TRAINING-FREE REFERENTIAL MONOCULAR 3D OBJECT LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We study referential monocular 3D object localization (RM3DOL): localizing a language-referred object as a metric camera-frame 9-DoF box from a single RGB image. The task couples fine-grained grounding with metric recovery under monocular depth and scale ambiguity. We propose a training-free framework that grounds the target semantically, independently reconstructs a gravity-aligned metric representation, and combines both outputs to recover position, dimensions, and orientation. We also introduce Ref3DOL, an indoor benchmark built from Omni3D-unified SUN RGB-D and Hypersim annotations. Its automated progressive-disambiguation pipeline uses category, appearance, and spatial cues to generate instance-specific queries without manual expression annotation. The category- and instance-level tracks contain 33,915 and 41,875 queries, respectively. Across multiple VLM grounding backbones, our framework achieves the best overall performance among the evaluated direct 3D-box prediction baselines, including Qwen3-family models and GPT-5.6 Sol. Code and data are available at `https://anonymous.4open.science/r/TrainingFreeRM3DOL-9744/README.md`.

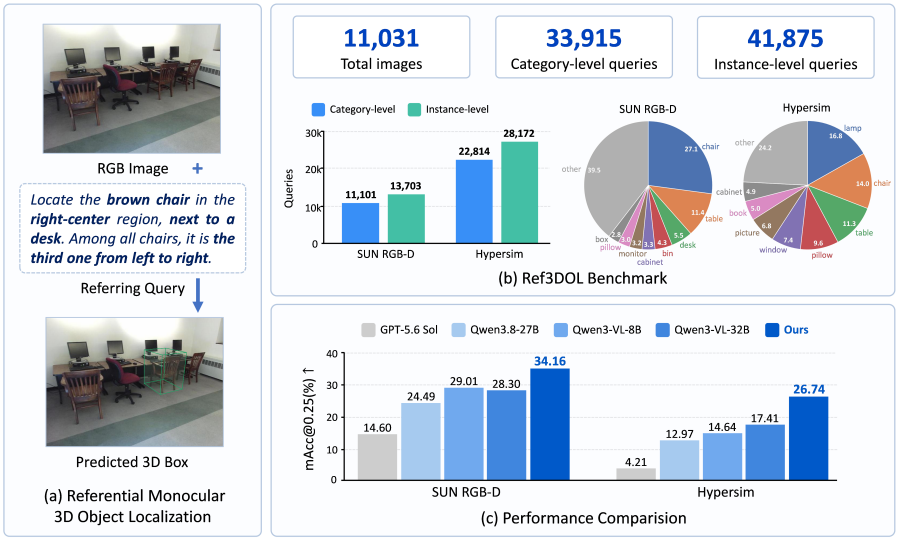

Figure 1: Overview of (a) RM3DOL input and output, (b) Ref3DOL image/query statistics and category distributions, with pie charts showing the eight most frequent categories, and (c) instance-level mAcc @0.25 comparisons on SUN RGB-D and Hypersim.

1 INTRODUCTION

Referential monocular 3D object localization (RM3DOL) aims to recover a language-referred object as a metric 9-DoF box in the camera coordinate frame from a single RGB image (Bai et al., 2025a; Man et al., 2025). The query identifies the target through category, appearance, or spatial cues but provides no metric 3D information. The model must therefore both distinguish the referred instance

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.

from visually similar objects and infer its 3D position, physical dimensions, and orientation. This metric representation connects language-based target selection with spatial reasoning and physical interaction without requiring depth sensors or prebuilt 3D maps. The task remains challenging because a single image provides no direct observation of depth or absolute scale, while the grounding strengths of vision–language models (VLMs) largely operate in the image plane and do not by themselves resolve metric geometry.

Recent work has extended VLMs to 3D perception. Qwen3-VL (Bai et al., 2025a) directly predicts 3D boxes after dedicated training, while LocateAnything3D (Man et al., 2025) uses structured next-token prediction and Chain-of-Sight reasoning to progress from 2D localization to 3D parameters. These advances provide a basis for adapting VLMs to RM3DOL, but rely on large-scale paired 2D–3D data and task-specific post-training. The resulting data and computational costs limit rapid adaptation to new models and scenes. This raises a question: *Can we achieve reliable RM3DOL without task-specific training?*

Addressing this question presents challenges in both data and methodology. On the data side, we are not aware of a publicly released benchmark matching our exact protocol, limiting reproducible evaluation and further development. General monocular 3D detection benchmarks such as Omni3D (Brazil et al., 2023) provide category-labeled 3D boxes but lack natural-language queries for instance-specific localization. Category labels alone cannot distinguish a target from other instances of the same category in an image, while manually annotating instance-specific queries is costly. On the methodological side, it is difficult for a single model to unify natural-language understanding, visual grounding, and metric 3D perception without task-specific training. Semantic grounding and metric geometric reconstruction require fundamentally different capabilities: semantic grounding must interpret linguistic attributes and spatial relations, resolve same-category distractors, and align the query with target-specific image regions, whereas geometric reconstruction must infer absolute depth and physical scale, establish a gravity-aware metric coordinate frame, and recover a physically consistent 3D box. Combining these heterogeneous requirements within a single training-free predictor is therefore particularly difficult.

To address the data challenge, we introduce Ref3DOL, a benchmark for RM3DOL built using the camera-centric 3D annotations of SUN RGB-D and Hypersim unified by Omni3D (Song et al., 2015; Roberts et al., 2021; Brazil et al., 2023). We develop an automated pipeline that combines visual and spatial cues to convert existing detection annotations into instance-specific natural-language queries. The benchmark comprises two complementary tracks: category-level localization, which accepts any instance of the queried category and evaluates category recognition and metric 3D recovery, and instance-level localization, which requires grounding a query to a unique target and additionally evaluates fine-grained semantic disambiguation. Overall, Ref3DOL covers 11,031 images, with 33,915 category-level queries and 41,875 instance-level queries.

To address the methodological challenge, we adopt a decoupled semantic–geometric design. A semantic grounding branch resolves the referring expression into target-specific visual evidence, while a geometry reconstruction branch independently establishes a metric, gravity-aware representation of the scene. Their outputs are coupled only during target reconstruction, where the referred object's geometry is isolated and transformed into a camera-frame 9-DoF box. This separation allows each branch to focus on its corresponding capability and replaces direct language-to-3D-box regression with a structured composition of semantic grounding and geometric reasoning, requiring neither RM3DOL-specific paired supervision nor parameter updates.

Our contributions are threefold:

1. We propose a training-free RM3DOL framework that decouples semantic grounding from metric geometric reconstruction to recover a camera-frame 9-DoF box from a single RGB image, without RM3DOL-specific supervision or parameter updates.
2. We introduce Ref3DOL, a benchmark with category- and instance-level tracks that automatically generates referring queries from Omni3D-unified SUN RGB-D and Hypersim annotations using appearance and spatial cues.
3. Experiments across multiple VLM grounding backbones show that our framework outperforms the evaluated direct 3D-box predictors overall, including 3D-trained Qwen3-VL variants and GPT-5.6 Sol.

