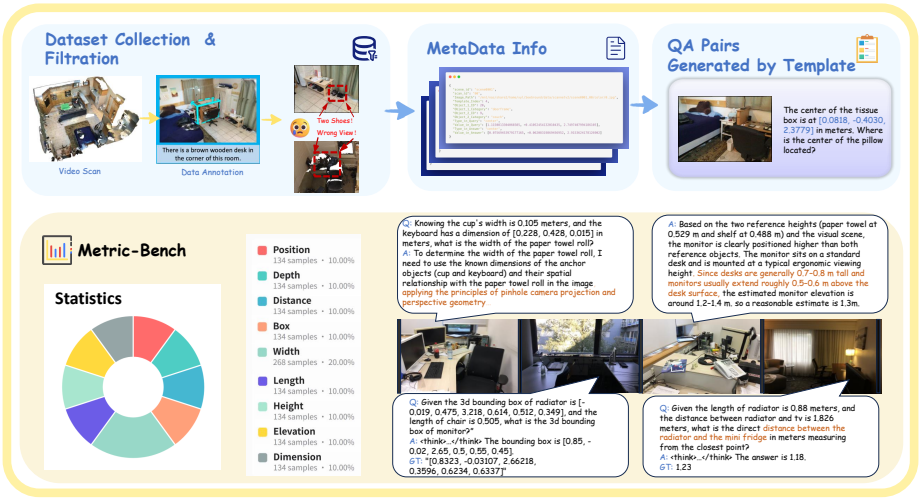


Fig. 2: Data construction pipeline and statistics in Metric-Bench.

focuses on the absolute sizes and positions of instances in indoor scenes. Metric-Bench currently comprises 1340 carefully curated question–answer pairs, each pair contains multiple real metrics of reference instances and requires the metric of a target object in the scene, which enables models to perform deep 3D spatial reasoning about metric scale based on relative positional relationships between instances in the projected image plane.

3.1 Data Collection

Metric-Bench is constructed from ScanNetV2 [11], a real-world indoor RGB-D dataset that provides per-frame camera intrinsics/extrinsics and 3D instance annotations. We focus on *absolute metric* spatial reasoning grounded in natural ego-view observations. Accordingly, spatial measurements in Metric-Bench are organized into five categories: *size*, *position*, *depth*, *distance*, and *3D bounding box*. The benchmark is formulated as question–answer pairs, where each question specifies multiple *reference metrics* and asks the model to infer a *target metric* under the same camera view. The overall construction pipeline and dataset statistics are summarized in Fig. 2.

While ScanNetV2 annotates 3D bounding boxes in the world coordinate system, human perception and vision-based measurement are naturally performed in the *ego* (camera) coordinate system and its projection onto the image plane. To align the benchmark with this observation process and to make metric reasoning perspective-aware, we convert the original 3D annotations from the world coordinates to the camera coordinates using the provided camera pose. In the camera coordinate system, (x, y, z) denotes the object center location, and the length/width/height are defined along the object's longest, shortest, and vertical

axes, respectively. This transformation standardizes metric definitions under a single view and ensures consistency with human-like size and distance estimation.

Metric-Bench is constructed in three steps:

Step 1: Filtering images and annotations. ScanNetV2 scenes are captured as videos, where motion blur can significantly degrade both object visibility and measurement reliability. We therefore remove blurry frames by computing the Variance of Laplacian [35], and retain only sufficiently sharp images. For each remaining image, we project the 3D object centers and extents (in camera coordinates) onto the image plane using camera intrinsics, and keep only objects that are visible in the field of view. This yields, for each selected image i, a set of valid 3D bounding boxes aligned with the corresponding ego-view observation.

Step 2: Query generation. For each filtered image, we extract the ground-truth metric attributes from its valid 3D bounding boxes and instantiate predefined question–answer templates with object and metric identifiers from the metadata. For example, *"If the {object$_1$} has a length of {length$_1$} meters, the {object$_2$} has a height of {height$_2$} meters, and the {object$_3$} has a width of {width$_3$} meters, what is the length of the {object$_4$}?"* The answer is {length$_4$}. To ensure diversity in object categories and spatial configurations, we require each image to contain at least four instances from distinct categories.

The templates cover three types: (i) *size-to-size inference*, which estimates a target object's size from a reference object's known size; (ii) *position/distance inference*, which deduces target position or distance from spatial cues of reference objects; (iii) *hybrid reasoning*, which jointly combines size and positional/distance cues. By varying object metrics and spatial layouts, the resulting queries evaluate whether a model can robustly reason about absolute metric quantities under relative spatial context and perspective constraints. Detailed template designs are provided in Sec.9 of the supplementary material.

Step 3: Human verification. We apply a dual verification protocol combining an automatic VLM-based checker with manual inspection to remove queries with invisible targets, duplicate questions, or ambiguous cases (e.g., multiple plausible targets). This step ensures accuracy, consistency, and minimal ambiguity of the benchmark annotations.

Following the above pipeline, we generate two isolated subsets: a 5K-sample training set (1,340 images) and a 1,340-sample test set (134 images). Importantly, the two subsets are drawn from *non-overlapping scenes* to prevent data leakage. All samples are curated for camera-view metric measurement tasks under a unified, standardized construction procedure.

4 Metric-guided Fine-tuning Strategy

To address the current limitations of VLMs in accurate metric reasoning, we develop a sparse reference-based spatial model through a systematic enhancement approach, integrating the proposed Metric-Bench with a reinforcement learning(RL) fine-tuning strategy specifically designed for in-context spatial-metric

understanding. The fine-tuning strategy explicitly trains the model to reason about other objects in a scene relative to reference object metrics, enabling the model to learn a more unified and accurate representation of spatial metric relationships and thereby enabling robust metric-aware spatial reasoning in the absence of camera intrinsics.

To better elicit the inherent scene understanding and reasoning capabilities of VLMs, we design a specialized prompt and verifiable rewards for 3D metric reasoning. The prompt explicitly guides the model to generate a CoT by leveraging the given metrics of reference objects, thereby enabling step-by-step inference of scene-metric information. The verifiable rewards function contains three parts, instructing the model to learn the referring metric-guided thinking process and the metrics of target objects.

Format Reward To enforce structural compliance in model outputs, we introduce a format reward that evaluates whether the model encapsulates its reasoning process and final answer within the designated delimiters, *e.g.*, `<thinking>...</thinking><Final answer>...</answer>`.

$$R_{format} = \begin{cases} 1, & \text{if } \hat{y} \text{ matches format,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $\hat{y}$ is the prediction of target metric value.

To enable the model to generate accurate metric predictions, we introduce two complementary numerical reward functions.

Exponential Precision(EP) reward The exponential precision reward continuously evaluates the prediction accuracy using an exponential decay over the absolute distance error:

$$R_{EP} = e^{-|\hat{y}-y|}, \tag{2}$$

where y is the ground-truth of the target metric. Under the exponential precision reward scheme, the reward value diminishes exponentially with increasing absolute error—i.e., larger deviations from the ground-truth result in substantially lower rewards.

