TOWARDS CAMERA-ROBUST 3D LOCALIZATION: EQUATION-ANCHORED TOOL-USE FOR MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

3D localization in Multimodal Large Language Models (MLLMs), including 3D object detection and 3D visual grounding, is fundamentally limited by camera intrinsic ambiguity: the same image admits different 3D scenes under different cameras. Existing MLLMs either ignore camera parameters and overfit to a canonical training intrinsic, or retrieve depth and 3D cues from external tools but treat the returned values as *reference cues* (numerical hints that the model is free to interpret implicitly) rather than formula variables in an explicit geometric computation. We propose an *equation-anchored tool-use* framework that re-purposes spatial tools as *formula variables*. The proposed framework proactively retrieves camera intrinsics and samples multi-point metric depths, writes the pinhole back-projection equation $\hat{X}_c = (u_c - c_x)\bar{Z}/f_x$ explicitly in Chain-of-Thought (CoT), and substitutes the tool outputs into the formula to compute a geometry-derived 3D center estimate. The MLLM then combines this estimate with visual evidence to predict the final center, dimensions, and orientation of the 9-DoF bounding box. On both 3D object detection and 3D visual grounding tasks under rescaled camera intrinsics from $0.5\times$ to $1.5\times$, our method outperforms RGB-only and tool-augmented baselines, with significant gains where the camera deviates most from the training scale. Code and data will be released.

1 INTRODUCTION

Recovering absolute 3D geometry from a single RGB image is a core ability for Multimodal Large Language Models to interact with the physical world, underpinning various 3D localization tasks such as 3D object detection and 3D visual grounding. While Multimodal Large Language Models (MLLMs) have achieved remarkable progress on 2D vision-language understanding (Alayrac et al., 2022; Bai et al., 2023; Liu et al., 2023; Hurst et al., 2024; Bai et al., 2025; Comanici et al., 2025), their performance on 3D localization tasks remains fundamentally limited. The reason is geometric: a single 2D image is inherently underdetermined, as the same projection admits infinitely many 3D scenes under different camera intrinsics. Without an explicit mechanism to propagate camera parameters through reasoning, an MLLM is forced to rely on implicit statistical priors memorized from its training data. While scaling across diverse datasets may allow the model to approximate common camera configurations, this implicit approximation does not explicitly account for camera geometry, and its 3D predictions degrade sharply once the camera deviates from the familiar distribution. Camera-robust 3D localization, therefore, benefits from explicitly incorporating camera information through a prescribed geometric computation rather than relying on a fixed intrinsic.

As illustrated in Figure 1, two paradigms have been explored to address this challenge, yet both leave camera information weakly grounded in the reasoning process. Early RGB-only MLLMs (Zheng et al., 2025b;a; Zhang et al., 2025) bypass camera parameters entirely and treat 3D localization as an end-to-end visual regression problem. Without access to camera intrinsics, they are forced to implicitly estimate physical scale based on visual cues derived from their training distribution. While this data-driven approximation may succeed under familiar camera setups, it lacks strict mathematical constraints, leaving the models vulnerable to geometric collapse when the camera undergoes out-of-distribution shifts. More recently, tool-augmented MLLMs (Zhou et al., 2025b; Wang et al., 2025; Tian et al., 2026; Zhang et al., 2026c) retrieve depth maps, metric distances, or even per-object 3D coordinates from external geometric tools. Although these methods successfully inject geomet-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

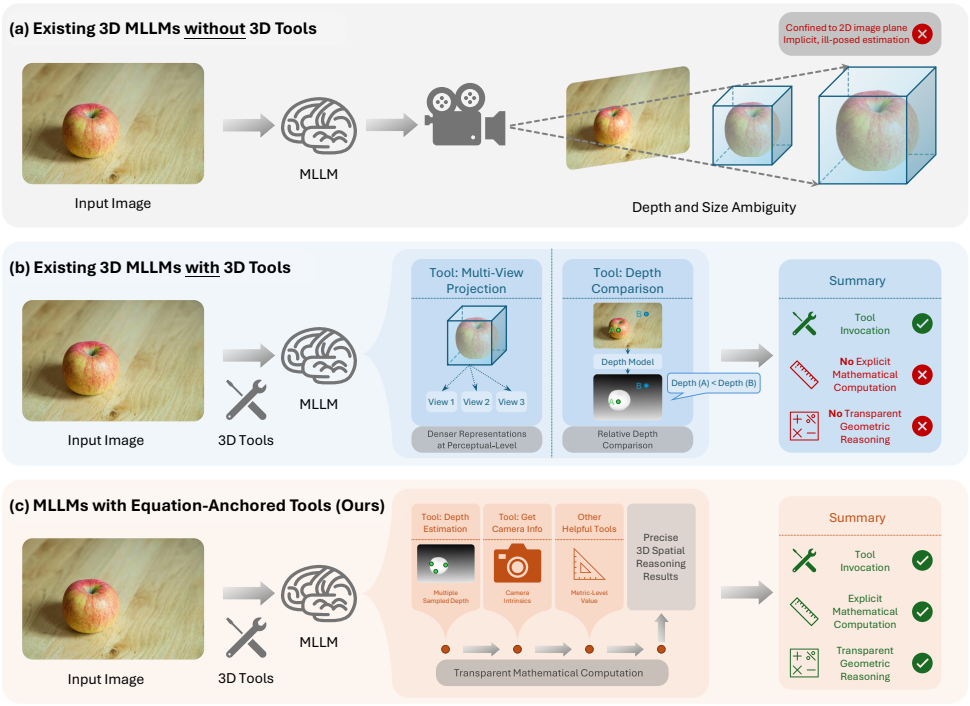

Figure 1: Comparison of spatial reasoning paradigms in Multimodal Large Language Models (MLLMs). (a) Existing RGB-only MLLMs rely on implicit spatial estimation confined to the 2D image plane, leading to depth and physical scale ambiguity. (b) Recent tool-augmented paradigm incorporates 3D tools but consumes the numerical returns merely as soft reference cues, failing to propagate camera parameters through the reasoning process. (c) Our proposed equation-anchored paradigm re-purposes retrieved metric depths and camera intrinsics as strict formula variables. By explicitly substituting these variables into the pinhole back-projection equation within a transparent Chain-of-Thought, the model computes a geometry-derived 3D center estimate, which conditions the learned prediction of the final 9-DoF box. This explicit camera-conditioned computation improves robustness under the evaluated intrinsic rescalings.

ric information into the reasoning context, they consume the returned values as *reference cues*: the MLLM is shown a numerical value, such as an object depth of 1.05m or a coordinate triplet, typically as a soft contextual hint rather than as the input to a specific geometric computation. For instance, Think3D (Zhang et al., 2026c) leverages 3D tools primarily for multi-view rendering and reconstruction, treating the resulting views as denser visual context, while LAST (Tian et al., 2026) provides numerical per-object depths and absolute 3D coordinates that the MLLM is nevertheless free to ignore, approximate, or downgrade to relative hints during downstream reasoning. In all of these methods, regardless of whether the tool output is a depth map or an absolute coordinate, its *usage pattern* stays implicit: the camera information is not explicitly used in a prescribed geometric computation before the final prediction.

We argue that the central bottleneck of camera-robust 3D localization is not whether tool outputs are numerical or qualitative, but *how the model uses them*. Motivated by this, we propose an equation-anchored tool-use framework that re-purposes spatial tools as *formula variables* rather than reference cues. Given a single RGB image, the MLLM proactively retrieves camera intrinsics (f_x, f_y, c_x, c_y) via a Camera Intrinsic Tool, and samples multi-point metric depths within object masks via a Multi-Point Depth Sampling Tool. It writes the pinhole back-projection equation $\hat{X}_c = (u_c - c_x)\bar{Z}/f_x$, $\hat{Y}_c = (v_c - c_y)\bar{Z}/f_y$, $\hat{Z}_c = \bar{Z}$ explicitly in its Chain-of-Thought, substituting the retrieved tool outputs into the formula step by step to obtain a geometry-derived center estimate. The MLLM then combines this estimate with visual evidence to predict the final center, dimensions, and orientation of the 9-DoF bounding box. A dedicated training pipeline supervises the model on traces that interleave tool calls with such substitutions, providing a reasoning template for the geometric computation.

This design bears two properties that structurally distinguish our framework from prior tool-augmented approaches. First, the retrieved camera intrinsics are explicitly used in the pinhole equation rather than provided only as optional context. For fixed pixel and depth inputs, changes in (f_x, f_y, c_x, c_y) deterministically change the geometry-derived center estimate. This estimate provides an explicit camera-conditioned signal for the learned 9-DoF predictor, which empirically improves robustness under the evaluated camera rescalings. Second, by elevating tool invocation to an explicit equation-anchored computational loop, our design pioneers a mathematically-driven tool-use paradigm for 3D localization. It enables MLLMs to invoke specialized geometric tools and process their numerical returns as strict mathematical variables rather than visual cues. This mathematically-grounded approach provides a starting point for future research in embodied AI, equipping spatial agents to actively interrogate complex physical environments instead of simple visual approximations.

Extensive experiments validate the efficacy of this paradigm. We benchmark our framework on 3D object detection and 3D visual grounding tasks, evaluating under varying camera rescale factors. Results show that existing baselines suffer from geometric collapse when the camera deviates from the familiar training scale. In contrast, by interleaving tool retrieval with transparent symbolic substitution to compute an explicit camera-conditioned center estimate, our equation-anchored framework improves spatial localization robustness across the evaluated rescaling range.