2 Related Work

Multimodal 3D object localization. Language-guided 3D localization has been studied with different forms of visual input. Point-cloud 3D visual grounding methods, such as ScanRefer and ReferIt3D, identify referred objects in reconstructed 3D scenes (Chen et al., 2020; Achlioptas et al., 2020). Monocular 3D visual grounding instead recovers metric target geometry from a single RGB image and referring descriptions containing appearance and spatial cues. Mono3DVG introduced this setting, and subsequent methods extend it through multi-object grounding, enhanced spatial-language modeling, or VLM-based prediction (Zhan et al., 2024; Guo et al., 2025; Li et al., 2026; Qu et al., 2026). Related monocular 3D detection methods localize category-specified objects and estimate metric 3D boxes from single images, including MonoFlex, MonoDETR, and Cube R-CNN (Zhang et al., 2021; 2023; Brazil et al., 2023). More recently, language–image models have been extended toward direct 3D perception: Cube-LLM learns 3D understanding from multimodal supervision, Qwen3-VL directly predicts 3D boxes after dedicated training, and LocateAnything3D uses structured next-token prediction and Chain-of-Sight reasoning (Cho et al., 2025; Bai et al., 2025a; Man et al., 2025). In contrast, we decouple language-based target identification from metric reconstruction and require neither RM3DOL-specific paired supervision nor task-specific parameter updates.

Monocular geometry estimation. Recent monocular depth and geometry models address absolute scale and camera variation through transferable representations, canonical camera transformations, camera conditioning, focal-length estimation, metric-scale geometry prediction, and fine-grained implicit representations (Yang et al., 2024; Hu et al., 2024; Piccinelli et al., 2024; Bochkovskiy et al., 2025; Wang et al., 2025a; Ganesan et al., 2026; Yu et al., 2026). Single-image calibration provides complementary cues: Perspective Fields estimates local perspective fields, GeoCalib jointly recovers camera intrinsics and gravity, and AnyCalib predicts per-pixel viewing rays (Jin et al., 2023; Veicht et al., 2024; Tirado-Garín & Civera, 2025). We compose these pretrained geometric capabilities rather than introducing new estimators: target pixels select the relevant geometry, the estimated vertical constrains box fitting, and the result is transformed into a camera-frame 9-DoF box.

3 Ref3DOL Benchmark

3.1 Benchmark Design and Task Definition

Ref3DOL is a benchmark for RM3DOL constructed from the test-set 3D annotations of SUN RGB-D and Hypersim unified by Omni3D (Song et al., 2015; Roberts et al., 2021; Brazil et al., 2023). SUN RGB-D and Hypersim provide real and synthetic indoor scenes, respectively. Given a single RGB image I and a natural-language query q, a model predicts the referred object's 3D bounding box in the camera coordinate frame:

$$\hat{B} = f(I, q), \qquad \hat{B} = (c_x, c_y, c_z, w, h, l, \phi, \theta, \psi), \tag{1}$$

where (c_x, c_y, c_z) denotes the box center, (w, h, l) its physical dimensions, and (ϕ, θ, ψ) its roll, pitch, and yaw angles, respectively.

The benchmark contains two complementary tracks. The *category-level* track assigns one query to each valid image–category pair, specifying only the target category and accepting any valid instance of that category. It evaluates category recognition and metric 3D perception. The *instance-level* track uses referring expressions with appearance and spatial cues, each linked to a unique target, to assess fine-grained grounding and 3D localization. Since category-level queries are obtained directly by grouping annotations by image and category, we focus below on constructing instance-level expressions.

3.2 Query Construction via Progressive Disambiguation

When an image contains multiple instances of the same category, the category alone cannot identify the target instance. We therefore progressively reduce the ambiguous candidate set using color, absolute-position, and relative-position cues until the target is uniquely identified.

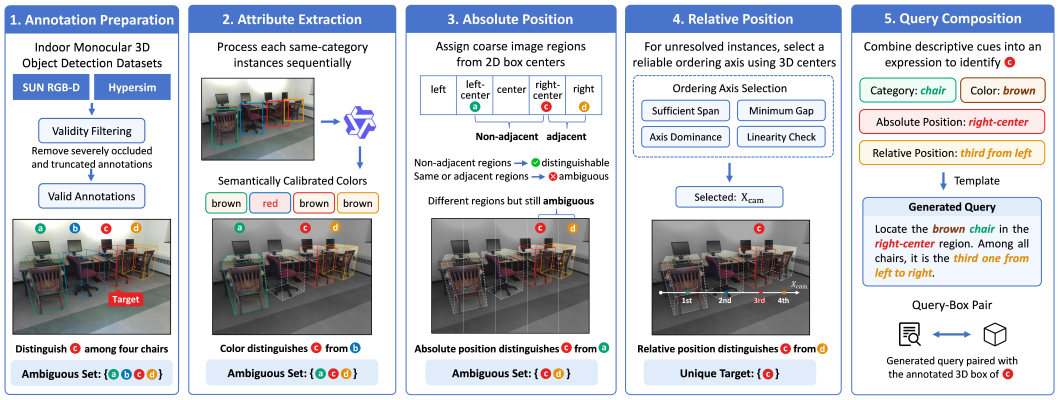

Figure 2: Ref3DOL instance-query construction. Color and coarse position narrow the candidates; reliable camera-frame ordering resolves remaining ambiguity.

Annotation preparation. We collect test-set images and annotations from SUN RGB-D and Hypersim, and follow Omni3D's visibility filtering rules (Brazil et al., 2023) to remove invalid objects, including those that are severely occluded or truncated.

Attribute extraction and semantic calibration. We highlight each valid same-category instance and use Qwen3-VL-32B-Instruct (Bai et al., 2025a) to extract its dominant color and visually salient reference objects. The color descriptions are then semantically clustered, grouping terms such as *brown* and *wood-colored*. We use the calibrated color attributes to disambiguate same-category instances. Reference-object descriptions are retained only as supplementary semantic context and do not participate in candidate-set reduction or determine target uniqueness.

Absolute-position disambiguation with coarse regions. We partition the image horizontally into left, left-center, center, right-center, and right regions using 2D box centers. Absolute position resolves ambiguity only when the target is separated from every same-category distractor by at least one intervening region, avoiding unreliable distinctions near region boundaries.

Relative-position disambiguation through reliable single-axis ordering. For instances that remain ambiguous, we further disambiguate them by ordering their 3D centers along the camera-frame left–right, up–down, and front–back axes. We select an axis using predefined rules that jointly consider group span, minimum adjacent spacing, axis dominance, and linearity. These rules also filter out groups containing too many same-category instances or spatial arrangements that are too complex to describe reliably in a query, ensuring that the resulting relative-position cues are dependable. The selected axis yields expressions such as "third from left to right" or "the nearer one".