Binned Proximity(BP) reward Furthermore, a binned proximity reward discretizes the deviation between prediction and ground-truth into hierarchical intervals to provide coarse-grained supervision:

$$R_{BP} = \begin{cases} 1, & \text{if } 0 \leq |\hat{y}-y| < 0.25, \\ 0.05, & \text{if } 0.25 \leq |\hat{y}-y| < 0.5, \\ 0.01, & \text{if } 0.5 \leq |\hat{y}-y| < 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Optimization The final reward R_{total} is computed as a sum of the format, precision, and proximity rewards:

$$R_{total} = R_{format} + R_{EP} + R_{BP}, \tag{4}$$

The combined reward encourages the model to leverage contextual guidance to reason about spatial relationships, and produce metric-aware final outputs.

The reinforcement fine-tuning strategy builds on Group Relative Policy Optimization (GRPO) [37]. Specifically, for each query q, GRPO samples a group of outputs $\{o_1, o_2, \ldots, o_G\}$ from the old policy $\pi_{\theta_{old}}$ and optimizes the following objective:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(O|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\rho_i A_i, \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)A_i\right) - \beta D_{KL}(\pi_\theta||\pi_{ref})\right)\right], \tag{5}$$

where $\rho_i = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}$ is the probability ratio, and the advantage A_i is computed by normalizing the total rewards within the group:

$$A_i = \frac{R_{total,i} - \text{mean}(\{R_{total,1}, \ldots, R_{total,G}\})}{\text{std}(\{R_{total,1}, \ldots, R_{total,G}\}) + \delta}. \tag{6}$$

5 Experiments

5.1 Experiment Setup

Baselines and Metrics We comprehensively evaluate 15 VLMs that include a diverse set of models spanning different architectures and parameter scales under zero-shot settings: (1) Proprietary models: GPT-5-mini [32] and Gemini-2.5-Flash [10]. (2) Open-source models: InternVL3.5 series [44], Qwen-VL series including Qwen2.5-VL [1] and Qwen3-VL [39], and LLaVA-OneVision [24] models. (3)Spatial-specialized model: VST [49] models and SpaceR [33]. Human evaluation results on Metric-Bench are also reported for benchmarking and comparative reference.

For evaluation metrics, we follow VSI-Bench [48] and use MRA (Mean Relative Accuracy) as our main metric. Besides, RMSE (Root Mean Squared Error) and δ_1 are also considered for a more intuitive performance comparison. For each prediction and ground-truth pair, the MRA is defined as

$$MRA = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(\frac{|\hat{y} - y|}{y} \leq 1 - \theta_i\right), \tag{7}$$

where $\theta_i = \{0.5, 0.55, ..., 0.95\}$, and $n = 10$.

The RMSE is defined following the depth estimation task, *i.e.*,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{8}$$

where N is the number of all test queries. The δ_1 metric is defined as,

$$\delta_1 = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\max\left(\frac{y_i}{\hat{y}_i}, \frac{\hat{y}_i}{y_i}\right) < 1.25\right). \tag{9}$$

Table 1: The metric estimation performance of public and our fine-tuned models on Metric-Bench. All public models are evaluated under zero-shot settings. Proprietary models and human performance are evaluated using a mini dataset, which contains 400 queries randomly sampled from the bench. Pos., Dep., Dis. are abbreviations for position, depth and distance. Length, width, height, elevation and dimension are all included in Size. The unit of RMSE is meters.

	MRA↑						RMSE↓	δ_1 ↑
	Avg.	Size	Pos.	Dep.	Dis.	Box		
Human	76.83	81.51	77.76	76.33	79.35	82.09	0.2671	0.7920
Proprietary Models (API)-Mini								
Gemini-2.5-flash	33.95	34.74	31.96	61.64	18.87	29.09	0.7886	0.3254
GPT-5-mini	37.84	38.47	33.06	58.88	**22.73**	<u>33.40</u>	0.7842	0.4014
Open-source Models								
LLaVA-OneVision-Qwen2-0.5B	20.65	23.77	23.25	7.99	14.05	25.49	1.2750	0.2105
LLaVA-OneVision-Qwen2-7B	26.81	28.55	24.38	35.62	16.15	20.52	0.9367	0.2763
InternVL3.5-1B	23.64	20.87	32.30	41.71	15.13	23.33	1.0776	0.2581
InternVL3.5-4B	33.72	35.96	26.50	49.93	19.48	24.63	0.8256	0.3298
InternVL3.5-8B	32.00	32.72	21.71	51.97	21.46	28.47	0.8637	0.3116
InternVL3.5-14B	31.49	37.85	16.75	24.56	17.98	28.31	0.9400	0.3135
Qwen2.5-VL-7B	27.58	27.75	23.66	47.34	16.71	19.84	0.7404	0.2577
Qwen2.5-VL-32B	32.71	33.82	24.93	50.98	20.45	28.07	0.8647	0.3196
Qwen3-VL-8B	37.04	40.44	31.93	44.91	21.14	29.73	0.7394	0.3695
Qwen3-VL-32B	39.24	42.11	<u>34.74</u>	49.80	<u>22.28</u>	31.83	0.7331	0.3836
Spatial-specialized Models								
VST-7B-RL	38.11	40.76	27.63	<u>55.43</u>	17.44	26.52	<u>0.7210</u>	0.4039
VST-7B-SFT	<u>39.61</u>	<u>42.23</u>	28.58	53.84	19.63	26.86	0.7463	<u>0.4058</u>
SpaceR-7B	28.97	27.66	23.97	52.49	22.15	24.11	0.8983	0.2947
MetricReasoner	**48.59**	**52.65**	**42.63**	**65.67**	19.81	**41.90**	**0.5323**	**0.4705**

Implementation Details To evaluate the relative efficacy of different fine-tuning paradigms, we apply both supervised fine-tuning (SFT) and reinforcement fine-tuning (RFT) to the Qwen3-VL-8B base model. MetricReasoner refers to the model that employs the RFT strategy. Both RFT and SFT Models are trained for 3 epochs to ensure full convergence and optimal performance. All experiments are implemented on 4 H100 GPUs with batch size of 128, group size of 32, and learning rate of 2×10^{-6}. Training for 3 epochs takes approximately 12 hours. We perform the scene-level split before QA generation. The training split used both in RFT and SFT, is constructed from ScanNetV2 training scenes, while Metric-Bench test samples are drawn from non-overlapping test scenes. Therefore, no scene, image, or object instance from Metric-Bench is used during RFT.

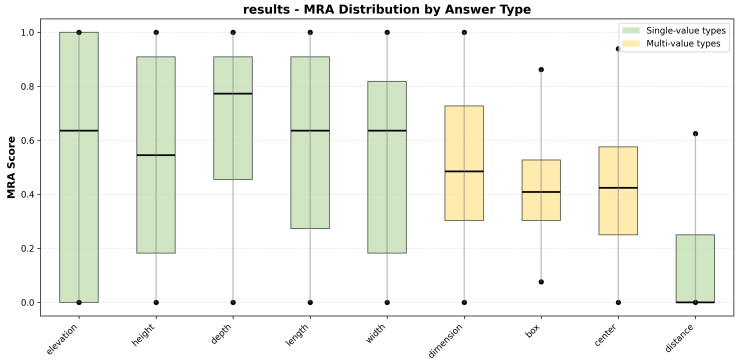


Fig. 3: Performance of MetricReasoner across all categories in Metric-Bench.