The major contributions of this work are threefold. First, we distinguish between using spatial tool outputs as *reference cues* and treating them as *formula variables* in camera-robust 3D localization. Many existing tool-augmented methods present these outputs as contextual information without explicitly defining how the corresponding quantities enter a geometric computation. Second, we propose an equation-anchored tool-use framework that supervises the MLLM to substitute retrieved camera intrinsics and multi-point depths into an explicit pinhole equation during Chain-of-Thought, computing a geometry-derived center estimate before predicting the final 9-DoF 3D bounding box. The equation determines the intermediate estimate, while the final center, dimensions, and orientation remain learned outputs conditioned on this estimate and visual evidence. The resulting reasoning chain is auditable, exposing every intermediate quantity for failure analysis. Third, we evaluate on both 3D object detection and 3D visual grounding tasks under rescaled camera intrinsics from $0.5\times$ to $1.5\times$, where our method outperforms both RGB-only and tool-augmented baselines, with significant gains appearing precisely where the camera deviates most from the training scale.

2 Related Work

2.1 MLLMs for Spatial Intelligence

While Multimodal Large Language Models have demonstrated remarkable success in standard 2D tasks (Alayrac et al., 2022; Bai et al., 2023; Liu et al., 2023; Hurst et al., 2024; Comanici et al., 2025), they inherently struggle with 3D spatial intelligence (Yang et al., 2025a; Chen et al., 2024; Yu et al., 2025; Cai et al., 2025; Fan et al., 2025; Wu et al., 2025; Feng et al., 2025; Zhang et al., 2026a), which requires a precise comprehension of physical scales, geometry, and spatial relationships. To mitigate this limitation, many prior works have focused on enriching spatial representations by incorporating auxiliary cues, such as depth maps (Cheng et al., 2024; Zhou et al., 2025a; Liu et al., 2025) and segmentation masks (Wang et al., 2023; Huang et al., 2024). However, most of these approaches treat auxiliary data as enhanced perceptual features rather than explicit variables for precise spatial reasoning. Alternatively, models tackling 3D scene understanding with RGB-only inputs (Zheng et al., 2025b;a; Zhang et al., 2025) attempt to interpret 3D environments from monocular visual cues, but inevitably suffer from geometric collapse and depth ambiguity due to their restriction to the 2D image plane. Recent methods explore diverse routes to spatial reasoning, including tool-assisted reasoning in LAST (Tian et al., 2026) and Think3D (Zhang et al., 2026c), spatial instruction tuning with depth and multi-view data in MM-Spatial (Daxberger et al., 2025), and grounded 3D prediction in GR3D (Cheng et al., 2026). Our method differs in how geometric information is used: it explicitly substitutes retrieved camera intrinsics and metric depths into the pinhole back-projection equation within the reasoning trace for 3D localization. By doing so, our paradigm bridges abstract visual perception with concrete geometric logic, transforming spatial understanding from implicit estimation into explicit mathematical reasoning.

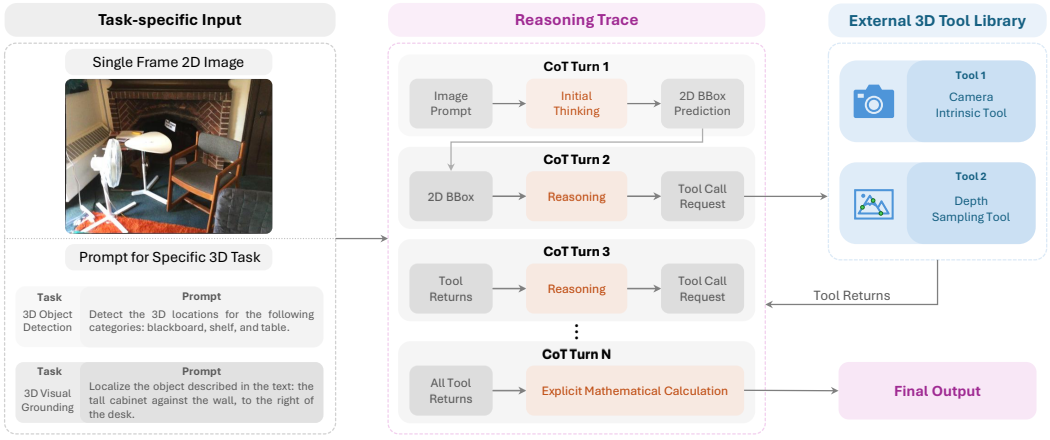

Figure 2: Overview of our proposed equation-anchored spatial agent. Given a single-frame RGB image and a task-specific prompt, the agent first leverages its internal visual perception to predict initial 2D bounding boxes without relying on external detectors. Subsequently, it engages in a structured multi-turn Chain-of-Thought reasoning trace, utilizing these 2D spatial anchors to interactively retrieve exact camera parameters and metric depths from an external 3D Tool Library. In the concluding turn, rather than treating the tool outputs as soft reference cues, the agent uses them as variables in the pinhole back-projection equation to compute a geometry-derived center estimate. The model then combines this estimate with visual evidence to predict the final center, dimensions, and orientation of the 9-DoF boxes.

2.2 Tool-Assisted Reasoning

To extend the reasoning capabilities of foundation models, early tool-augmented reasoning frameworks (Nakano et al., 2021; Shen et al., 2023; Wu et al., 2023) empowered LLMs to invoke external tools, such as code generation (Surís et al., 2023; Gao et al., 2023) and symbolic solvers (Pan et al., 2023; Li et al., 2024), enabling them to tackle complex logical and mathematical problems beyond their innate capabilities. While these text-based reasoning frameworks excel in logic and mathematics, they fundamentally fail to comprehend continuous spatial environments due to a lack of visual engagement. To bridge this gap, "thinking with images" paradigms (Su et al., 2025b; Huang et al., 2026; Zhang et al., 2026b; Zheng et al., 2026) have been proposed. In diagram-based geometry reasoning, GF-Reasoner interleaves natural-language reasoning with solver-executable formal programs (Yang et al., 2025b). In these CoT frameworks, some MLLMs (Hu et al., 2024) invoke specialized 2D vision experts, such as object tracking, grounding, and segmentation models, to generate intermediate visual evidence, while others perform intuitive 2D-level visual operations like cropping or masking (Fu et al., 2025; Su et al., 2025a; 2026; Sarch et al., 2025; Zheng et al., 2026).

To enrich the spatial context for complex 3D scene comprehension, recent efforts incorporate specialized spatial vision tools for depth estimation (Zhou et al., 2025b; Wang et al., 2025; Tian et al., 2026) and 3D reconstruction (Zhou et al., 2025b; Wang et al., 2025; Tian et al., 2026). Existing methods typically use these outputs as contextual inputs rather than explicitly substituting retrieved camera intrinsics and metric depths into a prescribed geometric equation. Rather than treating these outputs as coarse visual references, our method explicitly formulates spatial tool outputs as mathematical variables. By integrating multi-point depth samples and camera intrinsics into an explicit geometric reasoning chain, our agent performs transparent mathematical calculations, grounding 3D localization in explicit geometric formulations rather than implicit spatial regression.

3 Method

3.1 Overall Architecture

Figure 2 illustrates the overall architecture of our proposed equation-anchored spatial agent. To achieve camera-robust 3D spatial understanding and prediction, we empower the Multimodal Large

Language Model (MLLM) to interactively acquire explicit geometric data via a structured tool-use-and-reasoning loop.

Given a single-frame RGB image I and a task-specific natural language prompt T, the agent initiates the first Chain-of-Thought turn. It leverages its internal visual perception to predict a set of 2D bounding boxes $\mathcal{B}_{2D}$ for the target objects directly, without relying on external 2D detectors. Subsequently, the agent utilizes these 2D bounding boxes as initial spatial anchors to interact with an external 3D Tool Library. Over successive CoT turns, the MLLM issues tool call requests, such as the Camera Intrinsic Tool and the Depth Sampling Tool, to retrieve precise camera parameters (f_x, f_y, c_x, c_y) and explicit metric depth observations (u, v, Z) within the specified 2D regions. Importantly, rather than treating these retrieved values as implicit *reference cues*, our framework uses them as *formula variables*. By substituting the retrieved depths and camera intrinsics into the pinhole back-projection equation within its reasoning chain, the agent computes a geometry-derived center estimate $(\hat{X}_c, \hat{Y}_c, \hat{Z}_c)$. In the concluding CoT turn, the model uses this estimate together with visual evidence and learned priors to predict the final task-specific results, formatted as a JSON list of 9-DoF 3D bounding boxes $\mathcal{B}_{3D}$. Each bounding box is represented as $bbox_{3D} = [X_c, Y_c, Z_c, l, w, h, \psi, \theta, \phi]$, where (X_c, Y_c, Z_c) denotes the final center predicted using the geometry-derived estimate together with visual evidence, and $(l, w, h, \psi, \theta, \phi)$ denote the predicted physical dimensions and orientation angles.

3.2 3D Spatial Tools

To realize our equation-anchored framework, we introduce a geometry-driven 3D spatial toolkit designed to extract strict formula variables rather than implicit perceptual hints. This toolkit equips the MLLM with explicit and numerical geometric measurements, serving as the rigorous mathematical foundation for precise 3D spatial grounding.

Multi-Point Depth Sampling. To acquire object-level geometric data, the model queries the Depth Sampling Tool with a list of target objects. Each target object is formulated as a semantic category label c paired with a 2D bounding box $bbox_{2D}$, formalized as a query tuple $(c, bbox_{2D})$, where $bbox_{2D} = [u_{\min}, v_{\min}, u_{\max}, v_{\max}]$. The tool leverages a promptable segmentation model (Carion et al., 2025) to extract a precise object mask based on the initial 2D spatial anchor. From this masked region, it samples multiple valid pixels (up to N points per object) and returns depth triplets:

$$[(u_1, v_1, Z_1), \ldots, (u_N, v_N, Z_N)], \tag{1}$$

where (u_i, v_i) represents the 2D pixel coordinates of the sampled point, Z_i denotes the corresponding metric depth in meters, and N is set to 5. Specifically, the metric depth values Z_i are directly extracted from a dense depth map provided by datasets or predicted by a monocular depth estimator (Piccinelli et al., 2025) conditioned on the retrieved intrinsics. By explicitly formatting these outputs as numerical triplets, the tool provides metric depth samples that are aggregated into the formula variable $\bar{Z}$. Together with the 2D bounding-box center and camera intrinsics, $\bar{Z}$ is used to compute a geometry-derived center estimate.