Query composition. Finally, we combine the category, retained color attribute, coarse absolute position, optional relative ordering, and reference-object description using a shared template. Each composed query is paired with the target instance and its annotated 3D bounding box.

Further details of the above procedure are provided in Appendix A.

To verify that the generated instance-level queries uniquely identify their intended targets, for each query, we highlight every instance of the queried category in the image with a different randomly assigned color. We then provide the highlighted image and the query to Qwen3-VL-32B-Instruct and ask it to select the instance described by the query. It achieves selection accuracies of 96.70% on SUN RGB-D and 93.08% on Hypersim (94.27% overall). Manual inspection of the incorrect selections confirms that the referring expressions remain accurate. Overall, the results demonstrate that the constructed queries effectively distinguish target instances from same-category distractors.

3.3 Benchmark Statistics

As summarized in Table 1, SUN RGB-D provides real imagery and broader category coverage, whereas Hypersim provides denser synthetic scenes. Together they cover real-image variability and higher object density; instance-specific disambiguation yields more instance-level queries.

Table 1: Dataset-level statistics of Ref3DOL. The instance-level track filters out images and instances that do not satisfy the query-construction criteria.

Dataset	Domain	Images	Categories	Mean categories per image	Mean instances per image	Category queries	Instance queries
SUN RGB-D	Real-world	4,681	82	2.37	2.98	11,101	13,703
Hypersim	Synthetic	6,350	29	3.59	4.55	22,814	28,172
Total	Mixed	**11,031**	**82**	**3.07**	**3.88**	**33,915**	**41,875**

3.4 Evaluation Metrics

Following Omni3D, we use camera-frame 3D intersection over union (3D IoU) and *category-averaged localization accuracy* (mAcc). For category-level queries, s_i is the maximum IoU with any valid same-category box; for instance-level queries, it is the IoU with the referred box. A query succeeds when $s_i \geq \tau$.

Let $\mathcal{K}$ be the evaluated categories and $\mathcal{Q}_c$ the queries for category c. We define

$$\mathrm{mAcc}\,@\tau = \frac{1}{|\mathcal{K}|} \sum_{c \in \mathcal{K}} \frac{1}{|\mathcal{Q}_c|} \sum_{i \in \mathcal{Q}_c} \mathbb{1}[s_i \geq \tau], \tag{2}$$

where $\mathbb{1}[\cdot]$ is the indicator function. The prefix m denotes macro-averaging: accuracy is first computed within each category and then averaged equally across categories, reducing the influence of category-frequency imbalance. We report only $\mathrm{mAcc}\,@0.15$, $\mathrm{mAcc}\,@0.25$, and $\mathrm{mAcc}\,@0.50$ to evaluate localization under increasingly stringent geometric accuracy requirements.

4 Method

Overview. Figure 3 summarizes our training-free framework. A semantic branch converts the query into a target mask, while a geometric branch reconstructs metric 3D points in a gravity-aligned frame. The mask selects target points, which are filtered, fitted with a gravity-constrained box, and transformed into a camera-frame 9-DoF box. All pretrained components remain frozen, requiring no RM3DOL-specific paired supervision or parameter updates.

4.1 Semantic-guided Instance Extraction

Given an image I and query q, a VLM predicts a target box $\hat{B}_{\mathrm{2D}}$. Because it may include background or neighboring objects, we use it to prompt candidate instance masks and select the candidate whose box best overlaps $\hat{B}_{\mathrm{2D}}$.

The resulting mask M connects the branches: semantic grounding selects target pixels, and geometric reconstruction assigns their 3D coordinates, avoiding direct VLM regression of metric box parameters.

4.2 Gravity-guided Monocular Metric Reconstruction

From the input image I, we estimate a metrically scaled depth map D and the camera intrinsic matrix $\mathbf{K}$. Each pixel is then back-projected into the camera coordinate frame, whose X, Y, and Z axes point rightward, downward, and forward along the optical axis, respectively.

Since indoor images are often captured with handheld cameras and thus subject to camera tilt, we use a single-image camera calibration model to estimate the scene's downward gravity unit vector $\mathbf{g}$ in camera coordinates from the RGB image. Using Rodrigues' rotation formula, we construct an alignment rotation $\mathbf{R}_g$ satisfying

$$\mathbf{R}_g \mathbf{g} = -\mathbf{e}_y, \qquad \mathbf{e}_y = (0, 1, 0)^\top. \tag{3}$$

In the resulting gravity-aligned frame, the positive Y axis points upward and the XZ plane is horizontal. Back-projection and alignment give the per-pixel 3D representation

$$\mathbf{p}_g(u, v) = \mathbf{R}_g D(u, v) \mathbf{K}^{-1} [u, v, 1]^\top. \tag{4}$$

Here, (u, v) denotes the horizontal and vertical pixel coordinates, $D(u, v)$ is the metric depth along the camera's optical axis, and $\mathbf{p}_g(u, v)$ is the corresponding 3D point in the gravity-aligned frame.

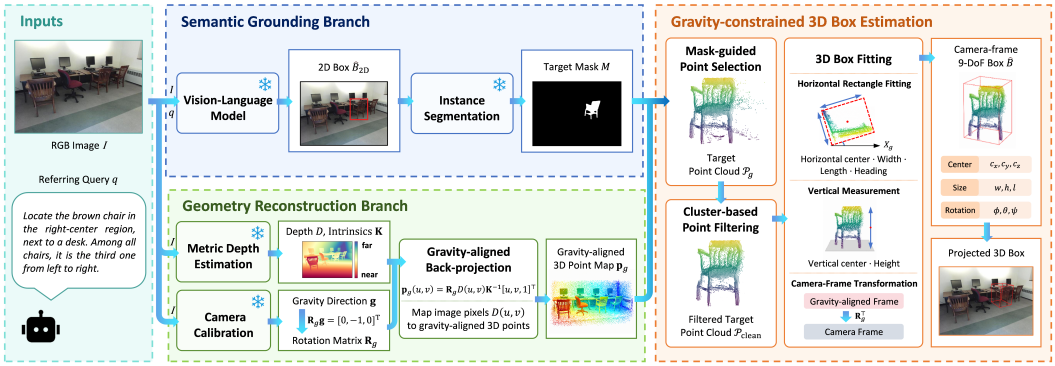

Figure 3: Training-free RM3DOL framework. Semantic grounding produces the target mask, while metric depth, camera intrinsics, and gravity are estimated to reconstruct scene geometry. The selected target points are filtered, fitted with a gravity-constrained box, and transformed into a camera-frame 9-DoF box. Snowflakes denote frozen pretrained models.

