TOWARD COMPREHENSIVE 3D GROUNDING: ORIENTATION GROUNDING THROUGH VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Grounding is a core capability of spatial vision-language models, yet most existing work focuses only on where a referred object is. Many 3D tasks also require knowing how it is oriented. Although existing 3D VLMs may predict oriented boxes, box pose does not explicitly capture object-centric orientation or symmetry-induced ambiguities. We introduce *orientation grounding*, a referring grounding task that predicts an object's 6D orientation and axial symmetry from a language or box query in single-view or multi-view scenes. To support this task, we construct **ReferOri**, with 331K multi-view and 387K single-view orientation-grounding queries obtained through scalable reconstruction, consistency checking, and human verification. We further present **OG-VLM**, which adapts a 3D VLM with structured box/orientation outputs, sign and symmetry tokens, and geometry-aware auxiliary losses. Across single-view and multi-view benchmarks, OG-VLM substantially outperforms orientation-aware VLM baselines and surpasses object-level orientation foundation models on scene-level referring benchmarks, showing that explicit orientation grounding is a distinct and learnable capability beyond localization. Downstream results validate its benefit for orientation-related spatial reasoning.

1 INTRODUCTION

Recent progress in spatial vision-language models (VLM), especially 3D VLM, has made grounding a central interface between language and geometry. Most existing grounding benchmarks and models focus on localization: given a referring expression, the model predicts a 2D or 3D box that identifies where the object is (Hong et al., 2023; Chen et al., 2024b; Zhang et al., 2024a; Li et al., 2025; Arnaud et al., 2025). More recent systems further use grounded objects as intermediate representations for spatial reasoning and 3D VQA (Ma et al., 2025b; Chen et al., 2025b). **This trend makes grounding more than a supporting component; it is becoming a prerequisite for reliable spatial understanding.**

Localization alone, however, only specifies part of an object's state. Many spatial and scene-level tasks (Linghu et al., 2024; Ma et al., 2025a) also require object-centric orientation: as shown in Fig 1, model may know there is a chair in front of himself but doesn't understand the true direction he can sit down. This information is not equivalent to the pose of an enclosing box. Although recent 3D VLMs (Zheng et al., 2025; Linghu et al., 2026) use oriented bounding-box supervision from datasets such as KITTI, SUN RGB-D, nuScenes, ARKitScenes, and EmbodiedScan (Geiger et al., 2013; Song et al., 2015; Caesar et al., 2020; Baruch et al., 2021; Wang et al., 2024a), the resulting orientation remains coupled to box geometry, coordinate conventions, and localization quality. In contrast, object-centric orientation asks how the referred object itself is directed, which often depends on part structure, local shape, symmetry and environment.

Despite the practical importance of orientation, dedicated studies on orientation grounding remain limited. Existing spatial VLMs can sometimes infer coarse directions from category priors or canonical poses (Ma et al., 2022; Azuma et al., 2022; Chen et al., 2024b), but such behavior is fragile under non-canonical object poses, partial occlusion, and symmetric or near-symmetric shapes. Studying orientation as an explicit grounding target therefore exposes a different capability from

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.

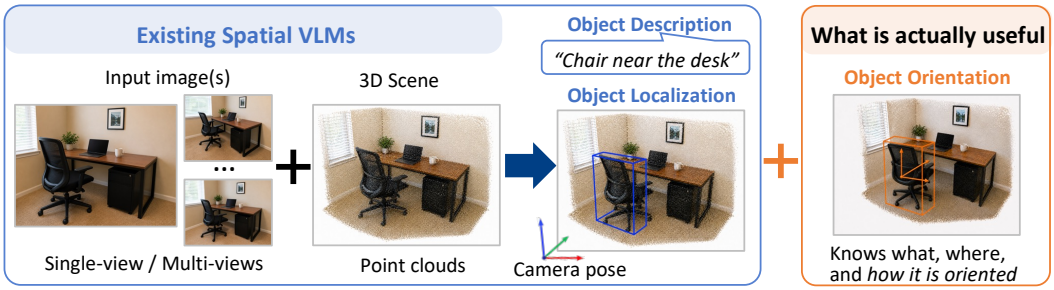


Figure 1: Orientation grounding extends localization grounding from predicting where a referred object is to estimating how it is oriented. ReferOri provides single-view and multi-view annotations with object boxes, 6D orientation, and axial symmetry labels, enabling OG-VLM to ground both object position and object-centric direction.

localization: the model must both identify the referred object and recover its object-centric axes from visual and geometric evidence.