Camera Intrinsic Retrieval. Given the image context, the Camera Intrinsic Tool extracts the focal lengths and principal point coordinates (f_x, f_y, c_x, c_y) from the dataset-provided camera calibration parameters. Importantly, rather than feeding these parameters into the MLLM only as soft contextual tokens, the training traces explicitly use them in the back-projection equation. Together with representative pixel coordinates (u, v) and metric depth Z, they support computation of a camera-conditioned center estimate, while the final 9-DoF box remains a learned prediction. By explicitly conditioning the center estimate on the camera intrinsics, this design reduces the model's reliance on a fixed-camera prior and supports generalization across cameras with different intrinsic parameters.

3.3 Explicit Geometric Chain-of-Thought

We formulate an explicit Geometric Chain-of-Thought to achieve camera-robust metric 3D localization. During SFT, we supervise the model to explicitly substitute retrieved tool outputs as formula variables into mathematical equations rather than treating them as soft reference cues. This supervision encourages a transparent and interpretable reasoning trajectory spanning five sequential steps:

Step 1: Semantic Grounding on the 2D Plane. Leveraging the robust 2D perception capabilities inherent in MLLMs, the agent directly localizes target objects on the 2D image plane without relying

on external detectors. It initially predicts bounding boxes in a normalized coordinate space, which aligns with the native localization format of the MLLMs' original training paradigm. To seamlessly bridge these intermediate 2D predictions with subsequent geometric tool invocations, the reasoning trajectory then guides the model to calculate the absolute pixel coordinates:

$$u_{\text{abs}} = \text{round}\left(\frac{u_{\text{norm}}}{1000} \cdot W\right), \quad v_{\text{abs}} = \text{round}\left(\frac{v_{\text{norm}}}{1000} \cdot H\right). \tag{2}$$

Here, $(u_{\text{norm}}, v_{\text{norm}})$ follows the normalized coordinate convention used by Qwen3-VL. Division by 1000 converts the model outputs to relative image coordinates before they are scaled by the image width W and height H. The resulting u_{abs} and v_{abs} denote the absolute pixel coordinates.

Step 2: Camera Intrinsic Retrieval. To initiate the 2D-to-3D spatial lifting, the model first issues a tool call to retrieve the camera parameters. This step isolates the camera configuration from implicit visual features, extracting the exact focal lengths and principal point coordinates (f_x, f_y, c_x, c_y) required as strict formula variables for subsequent back-projection.

Step 3: Metric Depth Acquisition. Subsequently, the agent invokes the Depth Sampling Tool to acquire the depth of the target objects from multiple points. By querying the tool with the localized 2D bounding boxes, the model collects valid metric depth values within the object-specific segmentation masks. To ensure geometric reliability, noisy depths smaller than 0.1 meters are systematically discarded from the tool's response, yielding a robust set of depth variables.

Step 4: Rigorous Mathematical Deduction. Rather than consuming the tool outputs as implicit reference cues, the Chain-of-Thought executes an equation-anchored deduction. Given the predicted absolute 2D bounding box derived in Step 1, denoted as $bbox_{2D} = [u_{\min}, v_{\min}, u_{\max}, v_{\max}]$, the model explicitly computes the 2D bounding box center via:

$$u_c = \frac{u_{\min} + u_{\max}}{2}, \quad v_c = \frac{v_{\min} + v_{\max}}{2}. \tag{3}$$

The sampled depths $\{Z_i\}_{i=1}^{N}$ are aggregated into an average depth $\bar{Z}$. Given (u_c, v_c), $\bar{Z}$, and K, the pinhole camera model yields a geometry-derived center estimate:

$$\hat{X}_c = \frac{(u_c - c_x) \cdot \bar{Z}}{f_x}, \quad \hat{Y}_c = \frac{(v_c - c_y) \cdot \bar{Z}}{f_y}, \quad \hat{Z}_c = \bar{Z}. \tag{4}$$

Equation 4 yields the geometry-derived center estimate $(\hat{X}_c, \hat{Y}_c, \hat{Z}_c)$, which is used to condition the subsequent 9-DoF prediction.

Step 5: Geometry-Conditioned 3D Spatial Estimation. Finally, the MLLM combines the center estimate with visual features and learned category and shape priors to predict the final 9-DoF bounding box $bbox_{3D}$:

$$bbox_{3D} = [X_c, Y_c, Z_c, l, w, h, \psi, \theta, \phi]. \tag{5}$$

Here, $(\hat{X}_c, \hat{Y}_c, \hat{Z}_c)$ provides a camera-conditioned metric reference for predicting the final center (X_c, Y_c, Z_c), while the dimensions (l, w, h) and orientation (ψ, θ, ϕ) are inferred from visual features and learned priors.

3.4 Training Pipeline

To empower standard MLLMs with this explicit 3D reasoning capability, we design a geometry-guided supervised fine-tuning (SFT) pipeline, where the model is supervised using structured, multi-turn equation-anchored reasoning traces rather than direct end-to-end 3D bounding box annotations.

Data Construction. We construct rigorously structured reasoning data based on the ScanNet (Dai et al., 2017) dataset, which features diverse indoor scenes to support the 3D object detection task. For 3D visual grounding, we process target frames from the ScanRefer (Chen et al., 2020) dataset to facilitate the construction of single-frame reasoning samples, utilizing Qwen3.5 to generate and verify context-aware referring expressions. For both tasks, ground-truth 3D boxes are projected onto the image plane to obtain 2D regions. Based on the 2D regions and object categories, we then employ SAM3 (Carion et al., 2025) to generate precise object masks, from which we extract

Table 1: 3D Object Detection performance across rescale factors. Best in **bold**, second underlined.

Method	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
Generalist MLLMs											
GPT-5.4-2026-03-05 (OpenAI, 2026a)	2.67	1.11	0.81	1.96	0.83	0.63	0.41	1.37	1.31	1.34	0.59
GPT-6 Astra (OpenAI, 2026b)	6.70	7.39	7.45	8.86	11.09	15.17	12.95	11.96	10.55	7.02	7.58
DeepSeek-v4-Flash (DeepSeek-AI, 2026)	0.00	0.24	0.00	0.11	0.00	0.12	0.00	0.17	0.00	0.42	0.00
Claude-Sonnet-4.6 (Anthropic, 2026b)	0.87	1.50	1.35	0.79	0.53	0.94	1.64	0.78	2.44	2.70	1.84
Claude-Opus-4.7 (Anthropic, 2026a)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kimi-K2.6 (Moonshot AI, 2026)	0.28	0.00	0.06	0.08	0.08	0.00	0.17	0.08	0.08	0.00	0.00
Qwen3.6-Flash (Qwen Team, 2026)	4.03	9.41	10.01	19.57	18.94	25.68	19.47	16.82	19.94	17.40	17.88
Seed-2.0-Pro-260215 (Seed Team, 2026)	2.00	5.58	7.56	14.50	17.40	24.00	20.84	19.23	17.20	14.28	12.54
Specialist MLLMs											
VG LLM-8B (Zheng et al., 2025a)	13.19	16.77	24.30	25.12	29.67	38.49	32.78	30.40	23.08	24.07	17.16
Qwen3-VL-8B-Instruct (Bai et al., 2025)	10.75	13.83	23.99	33.15	33.29	38.20	31.18	27.17	22.04	18.52	15.05
Ours-8B (w/ UniDepthV2 (Piccinelli et al., 2025))	16.97	**30.85**	33.33	35.26	**42.38**	44.13	39.70	31.88	27.16	21.65	20.76
Ours-8B (w/ GT depth)	**18.93**	30.03	**36.08**	**37.42**	41.69	**44.90**	**39.81**	**34.25**	**28.48**	**24.40**	**24.29**

Table 2: 3D Visual Grounding performance across rescale factors. Best in **bold**, second underlined.

Method	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
Generalist MLLMs											
GPT-5.4-2026-03-05 (OpenAI, 2026a)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GPT-6 Astra (OpenAI, 2026b)	2.92	4.51	3.98	4.77	6.37	7.16	5.84	4.51	2.92	2.65	3.71
DeepSeek-v4-Flash (DeepSeek-AI, 2026)	0.53	0.27	0.27	0.00	0.53	0.53	0.80	0.27	0.80	0.27	0.27
Claude-Sonnet-4.6 (Anthropic, 2026b)	0.27	0.00	0.00	0.00	0.00	0.00	0.27	0.00	0.00	0.00	0.27
Claude-Opus-4.7 (Anthropic, 2026a)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kimi-K2.6 (Moonshot AI, 2026)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Qwen3.6-Flash (Qwen Team, 2026)	3.71	7.96	11.67	19.89	26.79	33.42	40.05	38.46	31.30	25.20	22.28
Seed-2.0-Pro-260215 (Seed Team, 2026)	4.51	9.28	17.24	28.65	36.34	40.58	32.63	25.46	20.95	12.73	10.88
Specialist MLLMs											
VG LLM-8B (Zheng et al., 2025a)	**17.51**	22.28	27.06	34.75	43.77	48.28	47.75	40.85	35.28	30.77	24.67
Qwen3-VL-8B-Instruct (Bai et al., 2025)	8.49	16.98	28.38	41.64	51.72	53.85	48.54	40.32	32.36	23.61	20.69
Ours-8B (w/ UniDepthV2 (Piccinelli et al., 2025))	13.26	24.14	38.20	48.28	**57.03**	55.17	57.82	49.60	43.50	35.28	28.12
Ours-8B (w/ GT depth)	11.41	**25.46**	**42.71**	**50.13**	**57.03**	**58.62**	**58.89**	**51.19**	**43.77**	**36.07**	**32.89**

point-wise metric depth values to simulate the exact outputs of the Multi-Point Depth Sampling tool. Importantly, these trajectories strictly format ground-truth geometric measurements as explicit formula variables rather than reference cues, embedding them into a standardized multi-turn template (<tool_call>, <tool_response>, <think>, <answer>). This supervision exposes the model to the same step-by-step substitution pattern used at inference and encourages it to use the computed center estimate when predicting the final box.