We use the target mask to select points from this representation:

$$\mathcal{P}_g = \{\mathbf{p}_g(u,v) \mid M(u,v) = 1,\ (u,v) \in \Omega_{\text{valid}}\}, \tag{5}$$

where Ω_{valid} contains pixels with finite, positive depth and finite reconstructed coordinates. The resulting visible-surface point cloud combines the query-specified target identity with metric geometric observations.

4.3 Cluster-based Target Point Filtering

The mask-selected point cloud $\mathcal{P}_g$ may contain noise from depth and segmentation errors, so we apply cluster-based filtering. We use DBSCAN (Ester et al., 1996) to identify spatial clusters, with adaptive voxelization for large point clouds. Since occlusion may split one instance into multiple clusters, sufficiently large clusters are retained based on their relative sizes. We then select the corresponding points from the original cloud to obtain the cleaned point cloud $\mathcal{P}_{\text{clean}}$ for box fitting.

4.4 Gravity-constrained Box Fitting

Assuming upright indoor objects, we constrain the box's height axis to the estimated scene vertical. We fit a horizontal rectangle, measure the height and vertical center, and transform the resulting box from the gravity-aligned frame back to the camera frame.

Horizontal Rectangle Fitting. We orthogonally project $\mathcal{P}_{\text{clean}}$ onto the horizontal XZ plane and fit an approximate minimum-area enclosing rectangle, which serves as the horizontal cross-section of the 3D box. We search over rotations about the vertical Y axis, with $\mathbf{R}_y(\beta)$ denoting a candidate rotation. For each candidate angle β, define

$$\Delta_j(\beta) = \max_{\mathbf{p}\in\mathcal{P}_{\text{clean}}} [\mathbf{R}_y(-\beta)\mathbf{p}]_j - \min_{\mathbf{p}\in\mathcal{P}_{\text{clean}}} [\mathbf{R}_y(-\beta)\mathbf{p}]_j, \quad j \in \{x, z\}. \tag{6}$$

Here, $[\cdot]_j$ extracts the j-coordinate, and $\Delta_j(\beta)$ is the point cloud's extent along that axis after rotation. We sample candidate headings at a fixed angular step size satisfying $0 \le \beta < \pi/2$ to form the search set $\mathcal{A}$, and select

$$\alpha = \arg\min_{\beta\in\mathcal{A}} \Delta_x(\beta)\Delta_z(\beta). \tag{7}$$

The heading α describes the box's geometric axes, not a semantic front direction.

To obtain the horizontal center, width, and length, we rotate the points into a box-axis-aligned frame while keeping the origin unchanged:

$$\mathcal{P}_{\text{local}} = \{\mathbf{R}_y(-\alpha)\mathbf{p} \mid \mathbf{p} \in \mathcal{P}_{\text{clean}}\}. \tag{8}$$

Let $\mathbf{a}$ and $\mathbf{b}$ be its componentwise minimum and maximum. The horizontal center in the gravity-aligned frame and the horizontal dimensions are

$$\mathbf{c}_{\text{hor}} = \mathbf{R}_y(\alpha) \begin{bmatrix} (a_x + b_x)/2 \\ 0 \\ (a_z + b_z)/2 \end{bmatrix}, \qquad w = b_x - a_x, \quad l = b_z - a_z. \tag{9}$$

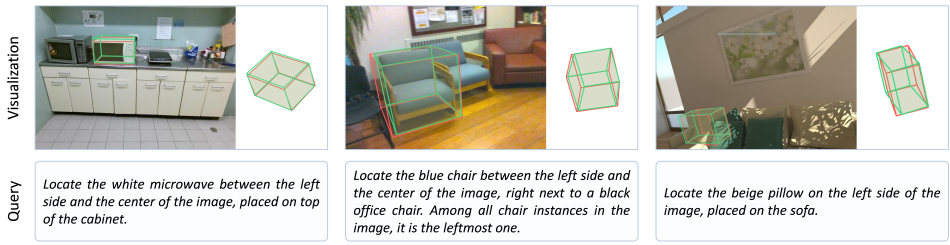

Figure 4: Instance-level qualitative results of our method. Ground-truth and predicted 3D boxes are shown in red and green, respectively.

Vertical Measurement. Rotation about Y preserves vertical coordinates. We therefore measure the object along the vertical Y axis and compute its height and vertical center directly from the lower and upper vertical coordinates:

$$c_{\text{vert}} = \frac{a_y + b_y}{2}, \qquad h = b_y - a_y. \tag{10}$$

Combining the horizontal and vertical estimates gives

$$\mathbf{c}_g = \mathbf{c}_{\text{hor}} + (0, c_{\text{vert}}, 0)^\top, \qquad \mathbf{d} = (w, h, l)^\top. \tag{11}$$

Here, $\mathbf{c}_g$ is the box center in the gravity-aligned frame, and $\mathbf{d}$ contains its width, height, and length. The center uses bound midpoints rather than the point centroid.

Camera-Frame Transformation. We map the fitted box back to the camera coordinate frame using the inverse alignment rotation:

$$\mathbf{c}_{\text{cam}} = \mathbf{R}_g^\top \mathbf{c}_g, \qquad \mathbf{R}_{\text{cam}} = \mathbf{R}_g^\top \mathbf{R}_y(\alpha), \tag{12}$$

Here, $\mathbf{R}_{\text{cam}}$ maps box-local coordinates to camera coordinates, while the dimensions $\mathbf{d}$ remain unchanged.

4.5 Camera-frame 9-DoF Box Recovery

We extract roll ϕ, pitch θ, and yaw ψ using the right-handed convention

$$\mathbf{R}_{\text{cam}} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi), \tag{13}$$

With $\mathbf{c}_{\text{cam}} = (c_x, c_y, c_z)^\top$, we output $\hat{B} = (c_x, c_y, c_z, w, h, l, \phi, \theta, \psi)$. Because $\mathbf{R}_{\text{cam}}$ composes the gravity-aligned heading α with the inverse alignment rotation, α is not any single camera-frame Euler angle; the resulting 9-DoF orientation remains constrained by the estimated scene vertical. The same parameters consistently define every corner as $\mathbf{v}_{\text{cam}} = \mathbf{R}_{\text{cam}}\mathbf{v} + \mathbf{c}_{\text{cam}}$ for $\mathbf{v} \in \{\pm w/2\} \times \{\pm h/2\} \times \{\pm l/2\}$.

5 Experiments

5.1 Experimental Setup

We evaluate both Ref3DOL tracks with identical inputs and camera-frame box outputs. Following Section 3, we report mAcc at 3D IoU thresholds 0.15, 0.25, and 0.50.