In this work, we define *orientation grounding* as a natural next step beyond localization grounding toward more comprehensive 3D world understanding. Given a queried single-view or multi-view 3D scene and either a referring expression or an object box, the goal is to predict the target object's 6D orientation together with axial symmetry labels along its canonical axes. This formulation raises three challenges: referring orientation labels are scarce, symmetry introduces inherent orientation ambiguities, and autoregressive VLMs are prone to sign flips and inconsistent binary predictions in structured numeric outputs.

To address these challenges, we introduce **ReferOri**, a referring orientation-grounding dataset built by linking existing referring datasets with scalable scene and object reconstruction. ReferOri combines multi-view consistency checks with human verification to produce object orientation and axial symmetry annotations, yielding 331K multi-view and 387K single-view training queries and supporting both scene-level referring grounding and object-centric orientation estimation. Building on ReferOri, we present **Orientation-Grounding VLM** (**OG-VLM**), an adaptation of an existing 3D VLM for orientation grounding. OG-VLM introduces explicit `<bbox>` and `<ori>` output spans, dedicated tokens for numerical signs and symmetry labels, and lightweight geometry-aware heads for box, orientation, and symmetry supervision. Experiments show substantial gains over VLM-based spatial grounding/reasoning models and object-level orientation foundation models, demonstrating that orientation grounding is a distinct and learnable capability beyond localization grounding alone.

In this paper, our contributions can be summarized as follows:

- We introduce *orientation grounding* as a referring 3D grounding task that extends localization grounding to object-centric orientation and axial symmetry, and construct **ReferOri**, a scalable benchmark spanning single- and multi-view scenes through an automated data construction pipeline.
- We present **OG-VLM**, an adaptation of an existing 3D VLM for orientation grounding that combines structured spatial output tokens with geometry-aware losses for box, orientation, and symmetry supervision.
- We conduct extensive experiments across multi-view, single-view, and orientation-specific benchmarks, showing consistent gains over orientation-aware VLM baselines and object-level orientation foundation models, while downstream evaluations further demonstrate the benefit of orientation grounding for orientation-related spatial reasoning.

2 Related Work

2.1 3D Referring Grounding

3D referring grounding links language to scene geometry by localizing queried objects. Prior work develops datasets and models for predicting target boxes from reconstructed scenes or multi-view observations (Chen et al., 2020; Zhang et al., 2023b; Hong et al., 2023; Chen et al., 2024b; Zhang et al., 2024a; Li et al., 2025; Arnaud et al., 2025), while recent spatial VLMs further use grounding as an intermediate representation for reasoning and 3D question answering (Ma et al., 2025b; Chen et al., 2025b; Liu et al., 2025a). These works establish localization as a foundation for spatial understanding,

but mainly predict the position and extent of referred objects. We retain this referring setting while extending the target to object-centric orientation and symmetry.

2.2 Orientation Dataset

Existing orientation datasets mainly fall into three categories. Object-level 3D datasets (Reizenstein et al., 2021; Collins et al., 2022; Wu et al., 2023; Ahmadyan et al., 2021) provide canonical geometry and object-centric poses, but typically lack scene context. Single-view scene-level datasets (Xiang et al., 2016; 2014; Ma et al., 2024) often infer pose through CAD matching, making them dependent on external 3D priors and sensitive to occlusion, truncation, and viewpoint ambiguity. Multi-view scene-level datasets (Geiger et al., 2013; Song et al., 2015; Caesar et al., 2020; Zhang et al., 2024b) provide oriented boxes or pose annotations, but mainly target detection and pose estimation rather than referring orientation grounding. ARKitScenes (Baruch et al., 2021) and EmbodiedScan (Wang et al., 2024a) offer rich scene geometry and can support referring-oriented training after adaptation. Our pipeline extends this line by constructing large-scale orientation and symmetry annotations directly under referring settings for both single-view and multi-view data.