Model Training. We adopt Qwen3-VL-8B-Instruct (Bai et al., 2025) as our base model. To preserve pretrained visual representations, the visual encoder and projector are frozen, updating only the decoder and language head. Within chat templates, user instructions and system prompts are masked from the loss, while optimization is applied to assistant-side tokens. Supervising the model on these equation-anchored tool-use traces encourages it to follow the same equation-based reasoning process during inference.

4 Experiments

4.1 Experimental Settings

Evaluation Metrics. For 3D object detection, we adopt the Average F1 (Avg F1) score with an Intersection over Union (IoU) threshold of 0.25 as the evaluation metric. For 3D visual grounding, we report the accuracy at an IoU threshold of 0.25 (Acc@IoU=0.25). Unless otherwise specified, the reported performance for 3D object detection is averaged across 31 common categories.

4.2 Benchmarking with the State-of-the-Art

Quantitative Benchmarking. Tables 1 and 2 present 3D object detection and 3D visual grounding results under varying rescale factors. Compared to specialist MLLMs, our framework achieves superior performance and better camera-robustness. The superior performance is largely attributed to two key design choices. First, rather than treating tool outputs as soft cues, our model extracts metric depths and intrinsics strictly as formula variables. Second, explicit equation-anchored reasoning uses these variables to compute a camera-conditioned center estimate before the final 9-DoF

Table 3: Ablation study of different spatial tools (rescale factor = 1.0).

Spatial Tools		3D Object Detection			3D Visual Grounding
Camera Intrinsic	Depth Sampling	Category 8	Category 20	Category 31	Performance
		60.73	45.90	38.43	50.13
✓		63.11	48.38	40.27	51.46
	✓	64.63	48.29	40.09	54.38
✓	✓	**67.47**	**54.05**	**44.13**	**55.17**

Table 4: Ablation study of different depth sources in the reasoning chain (rescale factor = 1.0).

	3D Object Detection			3D Visual Grounding
Depth Source	Category 8	Category 20	Category 31	Performance
UniDepthV2 (Piccinelli et al., 2025)	67.47	54.05	44.13	55.17
GT Depth	**70.51**	**55.27**	**44.90**	**58.62**

Table 5: Ablation study of the explicit equation-anchored CoT reasoning (rescale factor = 1.0).

	3D Object Detection			3D Visual Grounding
Explicit CoT	Category 8	Category 20	Category 31	Performance
	67.21	47.66	38.20	53.85
✓	**67.47**	**54.05**	**44.13**	**55.17**

prediction. Consequently, RGB-only models like VG LLM and Qwen3-VL-8B-Instruct overfit to canonical training intrinsics and suffer geometric collapse during camera shifts. For instance, in the 3D visual grounding task, the performance of Qwen3-VL-8B-Instruct drops from 53.85% at the original scale to 8.49% at the 0.5× scale. Furthermore, our framework equipped with ground-truth (GT) depth outperforms the variant using UniDepthV2 in most scales. This performance gain shows that our equation-anchored reasoning chain benefits from more accurate intermediate variables.

Moreover, evaluations of generalist MLLMs reveal failures in absolute 3D localization across varying camera scales. Despite their extensive multi-modal pre-training, these models remain limited in absolute 3D localization across varying camera scales. In the absence of camera intrinsics, absolute metric regression becomes a mathematically ill-posed problem, where even minor discrepancies in coordinate estimation result in zero IoU under the strict 3D evaluation criteria. This confirms that shifting from purely implicit heuristic estimation to explicit, equation-anchored geometric reasoning is beneficial for 3D spatial localization. GPT-6 Astra also remains below our method on both tasks across all evaluated camera scales. Unlike RGB-only prompting, our method uses camera intrinsics and metric depths through equation-anchored CoT supervision, providing explicit geometric information and task-specific training for metric 3D localization.

4.3 Ablation Study

Effectiveness of Spatial Tools. As Table 3 shows, the baseline lacking both spatial tools struggles across all evaluation metrics. Relying on purely implicit 2D-to-3D projection, it suffers from severe scale ambiguity. Incorporating the Camera Intrinsic Tool yields clear improvements by establishing the explicit geometric relationship between 2D pixels and 3D projection rays. Similarly, introducing the Depth Sampling Tool provides improvements by equipping the model with metric depth values to break the depth ambiguity. Finally, the best performance is achieved when both spatial tools are included. This combination successfully completes the required variables for the pinhole back-projection equation, effectively elevating the MLLM from a perceptual estimator to a mathematically-grounded spatial agent.

Impact of Depth Source Quality. In Table 4, we investigate the effect of depth source in our framework by substituting the predicted depths (UniDepthV2) with Ground Truth (GT) depths within the reasoning chain. Equipping the tool with GT depth yields substantial improvements, notably boosting 3D visual grounding from 55.17% to 58.62% and 8-category 3D object detection from 67.47% to 70.51%. Because tool outputs enter the explicit center computation, inaccuracies directly perturb the intermediate center estimate and can degrade the downstream 9-DoF prediction.

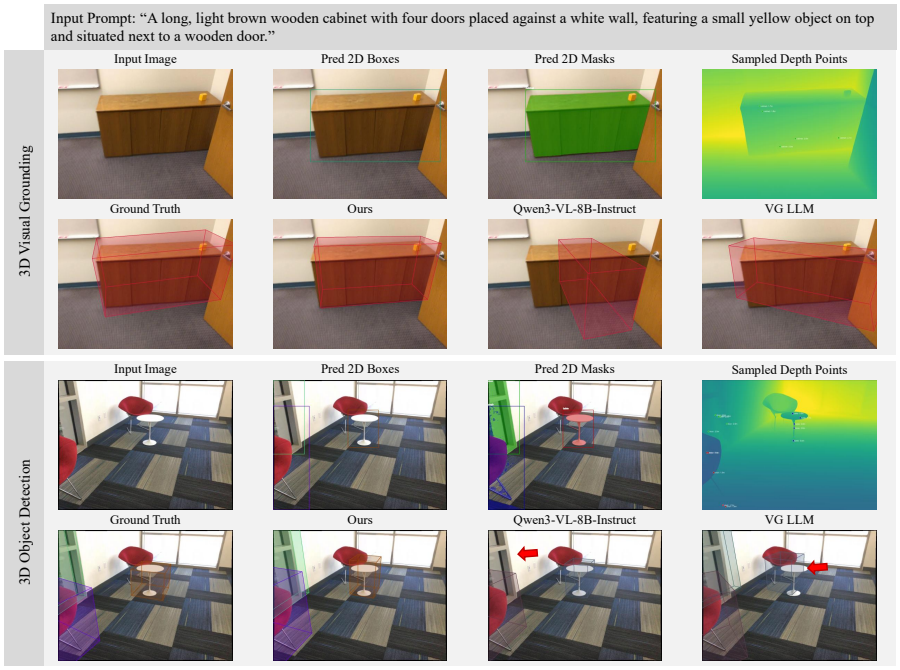

Figure 3: Qualitative comparison on 3D visual grounding (top) and 3D object detection (bottom). Red arrows highlight incorrect predictions.

Necessity of Explicit CoT Reasoning. Table 5 highlights the critical role of explicit mathematical derivation. Removing the equation-anchored Chain-of-Thought (w/o CoT) forces the model to directly regress 3D bounding boxes. This opaque mapping causes severe performance drops in complex scenarios (e.g., 31-category performance falls from 44.13% to 38.20%). In contrast, explicit CoT makes the substitution of geometric variables into projection formulas visible and yields better performance in this ablation, supporting the value of equation-based reasoning in our setting.

4.4 Visualization Results

Figure 3 presents comparisons on 3D visual grounding (top) and 3D object detection (bottom). We demonstrate the spatial tool outputs of our framework, including the predicted 2D boxes, 2D masks, and depth points, to illustrate the geometric deduction process. Our method produces more accurate 3D localizations than Qwen3-VL-8B-Instruct and VG LLM in these examples. Unlike these baselines, which rely on implicit visual heuristics, our framework explicitly uses retrieved metric depths and camera intrinsics to compute a metric center estimate. Together with visual evidence, this estimate guides the final 9-DoF box prediction, improving camera-aware spatial localization.

5 Conclusion

This paper presents an equation-anchored tool-use framework that elevates Multimodal Large Language Models from implicit perceptual estimators to mathematically-grounded spatial agents for precise 3D scene understanding. To overcome the fundamental camera intrinsic ambiguity inherent in single-image 3D localization, we propose a paradigm shift in how spatial tools are utilized. First, our proactive tool invocation mechanism strictly formulates spatial tool outputs, such as multi-point depth samples and camera intrinsics, as explicit formula variables rather than soft reference cues. Second, through a dedicated Chain-of-Thought training pipeline, we establish an explicit geometric reasoning chain that interleaves these numerical variables with transparent symbolic substitutions, computing a geometry-derived center estimate that conditions the learned 9-DoF prediction. Extensive experiments demonstrate that our framework mitigates scale ambiguity and improves 3D localization robustness across varying camera scales.

AI USE STATEMENT

In this work, we used generative AI tools for implementing parts of our methods, and polishing writing. We have not used generative AI tools for generating synthetic data sets, design or provide feedback on research methodology or experiments, and interpret result. Developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, assisting with translation, cleaning and reformatting dataset, and supporting qualitative and thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work. Code generated by LLM was verified and tested for correctness by authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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.