We compare against VLMs that directly predict a complete 3D bounding box, including Qwen3-VL-8B-Instruct, Qwen3-VL-8B-Thinking, and Qwen3-VL-32B-Instruct (Bai et al., 2025a), Qwen3.8-27B (Qwen Team, 2026), and GPT-5.6 Sol (OpenAI, 2026). To assess the adaptability of our framework, we use Qwen2.5-VL-3B-Instruct or Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), InternVL3.5-8B (Wang et al., 2025b), or GLM-4.6V-Flash (9B parameters) (Z.ai, 2025) as the 2D semantic grounding backbone. The remaining components use SAM 3 (Carion et al., 2026) for instance segmentation; InfiniDepth (Yu et al., 2026) to predict fine-grained depth, whose metric scale and corresponding camera intrinsics are obtained with MoGe-2 (Wang et al., 2025a); and GeoCalib (Veicht et al., 2024) for gravity estimation. The horizontal heading search uses a default angular step of 1°. All models are evaluated on Ref3DOL without further training; further implementation details are provided in the accompanying code.

5.2 Main Results

Table 2: Category- and instance-level localization accuracy (%). The best direct-prediction result in each column and track is underlined, and the best overall result is bold. For our method, the model in parentheses denotes the 2D semantic grounding backbone.

	SUN RGB-D			Hypersim		
Model	@0.15	@0.25	@0.50	@0.15	@0.25	@0.50
Category-level						
Qwen3-VL-8B-Instruct	<u>41.90</u>	28.13	7.85	22.68	15.60	3.62
Qwen3-VL-8B-Thinking	38.52	25.12	5.92	23.68	16.07	2.94
Qwen3-VL-32B-Instruct	41.75	<u>28.20</u>	<u>8.17</u>	<u>24.27</u>	<u>17.53</u>	<u>4.79</u>
Qwen3.8-27B	36.66	23.44	4.95	20.07	12.19	2.30
GPT-5.6 Sol	21.43	12.64	1.68	7.99	4.57	0.66
Ours (Qwen2.5-VL-7B-Instruct)	46.22	31.71	7.94	37.09	26.36	9.37
Ours (Qwen2.5-VL-3B-Instruct)	45.74	31.73	8.22	37.64	26.31	9.24
Ours (InternVL3.5-8B)	45.44	30.64	7.00	33.70	23.67	7.84
Ours (GLM-4.6V-Flash)	**47.40**	**33.62**	**8.38**	**38.22**	**27.27**	**9.53**
Instance-level						
Qwen3-VL-8B-Instruct	<u>43.08</u>	<u>29.01</u>	<u>7.54</u>	23.21	14.64	3.33
Qwen3-VL-8B-Thinking	38.47	27.09	5.65	22.40	15.51	2.67
Qwen3-VL-32B-Instruct	40.85	28.30	7.40	<u>23.41</u>	<u>17.41</u>	<u>4.80</u>
Qwen3.8-27B	38.18	24.49	5.02	20.47	12.97	2.77
GPT-5.6 Sol	23.91	14.60	1.53	7.33	4.21	0.60
Ours (Qwen2.5-VL-7B-Instruct)	46.52	31.75	7.65	36.13	24.89	8.40
Ours (Qwen2.5-VL-3B-Instruct)	41.23	27.51	6.61	34.83	24.32	7.86
Ours (InternVL3.5-8B)	46.89	31.67	6.88	34.90	24.06	7.49
Ours (GLM-4.6V-Flash)	**49.17**	**34.16**	**8.16**	**38.14**	**26.74**	**9.08**

Category-level localization. As shown in Table 2, at $\mathrm{mAcc}\,@0.25$, our GLM-4.6V-Flash variant achieves 33.62 on SUN RGB-D and 27.27 on Hypersim, improving over the strongest direct-prediction baselines by 5.42 and 9.74 points, respectively. Even the Qwen2.5-VL-3B-Instruct variant outperforms every direct-prediction baseline across all thresholds. This suggests that the improvement cannot be attributed solely to larger VLM capacity and supports the benefit of the decoupled design.

Instance-level localization. As shown in Table 2, at $\mathrm{mAcc}\,@0.25$, our GLM-4.6V-Flash variant achieves 34.16 on SUN RGB-D and 26.74 on Hypersim, exceeding the strongest direct-prediction baselines by 5.15 and 9.33 points, respectively. The consistent gains on the instance-level track support the benefit of resolving target identity before independently recovering metric geometry. Qualitative examples are shown in Figure 4.

Across grounding backbones, GLM-4.6V-Flash performs best overall, while the larger 3B–7B gap on the instance-level track suggests that fine-grained grounding benefits more from VLM capacity than category recognition.

Results using ground-truth camera intrinsics remain close to those obtained with estimated intrinsics and are reported in Appendix B.1.

5.3 Inference Efficiency

Table 3: Amortized inference time per SUN RGB-D query (seconds; lower is better).

Model	Category-level ↓	Instance-level ↓
Qwen3-VL-8B-Instruct	3.71	2.60
Qwen3-VL-32B-Instruct	4.52	3.86
Ours (Qwen2.5-VL-7B-Instruct)	1.63	**1.70**
Ours (GLM-4.6V-Flash)	**1.52**	2.01

Table 3 reports end-to-end amortized time on SUN RGB-D with one NVIDIA H200, eight CPU cores, and batch size one. Our variants outperform direct baselines at 1.52–2.01 seconds per query.

VLM grounding accounts for 70.8–80.3% of runtime; Appendix B.2 provides the stage-wise breakdown.

5.4 Effect of Target Distance

We divide instance-level queries into near, middle, and far bins using dataset-specific 33rd and 67th percentiles of ground-truth target-center depth: 2.11/2.93 meters for SUN RGB-D and 4.73/7.28 meters for Hypersim. We compute $\mathrm{mAcc}\,@0.25$ within each dataset and bin and average the two dataset-level results. Full three-threshold results are provided in Appendix B.3.

Table 4: Instance-level $\mathrm{mAcc}\,@0.25$ by target distance (%). Best direct-prediction results are underlined; best overall results are bold.

Model	Near ↑	Mid ↑	Far ↑
Qwen3-VL-8B-Instruct	29.00	21.72	<u>11.73</u>
Qwen3-VL-32B-Instruct	<u>32.08</u>	<u>24.36</u>	9.38
Ours (Qwen2.5-VL-7B-Instruct)	36.32	28.04	16.32
Ours (GLM-4.6V-Flash)	**38.57**	**30.50**	**16.83**

As shown in Table 4, all methods degrade monotonically with distance. Our GLM-4.6V-Flash variant achieves 38.57, 30.50, and 16.83 in the three bins, exceeding the strongest direct-prediction results by 6.49, 6.14, and 5.10 points, respectively. The limited far-bin accuracy indicates that distant targets remain a major bottleneck.

5.5 Ablation Studies

Table 5: Compact ablation results at $\mathrm{mAcc}\,@0.25$ (%). Cat. and Inst. denote the category- and instance-level tracks. All variants use GLM-4.6V-Flash. Full three-threshold results are provided in Appendix B.4.