2.3 Orientation Estimation

Non-VLM Related. Earlier approaches (Avetisyan et al., 2019; Ma et al., 2024; Chen et al., 2024a) recovered object orientation through computer-aided design (CAD) matching and alignment in images or 3D scenes, relying on large 3D CAD repositories and remaining sensitive to occlusion, truncation, and structural variation. Later methods moved toward direct pose or orientation estimation from visual input, including PoseCNN (Xiang et al., 2017), the OnePose series (Sun et al., 2022; He et al., 2022), the GenPose series (Zhang et al., 2023a; 2024b), POPE (Fan et al., 2024), and the Orient Anything series (Wang et al., 2024b; 2026). More recent reconstruction-based methods, such as One2Any (Liu et al., 2025b) and SAM3D (Chen et al., 2025a), estimate orientation by reconstructing object geometry. These works mainly formulate orientation as pose estimation, CAD alignment, or geometry recovery, whereas we study it as a grounding target under language and scene context.

VLM-Related. Recent VLM-based works have begun to incorporate orientation into spatial understanding, but it is rarely treated as a first-class grounding target with dedicated representation and supervision. SpatialReasoner (Ma et al., 2025b) uses orientation grounding to support downstream reasoning, yet inaccurate object-centric orientation can become a bottleneck for orientation-dependent reasoning, as reflected in our later evaluations. SOFAR (Qi et al., 2025) uses a separate semantic-orientation predictor for spatial reasoning and object manipulation, but focuses on object-level semantic directions rather than scene-level referring orientation grounding. VG-LLM (Zheng et al., 2025) predicts orientation as part of 9D oriented boxes, coupling rotation with box geometry and Euler-angle parameterization. More generally, query-conditioned object-centric orientation and symmetry-induced ambiguities remain insufficiently modeled under a unified cross-category reference frame in cluttered scenes. We address these issues with explicit referring orientation grounding, dedicated orientation/symmetry supervision, and geometry-aware modeling.

3 Methodology

3.1 Task Formulation

Given a scene observation X and a query q, orientation grounding predicts the referred object's spatial state $\hat{\mathcal{A}} = (\hat{B}, \hat{R}, \hat{S})$, where $\hat{B} \in \mathbb{R}^6$ is the predicted 3D bounding box, $\hat{R} \in SO(3)$ is the predicted object-centric rotation in the scene/world coordinate system, and $\hat{S} \in \{0, 1\}^3$ denotes the predicted axial symmetry along the canonical x, y, and z axes. We use $\mathcal{A} = (B, R, S)$ to denote the corresponding ground-truth annotation throughout the paper. The query can be a natural-language referring expression or a box query. The model predicts rotation using the continuous 6D representation of R (Zhou et al., 2019), i.e., the first two axes of the rotation basis, which is converted to $\hat{R} \in SO(3)$. Symmetry matters because several orientations can be perceptually equivalent for symmetric objects. We therefore define the symmetry-aware discrepancy

$$d_{\hat{S}}(R, \hat{R}) = \min_{Q \in \mathcal{G}(\hat{S})} \arccos\left(\frac{\mathrm{tr}\left(\hat{R}(RQ)^\top\right) - 1}{2}\right),$$

When no symmetry ambiguity is present, $\mathcal{G}(\hat{S}) = \{I\}$. This formulation separates orientation errors from equivalent poses induced by object symmetry. We apply the same symmetry-aware treatment at inference time.

3.2 Data Construction

ReferOri attaches object-centric orientation and symmetry annotations to existing referring data. For **single-view** scenes, we use RefCOCO (Yu et al., 2016). For **multi-view** scenes, we annotate ScanNet (Dai et al., 2017) objects and transfer the resulting annotations to ScanNet-based referring datasets for training and evaluation. Following the notation in Sec. 3.1, each referred object is annotated as $\mathcal{A} = (B, R, S)$, where B is the 3D box, $R \in SO(3)$ is the world-frame object orientation, and $S \in \{0, 1\}^3$ denotes axial symmetry. Algorithm 1 summarizes the overall annotation procedure, with the single-view and multi-view branches detailed below.