Anthropic. Claude Opus 4.7. `https://www.anthropic.com/claude/opus`, 2026a. Accessed: 2026-05-06.

Anthropic. Claude Sonnet 4.6. `https://www.anthropic.com/claude/sonnet`, 2026b. Accessed: 2026-05-06.

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

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

Zhongang Cai, Ruisi Wang, Chenyang Gu, Fanyi Pu, Junxiang Xu, Yubo Wang, Wanqi Yin, Zhitao Yang, Chen Wei, Qingping Sun, et al. Scaling spatial intelligence with multimodal foundation models. *arXiv preprint arXiv:2511.13719*, 2025.

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

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision*, pp. 202–221. Springer, 2020.

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26428–26438, 2024.

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

An-Chieh Cheng, Yang Fu, Yatai Ji, Ligeng Zhu, Guanqi Zhan, Zhuoyang Zhang, Zhaojing Yang, Song Han, Yao Lu, Pavlo Molchanov, Vidya Nariyambut Murali, Jan Kautz, Xiaolong Wang, Hongxu Yin, and Sifei Liu. Grounded 3D-aware spatial vision-language modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017.

Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, and Peter Grasch. MM-Spatial: Exploring 3D spatial understanding in multimodal LLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

DeepSeek-AI. DeepSeek-V4. https://www.deepseek.com/, 2026. Accessed: 2026-05-06.

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

Jie Feng, Jinwei Zeng, Qingyue Long, Hongyi Chen, Jie Zhao, Yanxin Xi, Zhilun Zhou, Yuan Yuan, Shengyuan Wang, Qingbin Zeng, et al. A survey of large language model-powered spatial intelligence across scales: Advances in embodied agents, smart cities, and earth science. *arXiv preprint arXiv:2504.09848*, 2025.

Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. Refocus: Visual editing as a chain of thought for structured image understanding. *International Conference on Machine Learning*, 2025.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Zhe Xu, Yao Hu, Shaohui Lin, et al. Vision-r1: Incentivizing reasoning capability in multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 63794–63812, 2026.

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

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, 2016.

Zenan Li, Zhi Zhou, Yuan Yao, Yu-Feng Li, Chun Cao, Fan Yang, Xian Zhang, and Xiaoxing Ma. Neuro-symbolic data generation for math reasoning. *Advances in Neural Information Processing Systems*, 37: 23488–23515, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

Yang Liu, Ming Ma, Xiaomin Yu, Pengxiang Ding, Han Zhao, Mingyang Sun, Siteng Huang, and Donglin Wang. Ssr: Enhancing depth perception in vision-language models via rationale-guided spatial reasoning. *Advances in Neural Information Processing Systems*, 2025.

Moonshot AI. Kimi K2.6. https://www.kimi.com/blog/kimi-k2-6, 2026. Accessed: 2026-05-06.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

OpenAI. GPT-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026a. Accessed: 2026-05-06.

OpenAI. GPT-6 Astra. https://developers.openai.com/api/docs/models/gpt-6-astra, 2026b. Accessed September 24, 2026.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3806–3824, 2023.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Qwen Team. Qwen3.6-Flash. https://qwen.ai/blog?id=qwen3.6-35b-a3b, 2026. Accessed: 2026-05-06.

Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael J Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning. *arXiv preprint arXiv:2505.23678*, 2025.

Seed Team. Seed-2.0-Pro. https://research.doubao.com/en/seed2, 2026. Accessed: 2026-05-06.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. *Advances in Neural Information Processing Systems*, 38:8222–8251, 2026.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025a.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025b.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11888–11898, 2023.

Shi-Yu Tian, Zhi Zhou, Kun-Yang Yu, Ming Yang, Yang Chen, Ziqiao Shang, Lan-Zhe Guo, and Yu-Feng Li. Last: Leveraging tools as hints to enhance spatial reasoning for multimodal large language models. *arXiv preprint arXiv:2604.09712*, 2026.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024a.

Shuai Wang, Daoan Zhang, Tianyi Bai, Shitong Shao, Jiebo Luo, and Jiaheng Wei. Last: Learning to think in space and time for generalist vision-language models. *arXiv preprint arXiv:2511.19261*, 2025.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DTQIjngDta.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024b.

Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *Advances in Neural Information Processing Systems*, 2025.

xAI. Grok-1.5 Vision Preview. https://x.ai/news/grok-1.5v, 2024. Introduces the RealWorldQA benchmark.

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

Tianyun Yang, Yunwen Li, Ziniu Li, Zhihang Lin, Ruoyu Sun, and Tian Ding. Bridging formal language with chain-of-thought reasoning to geometry problem solving. *arXiv preprint arXiv:2508.09099*, 2025b.

Songsong Yu, Yuxin Chen, Hao Ju, Lianjie Jia, Fuxi Zhang, Shaofei Huang, Yuhan Wu, Rundi Cui, Binghao Ran, Zaibin Zhang, et al. How far are vlms from visual spatial intelligence? a benchmark-driven perspective. *arXiv preprint arXiv:2509.18905*, 2025.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of ACL*, 2025.

Gongjie Zhang, Wenhao Li, Quanhao Qian, Jiuniu Wang, Deli Zhao, Shijian Lu, and Ran Xu. On the generalization capacities of mllms for spatial intelligence. *International Conference on Learning Representations*, 2026a.

Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *Advances in Neural Information Processing Systems*, 2025.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, et al. Thyme: Think beyond images. *International Conference on Learning Representations*, 2026b.

Zaibin Zhang, Yuhan Wu, Lianjie Jia, Yifan Wang, Zhongbo Zhang, Yijiang Li, Binghao Ran, Fuxi Zhang, Zhuohan Sun, Zhenfei Yin, et al. Think3d: Thinking with space for spatial reasoning. *arXiv preprint arXiv:2601.13029*, 2026c.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *Advances in Neural Information Processing Systems*, 2025a.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8995–9006, 2025b.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *International Conference on Learning Representations*, 2026.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *Advances in Neural Information Processing Systems*, 2025a.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023.

Zetong Zhou, Dongping Chen, Zixian Ma, Zhihan Hu, Mingyang Fu, Sinan Wang, Yao Wan, Zhou Zhao, and Ranjay Krishna. Reinforced visual perception with tools. *arXiv preprint arXiv:2509.01656*, 2025b.

A Appendix

A.1 Experimental Settings and Details

Implementation Details. We conduct experiments on 8 NVIDIA H100 GPUs with 80 GB of memory. Our model is trained for 3 epochs with an effective global batch size of 64, achieved through a per-GPU batch size of 1 and 8 gradient accumulation steps, requiring a total training time of approximately 3 hours. We use the AdamW optimizer with an initial learning rate of 1×10^{-5} and a weight decay of 0.01. We apply a cosine learning rate scheduler with a warm-up ratio of 0.03. We employ Qwen3-VL-8B-Instruct (Bai et al., 2025) as our Multimodal Large Language Model backbone. For each task, we separately fine-tune the Qwen3-VL baseline using only the final 3D bounding boxes as supervision and our model using equation-anchored CoT traces. During training, we fully fine-tune the language model while keeping the vision tower and multimodal projector frozen. All experiments are implemented using the PyTorch framework, integrating DeepSpeed ZeRO-2, gradient checkpointing, and bf16 mixed precision for efficient training.

Code Availability. At the time of our evaluation, among the representative tool-augmented methods like LAST (Tian et al., 2026), Think3D (Zhang et al., 2026c), MM-Spatial (Daxberger et al., 2025), and GR3D (Cheng et al., 2026), only Think3D provided publicly available code. We adapt Think3D to our single-image setting and report its results in Tables 8 and 9.

Context Length and Token Budget. During SFT, we set the tokenizer maximum sequence length to 8,192 tokens. During evaluation, we set it to 12,800 tokens and cap each model turn at 4,096 newly generated tokens. Each interaction runs for at most five model turns.

Supervision of 2D Box Prediction. We do not use a separate 2D detector or an auxiliary box-regression loss. During the construction of the SFT traces, each ground-truth 3D box is projected onto the image plane using the corresponding camera calibration, and the axis-aligned envelope of its projected corners is used as the 2D box target. The box is represented both in the $[0, 1000]$ normalized coordinate space of Qwen3-VL and in absolute pixel coordinates. Both representations are included in the first assistant turn, and the model is trained to generate the assistant response, including these 2D coordinates. During training, subsequent depth-tool calls are constructed using these projected boxes. At inference time, no ground-truth 2D boxes or external 2D detector are provided; the model generates the boxes from the RGB image and task instruction, and the generated pixel-coordinate boxes are passed to the depth-sampling tool.

A.2 Additional Experimental Results

Prompting Protocol for Generalist MLLM Baselines. The generalist MLLM baselines in Tables 1 and 2 receive only the RGB image and task instruction, together with the required 9-DoF output format. They do not receive predicted 2D boxes, camera intrinsics, depth samples, the back-projection equation, or access to the camera and depth tools. These results therefore serve as RGB-only baselines rather than same-information comparisons. We further evaluate the effect of directly providing geometric information to Qwen3-VL-8B-Instruct without task-specific SFT in Table 6. All three settings receive the same RGB images, task instructions, and 2D boxes. We then additionally provide the camera intrinsics and metric depth samples used by our method, followed by the explicit pinhole back-projection equation. Providing the tool outputs reduces detection performance from 16.02 to 9.73 while leaving grounding performance nearly unchanged. Adding the equation improves detection to 11.78 but reduces grounding to 42.71. These results show that simply including geometric tool outputs or an explicit equation in the prompt does not consistently improve 3D localization, highlighting the importance of learning to use this information through equation-anchored CoT supervision.