5.2 Main Results

Table 1 presents a comprehensive comparison of different models across multiple metrics on Metric-Bench. The Size denotes the average of MRA in dimensions, *i.e.*, width, length, height, elevation and dimension. Overall, proprietary models outperform most open-source counterparts, while spatial-specialized achieves the highest MRA score of 39.61, notably surpassing Gemini-2.5-flash (33.95), GPT-5-mini (37.84) and also exhibits better performance in both RMSE and δ_1. For open-source models, the InternVL series demonstrates relatively stable results, where InternVL3.5-4B shows competitive performance among all InternVL series. Meanwhile, Qwen3-VL-32B delivers the best results among the evaluated open-source general-purpose models, reaching 39.24 in MRA and 0.3836 in δ_1. In comparison, our proposed MetricReasoner significantly surpasses all baseline models. MetricReasoner, representing metric-guided RFT strategies, achieves the best MRA scores of 48.59, with δ_1 increasing to 0.4705, while substantially reducing RMSE to 0.5323. These results clearly demonstrate the superiority of our method in multi-dimensional spatial metric understanding and depth consistency modeling. Besides, it is worth noting that among all categories, most models perform better on depth estimation than on other metric items, with distance estimation yielding the poorest performance, which leaves challenges for further improvement on these metrics.

The detailed performance of our MetricReasoner is illustrated in Fig. 3, which indicates that models generally perform better on single-value categories, with more consistent and reliable MRA scores. Multi-value categories present a greater challenge, reflecting a higher degree of variability and lower prediction accuracy. Among the single-value categories, the score range for the distance metric is comparable to that of other attributes, but its median is lower, reflecting the model's performance shortcomings on this category.

Table 2: The performance of various amounts of referenced metrics.

$N_{ref.}$	MRA↑	RMSE↓	δ_1 ↑
0	47.56	0.5540	0.4586
2	48.26	0.5392	**0.4852**
4	**48.59**	**0.5323**	0.4705
5	46.85	0.5604	0.4382

Table 3: Ablation on two accuracy rewards BP and EP.

EP	BP	MRA↑	RMSE↓	δ_1 ↑
×	×	12.38	1.0515	0.0862
✓	×	45.95	0.5798	0.4281
×	✓	48.05	0.5587	0.4537
✓	✓	**48.59**	**0.5323**	**0.4705**

5.3 Ablation Study

Ablation of the number of referenced metrics Table.2 studies how the number of referenced metrics affects MetricReasoner. Without any reference, the model lacks an explicit calibration anchor and yields relatively low accuracy (MRA=47.56) and higher error (RMSE=0.5540). Introducing a small set of references consistently improves performance: $N_{ref.}$=2 increases δ_1 to 0.4852 and reduces RMSE to 0.5392, while $N_{ref.}$=4 achieves the best overall trade-off, giving the highest MRA (48.59) and the lowest RMSE (0.5323). However, further increasing the references to 5 leads to a clear degradation across all metrics, suggesting diminishing returns and potential information overload that hinder stable calibration. Overall, a moderate number of referenced metrics (e.g., 4) is optimal, and more references are not necessarily better.

Ablation of fine-tuning rewards Table 3 ablates the two accuracy rewards, EP and BP. Removing both rewards severely hurts performance, indicating that accuracy-oriented reinforcement is crucial for reliable metric prediction. Enabling either reward alone yields substantial gains: EP-only improves MRA to 45.95 and reduces RMSE to 0.5798, while BP-only further boosts MRA to 48.05 and δ_1 to 0.4537 with a comparable RMSE (0.5587), suggesting binned proximity reward provides a stronger supervision signal for correct metric calibration. Combining EP and BP leads to additive improvements. Overall, BP appears to be the dominant contributor, and a careful reward balancing strategy may be required to fully benefit from the combination.

5.4 Performance on Out-of-domain Spatial Benchmarks

To verify the preservation of the spatial reasoning capability of MetricReasoner, we evaluate our model on several spatial understanding benchmarks. Table 4 evaluates downstream embodied benchmarks on ERQA [15] and RoboSpatial [38]. Overall, MetricReasoner consistently delivers the strongest transfer performance, outperforming both the general VLM baseline and spatial-specialized models. On ERQA, MetricReasoner achieves the best accuracy (44.00), surpassing Qwen3-VL-8B (42.25) and the spatial-focused VST-7B-RL (40.25). On RoboSpatial, MetricReasoner further sets a new high in overall accuracy (55.14) and improves key sub-dimensions, notably Compatibility (56.19) and Configuration (83.74). In contrast, spatial-specialized models exhibit weaker embodied transfer,

Table 4: General performance of spatial-specialized model VST-7B-RL, our MetricReasoner and corresponding base model Qwen3-VL-8B on ERQA, RoboSpatial, and general benchmarks, including CountBench, V⋆ Bench and BLINK.

Models	ERQA ↑	RoboSpatial↑	CountBench↑	V⋆ Bench↑	BLINK↑
Qwen3-VL-8B	42.25	53.14	0.917	61.78%	58.61%
VST-7B-RL	40.25	42.29	0.891	59.16%	43.32%
MetricReasoner	**44.00**	**55.14**	**0.920**	**68.59%**	**81.82%**

Table 5: Results on RoboSpatial, CountBench, V⋆ Bench, and BLINK of different fine-tuning strategies. RL w/ COT&SFT denotes SFT on CoT-annotated data followed by reinforcement fine-tuning.

Strategy	RoboSpatial	CountBench	V⋆ Bench	BLINK
RL(MetricReasoner)	**55.14**	**0.9204**	**68.59%**	**81.82%**
RL w/ COT&SFT	45.14	0.9190	62.83%	34.53%
SFT	40.86	0.9093	63.35%	51.26%

with VST-7B-RL trailing substantially in RoboSpatial Acc. (42.29) and particularly struggling on Context (1.63). These results indicate that MetricReasoner learns spatial reasoning signals that generalize effectively to embodied decision-centric evaluations, yielding robust gains beyond both the base model and prior spatial-specialized alternatives.

For more detailed performance of spatial benchmarks, please refer to supplementary materials.

5.5 Performance on General Purpose Benchmarks

Table 4 also reports results of three models on three general purpose benchmarks *i.e.*, CountBench [34], V⋆ Bench [47], and BLINK [14]). In general-purpose evaluation, VST-7B-RL lags behind the strong general VLM baseline Qwen3-VL-8B on all three benchmarks (e.g., 59.16% vs. 61.78% on V⋆ Bench and 43.32% vs. 58.61% on BLINK), suggesting that VST-7B-RL exhibits limited performance on broad visual–language capabilities. In contrast, MetricReasoner not only maintains general understanding ability but also improves it: it achieves the best performance across the board, reaching 0.920 on CountBench, 68.59% on V⋆ Bench, and 81.82% on BLINK. These results indicate that our method enhances spatial reasoning without sacrificing generalization, and even delivers consistent gains over the general baseline, demonstrating robust transfer beyond the targeted spatial benchmarks.