	SUN RGB-D		Hypersim	
Variant	Cat.	Inst.	Cat.	Inst.
Full model	**33.62**	**34.16**	**27.27**	**26.74**
SAM 3 query grounding (no VLM 2D grounding)	31.63	19.43	26.96	15.04
2D box as mask (no instance segmentation)	20.73	22.02	11.08	11.83
Without DBSCAN filtering	31.28	31.66	23.44	23.13
Without pose recovery	27.05	27.59	21.79	21.14

Replacing VLM 2D grounding with SAM 3 query grounding reduces category-level $\mathrm{mAcc}\,@0.25$ by only 1.99 and 0.31 points on SUN RGB-D and Hypersim, but instance-level accuracy by 14.73 and 11.70 points, showing that SAM 3 handles category names while the VLM resolves appearance and spatial cues. Using the full predicted 2D box as the mask causes drops of up to 16.19 points because background and neighboring-object points contaminate the target cloud and distort its center, dimensions, and orientation, confirming the need for precise instance masks. Removing DBSCAN costs up to 3.83 points, whereas removing pose recovery costs 5.48–6.57 points across datasets and tracks; both are therefore important for accurate 3D box recovery. Together, these ablations validate each stage of the reconstruction pipeline.

Additional oracle results using ground-truth 2D boxes and metric depth are provided in Appendix B.5.

6 Conclusion

We studied referential monocular 3D object localization, which aims to recover a language-referred object as a metric 9-DoF box from a single RGB image. Our training-free framework decouples semantic grounding from metric geometric reconstruction, combining VLM-based target identification with pretrained geometry modules and gravity-constrained box fitting. We also introduced Ref3DOL, a benchmark built from SUN RGB-D and Hypersim with category- and instance-level evaluation tracks. The proposed decomposition generally yields stronger performance, with the best configuration achieving the strongest overall results, while ablations verify the effectiveness of the key components.

AI USE STATEMENT

We used generative AI models as research components in dataset construction, the proposed framework, and experimental evaluation. During the construction of Ref3DOL, Qwen3-VL-32B-Instruct was used to extract the dominant color of each target and visually salient reference objects. The color attributes were semantically calibrated and used for progressive disambiguation, whereas the reference objects were retained as supplementary semantic context. In our framework, Qwen2.5-VL-3B-Instruct, Qwen2.5-VL-7B-Instruct, InternVL3.5-8B, and GLM-4.6V-Flash were used as semantic grounding backbones to identify referred targets in the image plane, while metric 3D reconstruction was performed by the geometric branch. The experimental direct 3D-box prediction baselines were Qwen3-VL-8B-Instruct, Qwen3-VL-8B-Thinking, Qwen3-VL-32B-Instruct, Qwen3.8-27B, and GPT-5.6 Sol. Generative AI tools were also used for limited language polishing and proofreading. The authors designed the methodology and evaluation, verified the implementation and experimental results, reviewed all AI-assisted content, and take full responsibility for the final text, claims, and artifacts.

REFERENCES

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In *European conference on computer vision*, pp. 422–440. Springer, 2020.

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*, 2025a.

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

Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth Pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, volume 2025, pp. 75602–75637, 2025.

Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A large benchmark and model for 3D object detection in the wild. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13154–13164. IEEE, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

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.

Jang Hyun Cho, Boris Ivanovic, Yulong Cao, Edward Schmerling, Yue Wang, Xinshuo Weng, Boyi Li, Yurong You, Philipp Krähenbühl, Yan Wang, et al. Language-image models with 3D understanding. In *International Conference on Learning Representations*, volume 2025, pp. 36643–36674, 2025.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pp. 226–231, 1996.

Girish Chandar Ganesan, Yuliang Guo, Liu Ren, and Xiaoming Liu. UniDAC: universal metric depth estimation for any camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26953–26963, 2026.

Keyu Guo, Yongle Huang, Shijie Sun, Xiangyu Song, Mingtao Feng, Zedong Liu, Huansheng Song, Tiantian Wang, Jianxin Li, Naveed Akhtar, et al. Beyond human perception: Understanding multi-object world from monocular view. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3751–3760. IEEE, 2025.

Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3D v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (12):10579–10596, 2024.

Linyi Jin, Jianming Zhang, Yannick Hold-Geoffroy, Oliver Wang, Kevin Blackburn-Matzen, Matthew Sticha, and David F Fouhey. Perspective Fields for single image camera calibration. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17307–17316. IEEE, 2023.

Yuzhen Li, Min Liu, Zhaoyang Li, Yuan Bian, Xueping Wang, Erbo Zhai, and Yaonan Wang. Mono3DVG-EnSD: Enhanced spatial-aware and dimension-decoupled text encoding for monocular 3D visual grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6726–6734, 2026.

Yunze Man, Shihao Wang, Guowen Zhang, Johan Bjorck, Zhiqi Li, Liang-Yan Gui, Jim Fan, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. LocateAnything3D: Vision-language 3D detection with Chain-of-Sight. *arXiv preprint arXiv:2511.20648*, 2025.

OpenAI. GPT-5.6 system card, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10106–10116. IEEE, 2024.

Huaizhi Qu, Hossein Nourkhiz Mahjoub, Vaishnav Tadiparthi, Kwonjoon Lee, and Tianlong Chen. MonoVLM: Monocular 3D visual grounding with vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30986–30996, 2026.

Qwen Team. Qwen3.8-27B. Hugging Face, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10892–10902. IEEE, 2021.

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 567–576, 2015.

Javier Tirado-Garín and Javier Civera. AnyCalib: On-manifold learning for model-agnostic single-view camera calibration. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8044–8055. IEEE, 2025.

Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Learning single-image calibration with geometric optimization. In *European Conference on Computer Vision*, pp. 1–20. Springer, 2024.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. In *Advances in Neural Information Processing Systems*, volume 38, pp. 35928–35959, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. *Advances in neural information processing systems*, 37:21875–21911, 2024.

Hao Yu, Haotong Lin, Jiawei Wang, Jiaxin Li, Yida Wang, Xueyang Zhang, Yue Wang, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. InfiniDepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26920–26930, June 2026.

Z.ai. GLM-4.6V-Flash model card. Hugging Face, 2025. URL https://huggingface.co/zai-org/GLM-4.6V-Flash.

Yang Zhan, Yuan Yuan, and Zhitong Xiong. Mono3DVG: 3D visual grounding in monocular images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 6988–6996, 2024.

Renrui Zhang, Han Qiu, Tai Wang, Ziyu Guo, Ziteng Cui, Yu Qiao, Hongsheng Li, and Peng Gao. MonoDETR: Depth-guided transformer for monocular 3D object detection. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9121–9132. IEEE, 2023.

Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3D object detection. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3288–3297. IEEE, 2021.

A Instance-Level Query Construction Details

We use the same automatic construction pipeline for SUN RGB-D and Hypersim. After visibility filtering, each target is compared with all valid instances of the same category in the image and is progressively disambiguated using color, coarse absolute position, and reliable relative ordering. The 2D and 3D annotations described below are used only for offline query construction and are not provided as additional inputs during localization. Each target produces at most one final query, together with its target identifier, retained attributes, and successful disambiguation stage.

Visual attribute extraction and color calibration. For each target, we highlight its 2D box in red and prompt Qwen3-VL-32B-Instruct to extract a basic dominant color and an optional relation to a salient contextual anchor. The color is marked invalid when it is mixed, complex, or visually uncertain; an anchor relation is omitted when no large and clearly recognizable reference object is present. The prompt disallows inferred hidden properties and uncertain counting or ordering descriptions.

We lowercase and trim the color strings and map semantically equivalent variants to strict color groups. For example, brightness modifiers do not make *brown*, *dark brown*, and *light brown* reliable instance-level distinctions. Let $\mathcal{G}_i$ denote the set of calibrated color groups for instance i. Two instances i and j are color-distinguishable only when both colors are valid, both sets are nonempty, and

$$\mathcal{G}_i \cap \mathcal{G}_j = \emptyset. \tag{14}$$

We define the color-conflict indicator $C_{ij} = 1$ otherwise. Color alone resolves a target only if $C_{ij} = 0$ for every same-category distractor j. Reference-object relations are retained as supplementary context but do not determine uniqueness.

Coarse absolute-position disambiguation. For each instance, we select the first valid 2D box in the order `bbox2D_tight`, `bbox2D_trunc`, and `bbox2D_proj`. A box is valid if it has positive width and height and intersects the image. Given image width W and horizontal box boundaries $u_{\min}$ and $u_{\max}$, its normalized horizontal center is

$$r_i = \mathrm{clip}\left(\frac{u_{\min,i} + u_{\max,i}}{2W},\, 0,\, 1\right). \tag{15}$$

We assign r_i to one of the five regions in Table 6.

Table 6: Horizontal regions used for absolute-position descriptions.

Index	Description	Normalized interval
0	Left	$[0, 0.30)$
1	Between left and center	$[0.30, 0.40)$
2	Center	$[0.40, 0.60)$
3	Between center and right	$[0.60, 0.70)$
4	Right	$[0.70, 1.00]$

Let $z_i \in \{0, \ldots, 4\}$ be the region index. Instances are reliably separated by absolute position only if $|z_i - z_j| \geq 2$; instances in the same or adjacent regions remain position-conflicting. Let $P_{ij} = 1$ denote this position conflict, including cases in which a valid position cannot be obtained. Color and position are combined pairwise as

$$J_{ij} = C_{ij} \wedge P_{ij}. \tag{16}$$

The target is accepted at this stage only when $J_{ij} = 0$ for every same-category distractor. Thus, different distractors may be excluded by different cues, allowing color and position to disambiguate a target jointly even when neither cue suffices alone.

Relative-position ordering. If color and absolute position remain insufficient, we use annotated 3D centers to construct an ordering cue. Ordering is computed over all valid same-category instances in the image, not only the candidates left ambiguous by earlier stages, and is restricted to groups of two to four instances. We use `center_cam` when available and otherwise average the eight annotated 3D-box corners.

For an axis $a \in \{x, y, z\}$, let p_i^a be the camera-frame center coordinate of instance i. The three axes correspond to left-to-right, top-to-bottom, and near-to-far ordering, respectively. We define the span L_a, minimum adjacent spacing Δ_a, and axis dominance D_a as

$$L_a = \max_i p_i^a - \min_i p_i^a, \qquad \Delta_a = \min_{1 \le k < n} \left(p_{(k+1)}^a - p_{(k)}^a \right), \tag{17}$$

$$D_a = \frac{L_a}{\max(\max_{b \neq a} L_b, \epsilon)}, \qquad \epsilon = 10^{-6}, \tag{18}$$

where $p_{(k)}^a$ is the k-th coordinate after sorting along axis a. Table 7 gives the acceptance conditions shared by both datasets. The spacing thresholds combine a fixed metric separation with a fraction of the group span; uniform spacing is not required.

Table 7: Acceptance conditions for relative-position ordering. All distances are in meters.

Ordering cue	Span	Minimum adjacent spacing T_a	Dominance
Left–right (x)	$L_x \ge 0.25$	$\Delta_x \ge \max(0.20, 0.15 L_x)$	None
Top–bottom (y)	$L_y \ge 0.45$	$\Delta_y \ge \max(0.35, 0.20 L_y)$	$D_y \ge 1.50$
Near–far (z)	$L_z > 10^{-6}$	$\Delta_z \ge \max(1.40, 0.35 L_z)$	$D_z \ge 1.80$

For left–right ordering, groups of three or four instances additionally undergo a single-row layout check. The group passes directly when $L_z \le 1.00$; otherwise, it must satisfy

$$|\mathrm{corr}(x, z)| \ge 0.65, \tag{19}$$

where corr is the Pearson correlation coefficient. If either coordinate has standard deviation below 10^{-8} when the correlation must be evaluated, the check fails. Two-instance groups skip this layout check but must still satisfy the span and spacing conditions.

For every accepted axis, we compute

$$S_a = D_a + \frac{\Delta_a}{T_a}, \tag{20}$$

where T_a is the corresponding spacing threshold in Table 7. If the depth axis is accepted and $L_z \ge 1.50$, near–far ordering is selected. Otherwise, left–right and top–bottom are considered first: when both are accepted, top–bottom is selected only if $S_y \ge 1.25 S_x$, and left–right is selected otherwise; when only one is accepted, that axis is used. If neither is accepted but the depth axis is, near–far ordering is used. Groups with no accepted axis are excluded from automatic relative-position disambiguation.

After selecting an axis, instances are ranked in ascending coordinate order and the target rank is converted into a phrase such as "the third from left to right among all chairs in the image." Relative-ordering clauses are added only when earlier attributes do not already resolve the target. The overall retention order is category uniqueness, color, color combined with absolute position, and finally reliable relative ordering; targets unresolved after all stages are excluded from the instance-level query set.

B Additional Experimental Results

B.1 Results with Ground-Truth Camera Intrinsics

Table 8: Category- and instance-level localization accuracy using ground-truth camera intrinsics (%). Only variants of our method are shown; all other components follow the main experimental setting. The best result in each column and track is bold.