Geometric Prompting with GPT-6. Table 7 compares three prompting settings for GPT-6 Astra without task-specific SFT, all using the same predicted 2D boxes. Providing camera intrinsics and metric depth samples alone yields lower scores on both tasks. Adding the pinhole back-projection equation and calculation instructions raises both scores, but detection remains below the first setting while grounding achieves the highest score. These results indicate that directly supplying geometric information and equation instructions does not consistently improve localization across tasks and remains insufficient compared to our equation-anchored CoT-trained model.

Comparison with 4B Models. Tables 8 and 9 compare our framework using Qwen3-VL-4B-Instruct with the corresponding Qwen3-VL baseline and a single-view adaptation of Think3D (Zhang et al., 2026c) on both tasks across camera rescale factors. For our method and the Qwen3-VL baseline, the training and evaluation settings follow those of the 8B experiments, except for the backbone size. We adapt the public Think3D implementation using Qwen3-VL-4B-Instruct and Pi3 (Wang et al., 2026). Given a single RGB image and the task instruction, this baseline can request up to three rendered views from the Pi3 reconstruction

Table 6: Effect of geometric tool outputs and the pinhole back-projection equation on Qwen3-VL-8B-Instruct without task-specific SFT. All settings use the same RGB images, task instructions, and predicted 2D boxes at a camera rescale factor of $1.0\times$. Geometric tool outputs refer to camera intrinsics and ground-truth metric depth samples. The final setting additionally includes the pinhole back-projection equation and calculation instructions. Detection and grounding are evaluated using F1@0.25 and Acc@0.25, respectively.

Geometric Tool Outputs	Pinhole Equation	3D Object Detection F1@0.25	3D Visual Grounding Acc@0.25
–	–	**16.02**	45.89
✓	–	9.73	**46.15**
✓	✓	11.78	42.71

Table 7: Effect of geometric tool outputs and the pinhole back-projection equation on GPT-6 Astra without task-specific SFT. All settings use the same RGB images, task instructions, and predicted 2D boxes and labels at a camera rescale factor of $1.0\times$. Geometric tool outputs refer to camera intrinsics and ground-truth metric depth samples. The final setting additionally includes the pinhole back-projection equation and calculation instructions. Detection and grounding are evaluated using F1@0.25 and Acc@0.25, respectively.

Geometric Tool Outputs	Pinhole Equation	3D Object Detection F1@0.25	3D Visual Grounding Acc@0.25
–	–	**16.44**	26.53
✓	–	8.37	24.67
✓	✓	10.30	**31.03**

before predicting the required 9-DoF boxes. We generate task-specific SFT trajectories from the corresponding training splits using ground-truth 3D boxes as the final answer targets. 3D Detection and grounding models are initialized independently from Qwen3-VL-4B-Instruct and trained for three epochs. We report the final SFT checkpoints without reinforcement learning. Both variants of our method outperform the adapted Think3D at all camera scales on both tasks. At a rescale factor of $1.5\times$, our method with predicted depth achieves 21.59 F1@0.25 in detection and 27.59 Acc@0.25 in grounding, compared with 14.81 and 21.22 for Think3D, respectively. These results demonstrate the effectiveness of our framework relative to the adapted spatial tool-use baseline and, together with the 8B results, show its applicability to backbones of different model sizes.

Cross-Dataset Transfer with LoRA. We adapt our ScanNet-trained model using LoRA on 5,000 additional ScanNet examples resized according to the camera intrinsics and image resolutions of SUN RGB-D, without using SUN RGB-D images or 3D annotations for fine-tuning. Table 10(a) reports the results with predicted and ground-truth depth. SUN RGB-D has pixel focal lengths of approximately 519–570 pixels, compared with 1160–1178 pixels for ScanNet. For reference, Table 10(b) reports the original ScanNet-trained model's results at $0.5\times$ and $1.0\times$, corresponding to approximately 580–589 and 1160–1178 pixels, respectively. These results support our framework's ability to transfer across real-world camera setups after lightweight adaptation using intrinsic-adjusted ScanNet data, while maintaining good 3D localization performance on ScanNet.

General Ability Evaluation. Table 11 evaluates general abilities of Qwen3-VL-8B-Instruct and our model after supervised fine-tuning for 3D object detection. Our model maintains comparable performance on six benchmarks.

A.3 Construction of the Single-Frame Visual Grounding Dataset

To utilize the multi-view ScanRefer dataset for our single-frame 3D visual grounding task, we develop an automated data generation and verification pipeline to construct high-quality image-text pairs.

Single-Frame Extraction and 2D Projection. For each annotated 3D object in ScanRefer, we extract the target RGB frame. Utilizing the provided camera intrinsics and poses, we project the ground-truth 3D bounding box onto the 2D image plane to obtain a 2D bounding box. This 2D bounding box is explicitly drawn on the image to serve as a precise visual prompt, guiding the MLLM to focus on the target object.

Table 8: Comparison of 4B models on 3D object detection across camera rescale factors. Results are reported as F1@0.25 (%). Best in **bold**, second underlined.

Method	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
Qwen3-VL-4B-Instruct (Bai et al., 2025)	9.22	15.43	22.34	30.07	38.45	40.70	31.86	27.38	21.91	21.27	16.57
Think3D-4B (single-view, w/ SFT) (Zhang et al., 2026c)	12.98	20.36	24.39	28.49	32.18	35.53	28.05	24.58	19.89	17.15	14.81
Ours-4B (w/ UniDepthV2 (Piccinelli et al., 2025))	14.59	21.41	33.86	34.23	**38.97**	40.18	37.01	33.66	29.70	24.56	21.59
Ours-4B (w/ GT depth)	**16.15**	**22.96**	**37.10**	**35.66**	36.24	**47.43**	**38.36**	**34.85**	**30.39**	**28.43**	**24.82**

Table 9: Comparison of 4B models on 3D visual grounding across camera rescale factors. Results are reported as Acc@0.25 (%). Best in **bold**, second underlined.

Method	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
Qwen3-VL-4B-Instruct (Bai et al., 2025)	6.37	16.98	27.85	42.44	52.52	**57.03**	50.66	42.97	31.83	24.14	20.16
Think3D-4B (single-view, w/ SFT) (Zhang et al., 2026c)	4.51	13.00	25.46	37.93	52.25	54.11	49.34	40.85	30.24	20.42	21.22
Ours-4B (w/ UniDepthV2 (Piccinelli et al., 2025))	**12.73**	20.95	**39.79**	47.75	52.52	54.64	52.79	47.75	40.05	35.01	27.59
Ours-4B (w/ GT depth)	11.67	**23.08**	38.99	**49.60**	**56.50**	**57.03**	**55.44**	**48.01**	**44.83**	**37.67**	**31.30**

Automated Referring Expression Generation. Since the original ScanRefer text descriptions often rely on multi-view temporal context, we employ a powerful MLLM (Qwen3.5-Plus) to regenerate single-frame referring expressions. We prompt the MLLM with the 2D bounding box highlighted image and instruct it to generate a concise description detailing the object's category, visual attributes (e.g., color, material), and spatial relationships with adjacent structures. To strictly prevent category hallucinations during generation, we explicitly inject the ground-truth object name into the system prompt as a hard constraint. We set the generation temperature to $T = 0.7$ to encourage linguistic diversity in the resulting expressions.

Verification and Filtering via LLM-as-a-Judge. To ensure the quality of the generated dataset, we design a verification stage. We feed the newly generated referring expression alongside the highlighted image into the MLLM. Operating as a strict evaluator at a low temperature ($T = 0.1$ for deterministic judgment), the MLLM classifies the expression into one of three verdicts: *accurate*, *inaccurate* (containing factual errors), or *ambiguous*. Importantly, the *ambiguous* category identifies descriptions that are with correct content but fail to uniquely pinpoint the target object among similar instances in the scene. Discarding these ambiguous cases is essential, as they introduce noisy supervision signals that hinder robust 3D localization. Finally, only the sample pairs verified as strictly *accurate* are retained to formulate our final visual grounding dataset.

A.4 Limitations and Future Work

As shown in Table 1 and Table 2, while our framework improves across a wide range of scales, it exhibits performance drops at extreme rescale factors ($0.5\times$–$0.7\times$). We attribute this to the fragility of the external 2D segmentor and monocular depth estimator. Extreme intrinsic rescaling breaks implicit visual priors, causing models like SAM3 and UniDepthV2 to return inaccurate segmentation masks and metric depths. Because these tool outputs enter the center computation, perception errors perturb the intermediate estimate and may degrade the downstream 9-DoF prediction. Future work will explore camera-intrinsic-aware 2D segmentation models and depth estimators or empower the MLLM to self-correct inaccurate tool estimations.

A.5 System Prompts

Figures 4 and 5 show key components of the system prompts used for both 3D object detection and 3D visual grounding tasks.

A.6 Visualization Results

To further validate the camera-robustness and spatial precision of our proposed framework, we provide comprehensive qualitative comparisons against Qwen3-VL-8B-Instruct and VG LLM on both 3D object detection and 3D visual grounding tasks, as shown in Figures 6, 7, and 8.

We can observe from the visualizations that our method exhibits camera-robustness across different rescale factors. By explicitly using the rescaled camera intrinsics and retrieved metric depths to compute a camera-conditioned center estimate, our agent mitigates visual-scale ambiguity across the evaluated rescaling settings, illustrating the benefit of moving from purely implicit perception toward mathematically grounded tool use for robust 3D localization.

Table 10: 3D object detection results on ScanNet and SUN RGB-D after LoRA adaptation, with ScanNet results from the main experiments included for comparison. (a) Ours-8B after LoRA adaptation on 5,000 intrinsic-adjusted ScanNet examples. (b) Ours-8B results from Table 1 at $0.5\times$ and $1.0\times$, without additional LoRA adaptation. Results are F1@0.25 (%), averaged over 31 classes for ScanNet and 82 classes for SUN RGB-D. Predicted depth is obtained from UniDepthV2.