5.6 Comparison between SFT and RFT strategies

Table 5 shows the influence of different fine-tuning strategies on the MetricBench. RL w/ COT&SFT represents a two-stage fine-tuning. On stage1, we

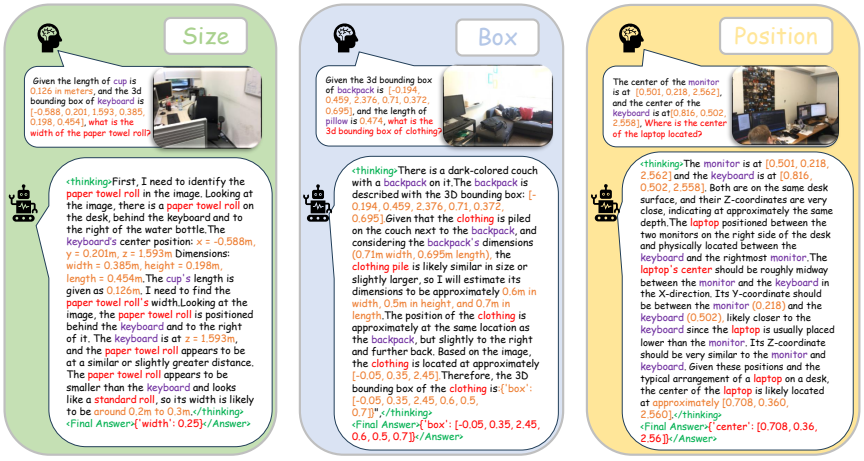


Fig. 4: Qualitative results of MetricReasoner.

use the CoT data generated by Gemini-2.5 [10] to fine-tune the base model, then our RFT strategy is employed on stage2. The results indicate that training strategies critically impact both spatial reasoning and general benchmark transfer: RL fine-tuning only(MetricReasoner) achieves the best overall performance, suggesting that reward-driven optimization yields spatially grounded behaviors that generalize well. In contrast, the RL w/ COT&SFT fine-tuning degrades RoboSpatial to 45.14 and V⋆ to 62.83%, implying that CoT-style supervision may introduce patterns misaligned with the downstream reward objective and is not fully corrected by subsequent RFT. Pure SFT with CoT data performs worst on RoboSpatial (40.86) and trails RL on all general benchmarks, suggesting that supervised fine-tuning alone is less effective at inducing robust, transferable spatial reasoning than direct RL optimization.

5.7 Qualitative results of MetricReasoner

To provide an in-depth understanding of the spatial reasoning capabilities of our MetricReasoner model, we present qualitative results that showcase the model's CoT process in inferring metric quantities from in-image contextual cues. Fig. 4 illustrates three representative cases across different metric categories: Size, Box, and Position. These examples demonstrate how the model integrates visual evidence with metric anchors and perspective cues to arrive at a physically consistent metric prediction, moving beyond mere semantic recall.

Size Estimation. The visualization refers to size estimation. The model is asked to estimate the width of a paper towel roll, using nearby objects such as the keyboard and cup as contextual references. The model's CoT first identifies the relative position of the target object and referring objects, as the paper

towel roll located on the desk, behind the keyboard and to the right of the water bottle. It then establishes metric cues from the scene, including the keyboard's 3D position and dimensions, as well as the cup's given length (0.126 m). Based on the relative visual scale, the model observes that the paper towel roll is smaller than the keyboard but comparable to a standard desktop object. The reasoning process is reasonable: it first localizes the target object, then anchors the estimation with nearby objects of known size, and finally combines relative comparison with object-size prior knowledge to infer that the paper towel roll's width is likely around $0.2 \sim 0.3$ m. This demonstrates the model's ability to leverage both in-image metric references and commonsense object priors for size estimation.

3D Bounding Box Inference. The second case illustrates the model's ability to infer the 3D bounding box of an unseen target object from nearby metric references. The target is a pile of clothing placed on a dark-colored couch, next to a backpack with a known 3D bounding box $[-0.194, 0.459, 2.376, 0.71, 0.372, 0.695]$. The model first localizes the clothing pile relative to the backpack, then uses the backpack's dimensions as a scale anchor to estimate the target size. Since the clothing pile is visually close to the backpack and appears slightly more spread out, the model predicts its dimensions as approximately 0.6 m in width, 0.5 m in height, and 0.7 m in length. It further infers that the clothing pile is located slightly to the right and farther back from the backpack, resulting in the predicted box $[-0.05, 0.35, 2.45, 0.6, 0.5, 0.7]$. This example shows that the model can combine spatial relations, nearby 3D anchors, and object-level priors to produce a physically plausible 3D box estimation.

3D Position Inference. The final example (Position) asks for the center position of a laptop, referencing the known centers of a monitor and a keyboard. The model leverages these two anchors to establish the local desk layout and infer the laptop's position within the same 3D coordinate frame. The CoT first observes that the monitor and keyboard have very similar Z-coordinates, indicating that they are placed at approximately the same depth on the desk surface. It then translates the 2D spatial relationship of the laptop—located between the keyboard and the rightmost monitor—into 3D coordinates by interpolating its X- and Y-positions between the two anchors while keeping a similar Z value. This leads to the estimated laptop center $[0.708, 0.360, 2.560]$. This example demonstrates the model's ability to combine co-planarity, relative layout, and metric anchors to infer plausible 3D object positions from image observations.

These qualitative results validate our core hypothesis: MetricReasoner, by imposing structured reasoning guidance, encourages VLMs to transition from prior memorization to a structured, contextual metric reasoning strategy. The resulting CoT is both interpretable and physically grounded, making the model's spatial inferences more reliable. For more visualization and analysis on downstream benchmarks, please refer to supplementary materials.

6 Conclusion

Inspired by how humans perceive 3D space through relative positioning, we introduce Metric-Bench, a benchmark designed to evaluate vision-language models' ability to infer the absolute spatial metric of a scene from a single image, given only sparse object-metric references. This setup eliminates reliance on camera intrinsics, thereby enabling robust adaptation across diverse real-world scenarios and generic imaging configurations. Our analysis reveals that current VLMs excel notably in estimating metric depth compared to other metric dimensions (*e.g.*, width or height), yet exhibit pronounced weaknesses in predicting metric distances between objects. On Metric-Bench, we demonstrate that implicitly supervising Chain-of-Thought (CoT) generation via a verifiable reward function substantially enhances spatial-metric reasoning. Our approach not only surpasses existing large proprietary models in metric estimation accuracy but also preserves strong generalization on downstream spatial understanding and manipulation tasks, without compromising the general reasoning capabilities.

Future avenues of improvement include in-context absolute metric learning from video inputs and task-specific fine-tuning strategies tailored to downstream spatial reasoning applications.