	SUN RGB-D			Hypersim		
Model	@0.15	@0.25	@0.50	@0.15	@0.25	@0.50
Category-level						
Ours (Qwen2.5-VL-7B-Instruct)	45.54	33.06	8.63	37.12	26.20	9.71
Ours (Qwen2.5-VL-3B-Instruct)	45.67	32.84	**9.07**	37.57	26.08	9.65
Ours (InternVL3.5-8B)	44.14	30.91	7.74	33.78	23.28	8.04
Ours (GLM-4.6V-Flash)	**47.67**	**34.91**	9.03	**38.62**	**26.79**	**9.94**
Instance-level						
Ours (Qwen2.5-VL-7B-Instruct)	46.16	32.64	8.32	35.80	24.63	8.69
Ours (Qwen2.5-VL-3B-Instruct)	41.15	28.68	7.41	34.69	24.27	8.00
Ours (InternVL3.5-8B)	46.65	32.04	7.64	34.79	23.81	7.69
Ours (GLM-4.6V-Flash)	**49.84**	**35.25**	**8.88**	**38.29**	**26.29**	**9.36**

Compared with the estimated-intrinsics results in Table 2, using ground-truth intrinsics changes individual accuracy values by at most 1.35 percentage points across datasets, tracks, thresholds, and grounding backbones. The differences are therefore small and do not alter the overall conclusions.

B.2 Stage-wise Inference Time Breakdown

Table 9 decomposes the inference cost of our two grounding backbones on the SUN RGB-D category- and instance-level tracks.

Table 9: Per-stage amortized inference time per query (seconds) on the SUN RGB-D dataset, with percentages of total time shown in parentheses.

	Category-level		Instance-level	
Stage	Qwen2.5-VL-7B-Instruct	GLM-4.6V-Flash	Qwen2.5-VL-7B-Instruct	GLM-4.6V-Flash
VLM 2D grounding	1.178 (72.3%)	1.078 (70.8%)	1.297 (76.3%)	1.610 (80.3%)
Gravity estimation	0.037 (2.3%)	0.037 (2.4%)	0.030 (1.7%)	0.030 (1.5%)
Depth reconstruction	0.151 (9.3%)	0.151 (9.9%)	0.121 (7.1%)	0.121 (6.0%)
SAM 3 segmentation	0.122 (7.5%)	0.119 (7.8%)	0.120 (7.1%)	0.124 (6.2%)
3D box fitting	0.142 (8.7%)	0.138 (9.0%)	0.131 (7.7%)	0.120 (6.0%)
Total	**1.630 (100%)**	**1.523 (100%)**	**1.699 (100%)**	**2.005 (100%)**

Measurements follow the setting of Table 3.

B.3 Full Distance-Stratified Results

Table 10 reports the complete three-threshold results corresponding to the distance analysis in Table 4. The distance bins and averaging procedure follow Section 5.4.

Table 10: Full instance-level distance-stratified localization accuracy (%). The best direct-prediction result in each column is underlined, and the best overall result is bold.

	Near			Mid			Far		
Model	@0.15	@0.25	@0.50	@0.15	@0.25	@0.50	@0.15	@0.25	@0.50
Qwen3-VL-8B-Instruct	42.43	29.00	7.29	32.34	21.72	5.60	20.36	11.73	2.76
Qwen3-VL-8B-Thinking	40.08	28.85	6.55	30.51	21.84	4.25	15.80	10.19	1.63
Qwen3-VL-32B-Instruct	43.69	32.08	8.69	32.32	24.36	6.43	14.50	9.38	2.21
Qwen3.8-27B	35.95	24.25	5.78	31.16	19.26	4.14	16.83	10.31	1.91
GPT-5.6 Sol	20.15	11.73	1.39	14.31	8.47	0.98	8.45	5.04	0.50
Ours (Qwen2.5-VL-7B-Instruct)	50.43	36.32	11.88	39.89	28.04	7.84	**26.61**	16.32	**2.87**
Ours (Qwen2.5-VL-3B-Instruct)	46.65	33.81	10.64	36.48	25.58	6.95	23.50	14.12	2.20
Ours (InternVL3.5-8B)	50.05	35.76	10.71	38.87	27.25	6.58	24.19	14.97	2.39
Ours (GLM-4.6V-Flash)	**53.15**	**38.57**	**12.44**	**42.83**	**30.50**	**8.70**	26.39	**16.83**	2.84

B.4 Full Ablation Results

Table 11 reports the complete three-threshold results corresponding to the compact ablation in Table 5.

Table 11: Full ablation results (%). All variants use GLM-4.6V-Flash as the 2D semantic grounding backbone.

Dataset	Task	Variant	@0.15	@0.25	@0.50
SUN RGB-D	Category	Full model	47.40	33.62	8.38
		SAM 3 query grounding	43.92	31.63	7.64
		2D box as mask	39.17	20.73	1.97
		Without DBSCAN	45.87	31.28	7.12
		Without pose recovery	41.27	27.05	3.57
	Instance	Full model	49.17	34.16	8.16
		SAM 3 query grounding	30.17	19.43	4.33
		2D box as mask	41.96	22.02	1.99
		Without DBSCAN	47.50	31.66	6.97
		Without pose recovery	43.32	27.59	3.60
Hypersim	Category	Full model	38.22	27.27	9.53
		SAM 3 query grounding	37.02	26.96	9.46
		2D box as mask	24.58	11.08	0.78
		Without DBSCAN	33.37	23.44	8.31
		Without pose recovery	32.58	21.79	5.61
	Instance	Full model	38.14	26.74	9.08
		SAM 3 query grounding	21.74	15.04	5.00
		2D box as mask	25.54	11.83	0.81
		Without DBSCAN	33.86	23.13	7.88
		Without pose recovery	32.03	21.14	5.24

B.5 Oracle Analysis

Table 12: Oracle analysis on the SUN RGB-D instance-level track using GLM-4.6V-Flash as the 2D semantic grounding backbone (%). All settings use ground-truth camera intrinsics. The default system uses predicted 2D boxes and predicted metric depth.

Setting	mAcc @0.15	mAcc @0.25	mAcc @0.50
Default	49.84	35.25	8.88
GT 2D box	58.58	43.01	10.96
GT depth	75.73	60.67	19.66
GT 2D box + GT depth	**87.20**	**72.56**	**24.00**

Table 12 isolates the two principal sources of upstream error. Replacing the VLM prediction with a ground-truth 2D box, while still using SAM 3 to produce the mask, improves mAcc @0.25 from 35.25 to 43.01, a gain of 7.76 points. Replacing predicted depth with ground-truth depth yields a larger improvement to 60.67, a gain of 25.42 points, identifying monocular metric depth as the dominant current bottleneck.

Using both ground-truth 2D boxes and depth raises the three metrics to 87.20, 72.56, and 24.00, improvements of 37.36, 37.31, and 15.12 points over the default system. The joint gain indicates that 2D grounding and depth errors accumulate during 3D recovery and have complementary effects. The remaining error is attributable to factors not replaced by these oracles, including instance-mask errors, occlusion, gravity estimation, and box fitting.