(a) Results after LoRA adaptation

Dataset	Pixel focal length f_x (px)	Pred. depth	GT depth
SUN RGB-D	518.9–570.3	14.33	19.53
ScanNet (rescale factor=1.0)	1160.4–1178.4	39.43	40.34

(b) ScanNet results from Table 1

Dataset	Pixel focal length f_x (px)	Pred. depth	GT depth
ScanNet (rescale factor=0.5)	580.2–589.2	16.97	18.93
ScanNet (rescale factor=1.0)	1160.4–1178.4	44.13	44.90

Table 11: General benchmark performance of Qwen3-VL-8B-Instruct and our model after supervised fine-tuning for 3D object detection. All scores are reported as percentages. Average denotes the unweighted mean across the six benchmarks.

Benchmark	Qwen3-VL-8B-Instruct	Ours-8B
MMMU-Pro (Yue et al., 2025)	28.61	28.79
MathVision (Wang et al., 2024a)	39.61	35.82
MMLU-Pro (Wang et al., 2024b)	66.77	44.63
RealWorldQA (xAI, 2024)	69.41	70.98
AI2D (Kembhavi et al., 2016)	83.65	83.58
IFEval (Zhou et al., 2023)	79.48	78.37
Average	61.26	57.03

A.7 Failure Case Analysis

Figure 9 illustrates two typical failure cases where the geometric deduction chain performs poorly due to inaccurate 2D segmentation generated by the Segment Anything Model 3 (SAM3). In Case 1 (top), the framework attempts to localize a couch and a table. The intermediate outputs reveal that SAM3 fails to generate tight object contours. The blue and red masks, intended for the couch and the table respectively, fail to cover their respective target objects. Instead, they wrongly cover large portions of the black background regions. Similarly, in Case 2 (bottom), the framework aims to localize a wardrobe and a cabinet. The generated 2D masks by SAM3 exhibit low quality. Rather than wrapping the visual features of the furniture, the blue mask (wardrobe) and green mask (cabinet) capture the background black regions. These failure cases highlight the importance of the quality of intermediate variables. Moreover, this also demonstrates the transparency of our approach, as errors can be explicitly traced to specific intermediate tool failures.

Our System Prompt for 3D Object Detection

```
You are an expert 3D Computer Vision Assistant capable of utilizing external tools to perform precise 3D object detection for multiple objects in a scene.

You MUST think step by step and call tools if needed before generating the final 3D bounding boxes.

## Core Instructions & Constraints
1. **Self-Predicted 2D Detection**: When 2D bounding boxes are needed in the process of predicting 3D bounding boxes step by step, you must **NOT** call a tool to detect 2D bounding boxes. You possess internal visual perception capabilities. You are required to:
   * First, predict the 2D bounding boxes in **normalized coordinates** $[0, 1000]$.
   * Second, mathematically **denormalize** them to absolute pixel coordinates based on the image resolution.
2. **Tool Usage**: Only use the specific tools listed below for camera parameters and depth sampling.

## Available Tools
You have access to the following specific tools. You must not fabricate tools or use tools not listed here.
[
  {
    "name": "get_camera_info",
    "description": "Retrieves the camera's intrinsic matrices required for 3D projection.",
    "parameters": {
      "type": "object",
      "properties": {},
      "required": []
    }
  },
  {
    "name": "sample_depth_points",
    "description": "Randomly samples multiple depth points within the 2D bounding box areas of the specified objects. Returns a list of objects, where each object contains the original 'label', 'bbox_2d', and a 'depth_samples' list. Each entry in 'depth_samples' is in the format [u, v, depth], where 'u' and 'v' are the absolute pixel coordinates and 'depth' is the physical distance in meters.",
    "parameters": {
      "type": "object",
      "properties": {
        "object_2d_bbox_list": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "label": {
                "type": "string",
                "description": "The category label of the object."
              },
              "bbox_2d": {
                "type": "array",
                "items": {
                  "type": "integer"
                },
                "description": "The absolute 2D bounding box [xmin, ymin, xmax, ymax]."
              }
            },
            "required": ["label", "bbox_2d"]
          },
          "description": "A list of objects containing their category labels and 2D bounding boxes for depth sampling."
        }
      },
      "required": ["object_2d_bbox_list"]
    }
  }
]
```

Figure 4: The system prompt for the 3D object detection task in our method.

Our System Prompt for 3D Visual Grounding

You are an expert 3D Computer Vision Assistant capable of utilizing external tools to perform precise 3D visual grounding — localizing a single object described by a natural language text in 3D space.
You MUST think step by step and call tools if needed before generating the final 3D bounding box.

Core Instructions & Constraints
1. **Self-Predicted 2D Detection**: When 2D bounding boxes are needed in the process of predicting 3D bounding boxes step by step, you must **NOT** call a tool to detect 2D bounding boxes. You possess internal visual perception capabilities. You are required to:
   * First, predict the 2D bounding box in **normalized coordinates** $[0, 1000]$.
   * Second, mathematically **denormalize** it to absolute pixel coordinates based on the image resolution.
2. **Tool Usage**: Only use the specific tools listed below for camera parameters and depth sampling.

Available Tools
You have access to the following specific tools. You must not fabricate tools or use tools not listed here.

```
[
  {
    "name": "get_camera_info",
    "description": "Retrieves the camera's intrinsic matrices required for 3D projection.",
    "parameters": {
      "type": "object",
      "properties": {},
      "required": []
    }
  },
  {
    "name": "sample_depth_points",
    "description": "Randomly samples multiple depth points within the 2D bounding box area of the specified object. Returns a list containing the original 'label', 'bbox_2d', and a 'depth_samples' list. Each entry in 'depth_samples' is in the format [u, v, depth], where 'u' and 'v' are the absolute pixel coordinates and 'depth' is the physical distance in meters.",
    "parameters": {
      "type": "object",
      "properties": {
        "object_2d_bbox_list": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "label": {
                "type": "string",
                "description": "The category label of the object."
              },
              "bbox_2d": {
                "type": "array",
                "items": {
                  "type": "integer"
                },
                "description": "The absolute 2D bounding box [xmin, ymin, xmax, ymax]."
              }
            },
            "required": ["label", "bbox_2d"]
          },
          "description": "A list containing the object's category label and 2D bounding box for depth sampling."
        }
      },
      "required": ["object_2d_bbox_list"]
    }
  }
]
```

Figure 5: The system prompt for the 3D visual grounding task in our method.

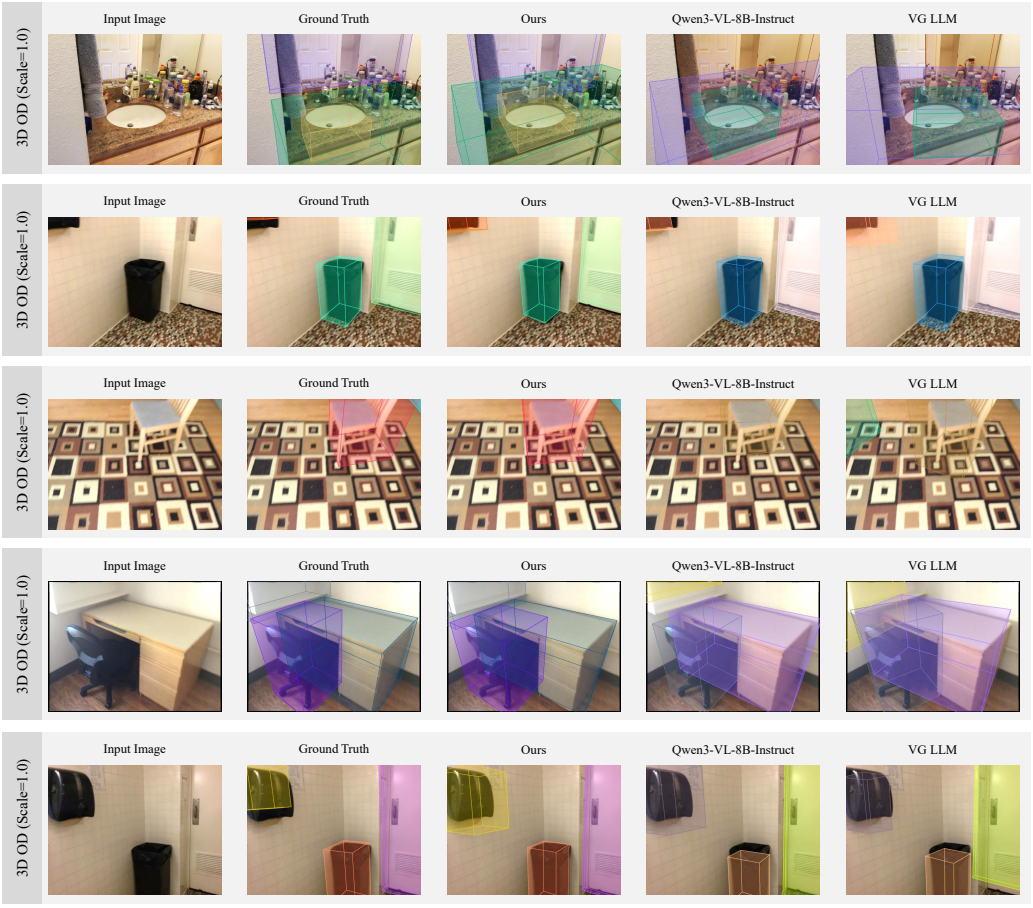

Figure 6: Qualitative visualization of 3D object detection results under camera rescale factors of 1.0×.