7 Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62576315, No. 62506338)

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025) 1, 8
2. Bochkovskii, A., Delaunoy, A., Germain, H., Santos, M., Zhou, Y., Richter, S.R., Koltun, V.: Depth pro: Sharp monocular metric depth in less than a second. In: International Conference on Learning Representations (2025), https://arxiv.org/abs/2410.02073 3
3. Cai, Z., Yeh, C.F., Xu, H., Liu, Z., Meyer, G., Lei, X., Zhao, C., Li, S.W., Chandra, V., Shi, Y.: Depthlm: Metric depth from vision language models. arXiv preprint arXiv:2509.25413 (2025) 2
4. Cai, Z., Wang, Y., Sun, Q., Wang, R., Gu, C., Yin, W., Lin, Z., Yang, Z., Wei, C., Shi, X., et al.: Has gpt-5 achieved spatial intelligence? an empirical study. arXiv preprint arXiv:2508.13142 (2025) 2
5. Chen, B., Xu, Z., Kirmani, S., Ichter, B., Sadigh, D., Guibas, L., Xia, F.: Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In: CVPR (2024) 4
6. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv preprint arXiv:2506.18088 (2025) 22

7. Chen, Y.C., Deza, A., Konkle, T.: How big should this object be? perceptual influences on viewing-size preferences. Cognition **225**, 105114 (2022) 2
8. Chen, Y., Qi, Z., Zhang, W., Jin, X., Zhang, L., Liu, P.: Reasoning in space via grounding in the world. arXiv preprint arXiv:2510.13800 (2025) 4
9. Cheng, A.C., Fu, Y., Chen, Y., Liu, Z., Li, X., Radhakrishnan, S., Han, S., Lu, Y., Kautz, J., Molchanov, P., et al.: 3d aware region prompted vision language model. arXiv preprint arXiv:2509.13317 (2025) 4
10. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., 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) 1, 8, 13
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017) 2, 5
12. Daxberger, E., Wenzel, N., Griffiths, D., Gang, H., Lazarow, J., Kohavi, G., Kang, K., Eichner, M., Yang, Y., Dehghan, A., et al.: Mm-spatial Exploring 3d spatial understanding in multimodal llms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7395–7408 (2025) 2
13. Dongfang, Z., Zheng, X., Weng, Z., Lyu, Y., Paudel, D.P., Van Gool, L., Yang, K., Hu, X.: Are multimodal large language models ready for omnidirectional spatial reasoning? arXiv preprint arXiv:2505.11907 (2025) 4
14. Fu, X., Hu, Y., Li, B., Feng, Y., Wang, H., Lin, X., Roth, D., Smith, N.A., Ma, W.C., Krishna, R.: Blink: Multimodal large language models can see but not perceive. arXiv preprint arXiv:2404.12390 (2024) 12
15. Gemini Robotics Team, Abeyruwan, S., Ainslie, J., Alayrac, J.B., Arenas, M.G., Armstrong, T., Balakrishna, A., Baruch, R., Bauza, M., Blokzijl, M., et al.: Gemini robotics: Bringing AI into the physical world. arXiv preprint arXiv:2503.20020 (2025) 4, 11
16. Gogel, W.C., Da Silva, J.A.: Familiar size and the theory of off-sized perceptions. Perception & Psychophysics **41**(4), 318–328 (1987) 2
17. Hu, M., Yin, W., Zhang, C., Cai, Z., Long, X., Chen, H., Wang, K., Yu, G., Shen, C., Shen, S.: Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation (2024) 3
18. Huang, T., Zhang, Z., Tang, H.: 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. arXiv preprint arXiv:2507.23478 (2025) 4
19. Ji, B., Agrawal, S., Tang, Q., Wu, Y.: Enhancing spatial reasoning in vision-language models via chain-of-thought prompting and reinforcement learning. arXiv preprint arXiv:2507.13362 (2025) 4
20. Jia, M., Qi, Z., Zhang, S., Zhang, W., Yu, X., He, J., Wang, H., Yi, L.: Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. arXiv preprint arXiv:2506.03135 (2025) 4
21. Jin, Z., Tu, R.C., Liao, J., Sun, W., Luo, X., Liu, S., Tao, D.: Spazer: Spatial-semantic progressive reasoning agent for zero-shot 3d visual grounding. arXiv preprint arXiv:2506.21924 (2025) 4
22. Koch, E., Baig, F., Zaidi, Q.: Picture perception reveals mental geometry of 3d scene inferences. Proceedings of the National Academy of Sciences **115**(30), 7807–7812 (2018) 2
23. Konkle, T., Oliva, A.: A familiar-size stroop effect: real-world size is an automatic property of object representation. Journal of Experimental Psychology: Human Perception and Performance **38**(3), 561 (2012) 2

24. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024) 8
25. Liang, W., Sun, G., He, Y., Dong, J., Dai, S., Laptev, I., Khan, S., Cong, Y.: Pixelvla: Advancing pixel-level understanding in vision-language-action model. arXiv preprint arXiv:2511.01571 (2025) 4
26. Liao, Y.H., Mahmood, R., Fidler, S., Acuna, D.: Reasoning paths with reference objects elicit quantitative spatial reasoning in large vision-language models (2024), https://arxiv.org/abs/2409.09788 21
27. Liao, Z., Xie, Q., Zhang, Y., Kong, Z., Lu, H., Yang, Z., Deng, Z.: Improved visual-spatial reasoning via r1-zero-like training. arXiv preprint arXiv:2504.00883 (2025) 4
28. Liu, Y., Ma, M., Yu, X., Ding, P., Zhao, H., Sun, M., Huang, S., Wang, D.: Ssr: Enhancing depth perception in vision-language models via rationale-guided spatial reasoning. arXiv preprint arXiv:2505.12448 (2025) 2
29. Ma, W., Chen, H., Zhang, G., Chou, Y.C., Chen, J., de Melo, C., Yuille, A.: 3dsr-bench: A comprehensive 3d spatial reasoning benchmark. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6924–6934 (2025) 4
30. Maruya, A., Zaidi, Q.: Mental geometry of perceiving 3d size in pictures. Journal of Vision **20**(10), 4–4 (2020) 2
31. Munasinghe, S., Gani, H., Zhu, W., Cao, J., Xing, E., Khan, F.S., Khan, S.: Videoglamm: A large multimodal model for pixel-level visual grounding in videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 19036–19046 (2025) 4
32. OpenAI: Gpt-5 system card. openai.com/index/gpt-5-system-card (2025) 1, 8
33. Ouyang, K., Liu, Y., Wu, H., Liu, Y., Zhou, H., Zhou, J., Meng, F., Sun, X.: Spacer: Reinforcing mllms in video spatial reasoning. arXiv preprint arXiv:2504.01805 (2025) 8
34. Paiss, R., Ephrat, A., Tov, O., Zada, S., Mosseri, I., Irani, M., Dekel, T.: Teaching clip to count to ten. arXiv preprint arXiv:2302.12066 (2023) 12
35. Pertuz, S., Puig, D., Garcia, M.A.: Analysis of focus measure operators for shape-from-focus. Pattern Recognition **46**(5), 1415–1432 (2013) 6
36. Qi, Z., Zhang, Z., Yu, Y., Wang, J., Zhao, H.: Vln-r1: Vision-language navigation via reinforcement fine-tuning. arXiv preprint arXiv:2506.17221 (2025) 4
37. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024) 8
38. Song, C.H., Blukis, V., Tremblay, J., Tyree, S., Su, Y., Birchfield, S.: RoboSpatial: Teaching spatial understanding to 2D and 3D vision-language models for robotics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025), oral Presentation 4, 11
39. Team, Q.: Qwen3-vl Sharper vision, deeper thought, broader action. https://qwen.ai/blog?id=qwen3-vl (2025) 1, 8
40. Team, V., Hong, W., Yu, W., Gu, X., Wang, G., Gan, G., Tang, H., Cheng, J., Qi, J., Ji, J., Pan, L., Duan, S., Wang, W., Wang, Y., Cheng, Y., He, Z., Su, Z., Yang, Z., Pan, Z., Zeng, A., Wang, B., Chen, B., Shi, B., Pang, C., Zhang, C., Yin, D., Yang, F., Chen, G., Xu, J., Zhu, J., Chen, J., Chen, J., Chen, J., Lin, J., Wang, J., Chen, J., Lei, L., Gong, L., Pan, L., Liu, M., Xu, M., Zhang, M., Zheng, Q., Yang, S., Zhong, S., Huang, S., Zhao, S., Xue, S., Tu, S., Meng, S., Zhang, T., Luo, T.,