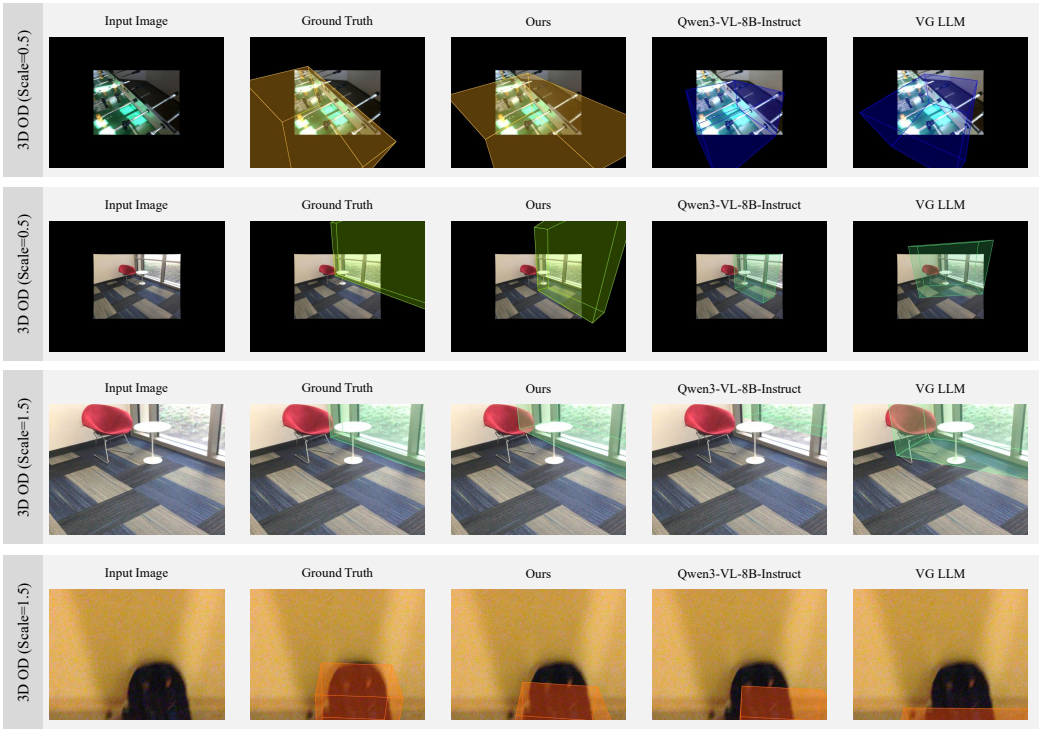

Figure 7: Qualitative visualization of 3D object detection results under camera rescale factors of $0.5\times$ and $1.5\times$.

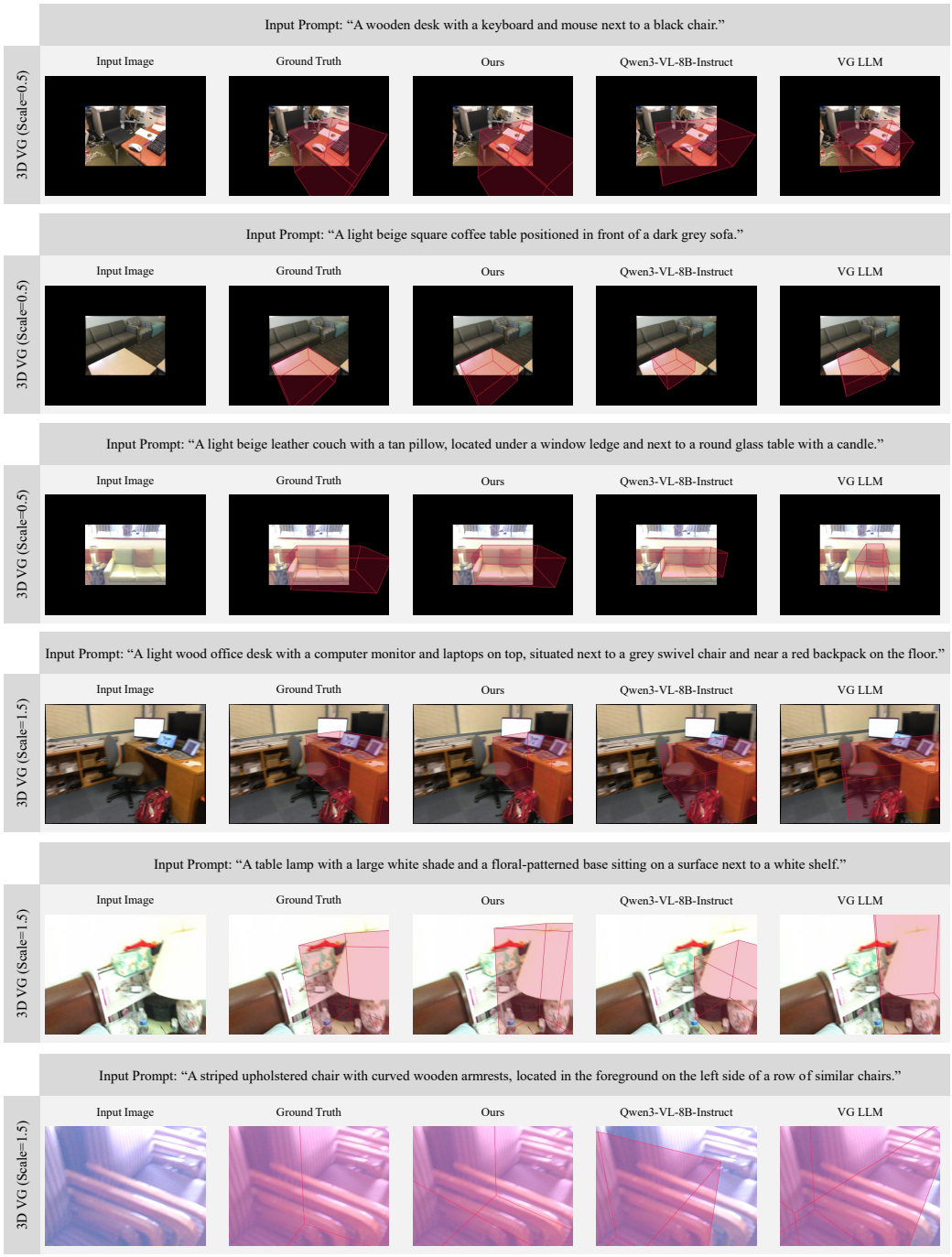

Figure 8: Qualitative visualization of 3D visual grounding results under camera rescale factors of $0.5\times$, $1.0\times$, and $1.5\times$.

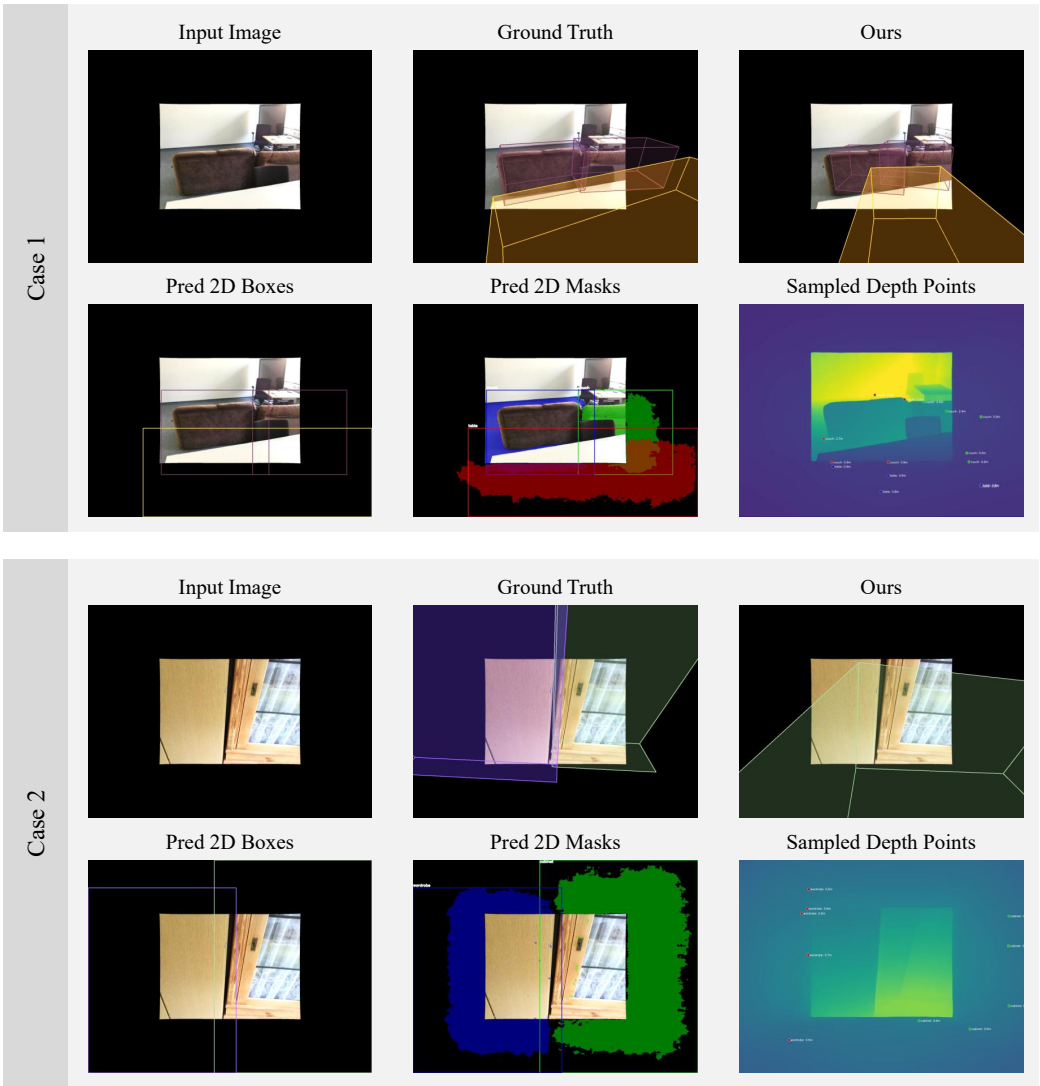

Figure 9: Failure cases.