Hao, T., Tong, T., Li, W., Jia, W., Liu, X., Zhang, X., Lyu, X., Fan, X., Huang, X., Wang, Y., Xue, Y., Wang, Y., Wang, Y., An, Y., Du, Y., Shi, Y., Huang, Y., Niu, Y., Wang, Y., Yue, Y., Li, Y., Zhang, Y., Wang, Y., Wang, Y., Zhang, Y., Xue, Z., Hou, Z., Du, Z., Wang, Z., Zhang, P., Liu, D., Xu, B., Li, J., Huang, M., Dong, Y., Tang, J.: Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning (2025), https://arxiv.org/abs/2507.01006 1

41. Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5261–5271 (2025) 3
42. Wang, R., Xu, S., Dong, Y., Deng, Y., Xiang, J., Lv, Z., Sun, G., Tong, X., Yang, J.: Moge-2: Accurate monocular geometry with metric scale and sharp details (2025), https://arxiv.org/abs/2507.02546 3
43. Wang, T., Zhu, X., Pang, J., Lin, D.: FCOS3D: fully convolutional one-stage monocular 3D object detection. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 913–922 (2021) 3
44. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025) 8
45. Wang, W., Tan, R., Zhu, P., Yang, J., Yang, Z., Wang, L., Kolobov, A., Gao, J., Gong, B.: Site: towards spatial intelligence thorough evaluation. arXiv preprint arXiv:2505.05456 (2025) 4
46. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: DETR3D: 3D object detection from multi-view images via 3D-to-2D queries. pp. 180–191. PMLR (2022) 3
47. Wu, P., Xie, S.: V*: Guided visual search as a core mechanism in multimodal llms. arXiv preprint arXiv:2312.14135 (2023) 12
48. Yang, J., Yang, S., Gupta, A.W., Han, R., Fei-Fei, L., Xie, S.: Thinking in space: How multimodal large language models see, remember, and recall spaces. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10632–10643 (2025) 2, 4, 8
49. Yang, R., Zhu, Z., Li, Y., Huang, J., Yan, S., Zhou, S., Liu, Z., Li, X., Li, S., Wang, W., et al.: Visual spatial tuning. arXiv preprint arXiv:2511.05491 (2025) 8
50. Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., Shen, C.: Metric3d: Towards zero-shot metric 3d prediction from a single image. In: ICCV. pp. 9043–9053 (2023) 3
51. Zhang, H., Jiang, H., Yao, Q., Sun, Y., Zhang, R., Zhao, H., Li, H., Zhu, H., Yang, Z.: Detect anything 3d in the wild. arXiv preprint arXiv:2504.07958 (2025) 3
52. Zong, Y., Zhang, Q., An, D., Li, Z., Xu, X., Xu, L., Tu, Z., Xing, Y., Dabeer, O.: Ground-v: Teaching vlms to ground complex instructions in pixels. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 24635–24645 (2025) 4

In this supplementary material, we provide the design of prompt in Sec. 8, our query template in Sec. 9, detailed answer format in Sec. 10, and qualitative results of MetricReasoner on Metric-Bench and downstream manipulation tasks in Sec. 12 and Sec. 13.

8 Prompt Design

The prompt consists of three key components: (1) **3D Fundamentals**, foundational concepts about the pinhole imaging mechanism, the definition of camera coordinate system and the format of 3D bounding box. (2) **Reasoning Guidance**, clear instructions on how to derive metric clues from visual and textual inputs, leveraging reference objects of known metric sizes. (3) **Output Specification**, a standardized format for structured, interpretable responses. This structured design ensures consistent, traceable, and physically grounded metric reasoning.

Prompt

You are given an image along with a textual description that includes 3D metric information about certain anchor objects in the scene, represented in the camera coordinate system. Your task is to reason about the 3D spatial layout and infer the metric property (e.g., height) of a specified target object based on visual cues, spatial relations, and provided anchor metric measurement. Prerequisite knowledge:
1. **The Camera Coordinate System**: Recall that the camera coordinate system has its origin at the optical center, with:
- **Z-axis** pointing in the imaging direction (forward),
- **X-axis** parallel to the image horizontal (positive right),
- **Y-axis** parallel to the image vertical (positive downward).
All positions and dimensions are in meters.
2. **Anchor box Information**: If the 3D bounding box of the anchor object (e.g., a book) is given, the format is '[x, y, z, width, height, length]', where: '[x, y, z]' is the center position in camera coordinates, '[width, height, length]' correspond to the shorter horizontal dimension, vertical dimension, and longer horizontal dimension, respectively.
The question that need to be reasoned:
Please provide your response in two parts: **Thinking Process** and **Final Answer**.
3.**Thinking Process**: Your analysis where you explain your reasoning step-by-step, which includes but is not limited to the following aspects:
a. **Define the Reference Anchor and Its Known Physical Quantity**
b. **Establish Spatial Pose and Contextual Layout of the Anchor**
c. **Identify and Locate the Target Object in Image Space**
d. **Construct Spatial Relationship Graph (with Optional Intermediates)**

Prompt

e. **Estimate Relative Quantity Offset or Scaling Ratio**
f. **Derive Absolute Quantity via Explicit Calculation Chain**
g. **Document Assumptions, Limitations, and Uncertainty Bounds**
4. **Final Answer**: Your final numerical estimate for the target object's physical quantity (e.g., height in meters), wrapped in '`<Final answer>`' tags and formatted as a JSON object with the appropriate key.
Example of the required output format which you should follow: `<thinking>`Step-by-step reasoning following the seven aspects above, grounded in visual analysis and explicit calculation.`</thinking>` Please output the result in JSON format: `<Final answer>'length': <value></answer>`.

9 Query Template

To systematically evaluate a model's capability in metric reasoning and spatial understanding within 3D scenes, we design a set of structured query templates for automatically generating question-answer samples that cover diverse spatial attributes. Specifically, these templates are organized into three categories. The first category, Dimensions, focuses on assessing the model's ability to understand and infer geometric properties such as object length, width, height, and 3D bounding boxes. The second category, Position and Distance, examines spatial relationships including object centers in the camera coordinate system or ground reference frame, as well as depth, elevation, and relative distance. The third category, Complex and Mixed Queries, further combines geometric dimensions, positional relations, and local constraints to construct more challenging problems that require multi-step reasoning. By instantiating these templates with variables such as object identifiers, size attributes, center coordinates, bounding boxes, and pairwise relations, we can generate large-scale query samples with unified semantic structure, controllable difficulty, and broad coverage of reasoning patterns. This provides a fine-grained benchmark for evaluating a model's spatial scale perception, geometric consistency modeling, and compositional reasoning ability.

Templates

Dimensions
1. If the $object_0$ has a length of $length_0$ meters, the $object_1$ has a width of $width_1$ meters, the $object_2$ has a height of $height_2$ meters, what is the length of the $object_3$?
2. Knowing the $object_0$'s width is $width_0$ meters and the $object_1$ has a dimension of $dimension_1$ meters, while the $object_2$ has a length of $length_2$ meters, what is the width of the $object_3$?

Templates

3. Considering the height of the $object_0$ is $height_0$ meters and the $object_1$ has a width of $width_1$ meters, the $object_2$ is $depth_2$ meters from the camera, what would be the height measurement for the $object_3$?
4. If the $object_0$ has a dimension of $dimension_0$ in meters, the 3d bounding box of $object_1$ is box_1, the $object_2$ has a width of $width_2$ meters, what would be a corresponding dimension for the $object_3$?

Position and Distance

5. The center of the $object_0$ is at $center_0$ in meters in camera coordinate system, the center of the $object_1$ is at $center_1$, the $object_2$ has a bounding box box_2, Where is the center of the $object_3$ located?
6. If the $object_0$ is $depth_0$ meters away from the camera and the $object_1$ is $depth_1$ meters away, while the $object_2$ is positioned $elevation_2$ meters above the ground, how far is the $object_3$ from the same viewpoint?
7. The $object_0$ is positioned $elevation_0$ meters above the ground and the $object_1$ is positioned $elevation_1$ meters above the ground; the $object_2$ has a length of $length_2$ meters, What is the $object_3$'s elevation from the ground?

Complex and Mixed Queries

8. Given the length of $object_0$ is $length_0$ meters, the distance between the $object_0$ and the $object_1$ is $distance_1$ meters, the $object_2$ has a height of $height_2$ meters, what is the direct distance between the $object_2$ and the $object_3$ in meters measuring from the closest point?
9. Given the 3d bounding box of $object_0$ is box_0, the length of $object_1$ is $length_1$, the width of $object_2$ is $width_2$, what is the 3d bounding box of $object_3$?
10. Given the length of $object_0$ is $length_0$ in meters and the 3d bounding box of $object_1$ is box_1, the $object_2$ has a height of $height_2$ meters, what is the width of the $object_3$?

10 Answer Format

For format of answers, length, width, height, depth, elevation and distance are single-value output, while dimension, center and box output a list that contains dimensions [width, height, length], coordinates [x, y, z] and both. The distance is defined by the shortest distance between objects and height is the object height for itself, while elevation represents the altitude above the ground.

11 Performance on Q-Spatial++

Metric-Bench differs from Q-Spatial Bench [26] in both task formulation and reference-based metric grounding. Q-Spatial Bench evaluates direct single-image quantitative estimation and uses SpatialPrompt to encourage models to find reference objects, but these references are not explicitly provided with known metrics, so monocular scale ambiguity remains. In contrast, Metric-Bench provides sparse reference-object measurements as in-context anchors, requiring models to perform reference-grounded scale calibration, and covers richer outputs beyond distance and size. In addition, we evaluate MetricReasoner on Q-Spatial++

bench in a fully zero-shot way. This bench is captured from real scenes and is out-of-distribution for our model, MetricReasoner achieves 64.77 RMSE, compared with 143.54 for Qwen3-VL-8B and 72.58 for Gemini-2.5-Flash(cm).

12 Qualitative results on Metric-Bench

In Fig. 5, we present the inference results of two open-source models on Metric-Bench. Compared to commercial multimodal large language models, such as Gemini-2.5-Flash and GPT-5-Mini, the performance of InternVL3.5-38B and Qwen3-VL-32B is comparatively limited, primarily attributable to their constrained model scale. Our experimental analysis reveals that, while open-source models are capable of estimating the target scale based on known conditions and pixel-level ratios, they rarely leverage camera parameter estimation (e.g., focal length) to infer object depth and distance. In contrast, commercial models such as Gemini-2.5-Flash can explicitly incorporate the pinhole camera model: given known geometric priors, they analytically derive intrinsic camera parameters and thereby reason about object scale within the scene.

In the inference process of open-source models, reasoning is typically conducted in a stepwise manner: (i) identifying and localizing the given reference object; (ii) analyzing the spatial relationship between the target and the reference; and (iii) estimating the target's scale via projected pixel counts and relative scaling ratios.

A prevalent failure mode observed in our experiments stems from insufficient modeling of perspective effects. When the reference and target objects are in close proximity, predictions tend to be accurate; however, as inter-object distance increases, perspective distortion becomes non-negligible, and metric estimation based solely on pixel ratios incurs substantial error. This highlights a fundamental challenge for monocular vision-based models: robust spatial metric reasoning under perspective projection remains a significant bottleneck in single-image 3D understanding.

13 Qualitative results on manipulation tasks

In Fig. 6, we report the zero-shot results of MetricReasoner on RoboTwin2.0 benchmark [6]. Given the center position of a reference object in an operational scene, MetricReasoner can accurately localize the target object and infer its corresponding center position by estimating the relative position and scale relationship.

14 Performance of finer BP rewards

We evaluate finer Binned Proximity (BP) rewards, ranging from 5,6 and 8 intervals, compared to the 4 intervals we adopted, which is 0.5323(4),0.5685(5),0.5545(6) and 0.5367(8) in RMSE, and 48.59(4), 46.14(5), 48.88(6), 47.69(8) in MRA. Finer

Given the length of shoes is 0.4 meters, and the distance between shoes and keyboard is 0.539 meters, what is the direct distance between the shoes and the computer tower in meters measuring from the closest point? **GT:0.52136**

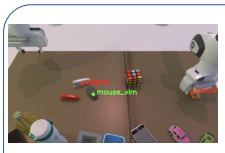
InternVL3.5-38B

To determine the distance between the shoes and the computer tower, we need to consider the spatial arrangement of the objects in the image and the given measurements.
1. **Anchor Objects and Their Dimensions:**
- **Shoes:** Given length is 0.4 meters.
- **Keyboard:** Not explicitly given, but typically around 0.45 meters in length.
- **Computer Tower:** Not explicitly given, but typically around 0.6 meters in height.
2. **Relative Positions:**
- The shoes are placed on the floor, and the keyboard is on the desk.
- The distance between the shoes and the keyboard is given as 0.539 meters.
3. **Estimating the Distance:**
- The computer tower is placed on the floor, close to the desk.
- Assuming the tower is directly below the desk, the distance from the shoes to the tower can be approximated by considering the horizontal distance from the shoes to the desk edge and the vertical distance from the desk to the tower.
4. **Calculating the Distance:**
- The horizontal distance from the shoes to the desk edge is 0.539 meters.
- The vertical distance from the desk to the floor (where the tower is) is approximately the height of the desk, which is typically around 0.75 meters.
Using the Pythagorean theorem to find the direct distance:
\\text{Distance} = \\sqrt{(\\text{horizontal distance})^2 + (\\text{vertical distance})^2}
\\text{Distance} = \\sqrt{(0.539)^2 + (0.75)^2} \\]\n\\[\\text{Distance} = \\sqrt{0.290521 + 0.5625} \\]\n\\[\\text{Distance} = \\sqrt{0.853021}
\\text{Distance} \\approx 0.923 meters
So, the direct distance between the shoes and the computer tower is approximately:```json\n**{\"distance\": 0.923}**\n```

Qwen3-VL-32B

Step 1: Understand the given anchors and their spatial relationships
The length of the shoes is 0.4 meters.
The distance between the shoes and the keyboard is 0.539 meters.
From the image:The keyboard is placed on the desk surface, near the front edge.
The shoes are located under the desk, on the floor, to the left side (from viewer's perspective).
The computer tower is also under the desk, on the right side, close to the leg of the desk.
Step 2: Use pixel-based measurement to estimate relative positions
We will use the known shoe length (0.4 m) as a scale reference to estimate distances in the scene.
Estimate the distance from shoes to keyboard:
In the image, we can visually measure the pixel distance between the shoes and the keyboard.The shoes are at the bottom-left under the desk.
The keyboard is on the desk, slightly to the right and above the shoes.
By measuring the vertical and horizontal pixel distances, we can compute the Euclidean distance in pixels.
Pixel distance between shoes and keyboard ≈ 280 pixels (based on visual estimation from the image).Given that this corresponds to 0.539 meters, we can compute the scale factor:
Scale = 0.539 m / 280 pixels ≈ 0.001925 m/pixel
Step 3: Measure pixel distance between shoes and computer tower
Now, we measure the pixel distance between the shoes and the computer tower.
The shoes are on the left side under the desk.
The computer tower is on the right side, under the desk, next to the desk leg.
Visually, the horizontal distance between them is large, and there's also some vertical separation (since the tower is closer to the front of the desk, while shoes are further back).
Estimating pixel distance:
Horizontal pixel distance ≈ 360 pixels
Vertical pixel distance ≈ 70 pixels
So, Euclidean pixel distance = $\sqrt{360^2 + 70^2}$ = $\sqrt{129600 + 4900}$ = $\sqrt{134500}$ ≈ 366.7 pixels
Step 4: Convert pixel distance to real-world metric
Using the scale factor (0.001925 m/pixel):
Distance = 366.7 pixels × 0.001925 m/pixel ≈ 0.706 meters
This is approximately 0.71 meters.
Step 5: Refine using perspective and geometry
However, note that the shoes and the computer tower are both on the floor, so the vertical offset is minimal — mostly horizontal displacement.
But the camera is looking slightly downward, so the actual ground plane distance may be slightly less than the pixel Euclidean distance due to perspective foreshortening.
Alternatively, since the keyboard is on the desk (higher up), its distance to the shoes includes vertical component, whereas the tower is on the same floor level as the shoes.
Therefore, the direct distance between shoes and tower should be measured horizontally on the ground plane.The horizontal pixel distance is ~360 pixels → 360 × 0.001925 ≈ 0.693 meters.Considering possible measurement error and rounding, we can round to **0.68 meters**.

Fig. 5: Results of open-source VLMs(InternVL3.5-38B and Qwen3-VL-32B) on Metric-Bench.



The center of the stapler is at [-0.170, 0.003, 0.795] in meters in camera coordinate system, and the center of the bread is at [0.420, 0.048, 0.762], Where is the center of the mouse located?

GT: [-0.108434,0.053334,0.758052]

</thinking>The stapler is located at [-0.170, 0.003, 0.795], this places it **on the left side of the scene**, **near the center of** the table, with a small **positive Y offset (0.003)** and a **Z value of 0.795**, suggesting it's on the table surface. The bread is at [0.420, 0.048, 0.762] to **the right of the stapler, slightly further forward (lower Z) and slightly higher (positive Y),** indicating it's also on the table. Both objects are on the same plane (table), and their **Z values are close (0.795 vs 0.762),** suggesting they are at **roughly the same depth** from the camera. The table surface appears flat, and the objects are arranged in a way that suggests a 2D plane with depth.
The target object is the mouse. In the image, a dark gray, oval-shaped object is located **between the stapler and the Rubik's cube**, **near the center of the table**. It is positioned **at approximately the same depth as the stapler and bread, but slightly to the right and slightly forward.**
We can use the stapler and bread as reference points to estimate the mouse's position.
we can estimate the mouse's position by averaging the coordinates of the stapler and bread, adjusted for the mouse's visual position. The mouse appears to be **roughly midway in X**, and its Y is about **1/4 of the way from stapler's Y to bread's Y**. The Z is **slightly less** than the stapler's Z (0.795) and slightly more than the bread's Z (0.762), so we can average them: **(0.795 + 0.762) / 2 = 0.7785**.
However, this is a rough average. Since the mouse is visually **closer to** the stapler, we can adjust the Y-coordinate to be closer to bread's Y. The bread's Y is 0.048, and the stapler's is 0.003, so the difference is 0.045. If the mouse is **about 2/3 of the way from stapler to bread**, then **Y = 0.003 + (2/3)*0.045 = 0.003 + 0.03 = 0.033**. For Z, since the mouse is visually **closer to the bread** (lower Z), we can adjust Z to be closer to bread's Z: **Z = 0.795 - (1/3)*(0.795-0.762) = 0.795 - 0.011 = 0.784**.**</thinking>**
< Final answer >{\"center\": [-0.105, 0.058, 0.770]}</answer>"

The center of the mouse is at [0.139, 0.003, 0.759] in meters in camera coordinate system, and the center of the coffee-box is at [0.402, -0.078, 0.820], Where is the center of the soap located?

GT: [-0.176853,0.054626,0.720486]

</thinking>The mouse is located at [0.139, 0.003, 0.759], which is relatively **close to the camera (z=0.759m)** and **near the center of the scene (x=0.139m, y=0.003m)**.
The coffee-box is at [0.402, -0.078, 0.820], **slightly further away (z=0.820m)** and **shifted to the right and slightly down** (x=0.402m, y=-0.078m).
The spatial layout suggests they are **arranged in a row or cluster** on a flat surface, with the coffee-box to the right of the mouse.
The target object is the soap.
From the image, we can see a yellow, rectangular object on the **left side of the scene**, which appears to be the soap. The soap, mouse, and coffee-box are all on the same surface.
The mouse is at x=0.139, and the soap is **to the left of** it, likely at a **smaller x-coordinate**. The coffee-box is at x=0.402, which is to **the right of** the mouse. The soap is likely positioned **between the left edge of the scene and the mouse**. **</thinking>**
< Final answer >{\"center\": [-0.157,0.053,0.665]}</answer>"

Fig. 6: Visualization of MetricReasoner on RoboTwin2.0. Points on images are the projection of the predicted results.

bins do not consistently improve performance, suggesting that coarse BP provides stable proximity guidance while EP supplies fine-grained supervision.

Table 6: BP reward supplementary experiments.

Reward	**MRA**↑	**RMSE**↓	δ_1 ↑
BP	48.59	**0.5323**	0.4705
BP-5	46.14	0.5685	0.4290
BP-6	**48.88**	0.5545	**0.4800**
BP-8	47.69	0.5367	0.4611