Algorithm 1 Orientation annotation pipeline

Require: Referred sample (I, b) or multi-view sample $\{(I_t, b_t)\}_{t=1}^{T}$
Ensure: Orientation grounding annotation $\mathcal{A} = (B, R, S)$

if single-view **then**
  Estimate $T_{\text{world}\leftarrow\text{cam}}$ with Map Anything
  Obtain object mask M from b using SAM2
  Reconstruct object point cloud $\mathcal{P}$ and pose $T_{\text{cam}\leftarrow\text{obj}}$ with SAM3D
  Estimate symmetry labels S from $\mathcal{P}$
  Compute world-frame orientation R
  Construct 3D box B
else
  Select the top-2 most informative views
  Estimate two candidate world-frame orientations $\{R_1, R_2\}$ using the same SAM2–SAM3D pipeline
  if $\angle(R_1, R_2) < \tau$ **then**
    Set $R \leftarrow R_1$
  else
    Apply lightweight human correction on a coarse annotation to obtain R
  end if
  Estimate S and construct B
end if
return $\mathcal{A} = (B, R, S)$

Single-view construction. Given an image I and a referred bounding box b, we first use Map Anything (Keetha et al., 2025) to estimate scene depth and camera parameters, yielding a camera-to-world transformation $T_{\text{world}\leftarrow\text{cam}}$. We then apply SAM2 (Ravi et al., 2024) to refine the target mask M, and feed (I, M) into SAM3D (Chen et al., 2025a) to reconstruct an object point cloud $\mathcal{P}$ together with an object-to-camera transformation $T_{\text{cam}\leftarrow\text{obj}}$. Based on $\mathcal{P}$, we estimate the axial symmetry labels S, and treat the canonical coordinate system of the reconstructed object as an orientation proxy. The final world-frame orientation is computed as $R_{\text{obj}}^{\text{world}} = R_{\text{world}\leftarrow\text{cam}}\, R_{\text{cam}\leftarrow\text{obj}}$.

Multi-view construction. For ScanNet scenes, we select the two most informative views of the referred object, approximated by the largest visible mask area. We estimate object orientation independently from both views using the same SAM2–SAM3D pipeline and transform the results into the shared world frame using the native multi-view camera parameters. If the angular discrepancy between the two estimates is below a threshold τ, we directly accept the annotation; otherwise, we retain one estimate as a coarse annotation and apply lightweight human correction.

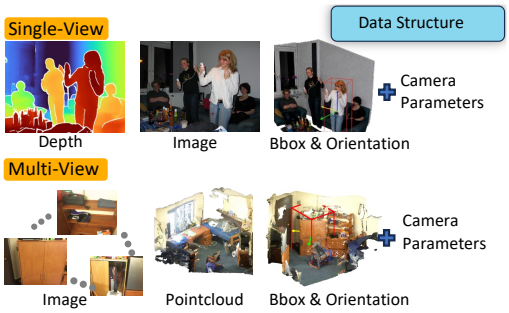


Figure 2: Overview of ReferOri dataset structure.

Verification. We validate annotation quality through human inspection. For single-view data, five rounds of 100 randomly sampled annotations are reviewed for referent correctness, orientation consistency with the source image and reconstructed geometry, and coordinate consistency, yielding a 96% human-verified correctness rate. For multi-view ScanNet data, all ∼10K object annotations are manually inspected and corrected when necessary. We additionally audit 800 images with two independent evaluators, obtaining 97.5% inter-annotator agreement. Detailed verification protocols and annotation-module validation are provided in Appendix B.6. The resulting structure of ReferOri after annotation and verification is illustrated in Fig. 2.

3.3 OG-VLM Overview

OG-VLM predicts the 3D box, orientation, and axial symmetry of a queried object (Fig. 3) from single-/multi-view input or a point cloud, together with a language or box query. Built on GS-

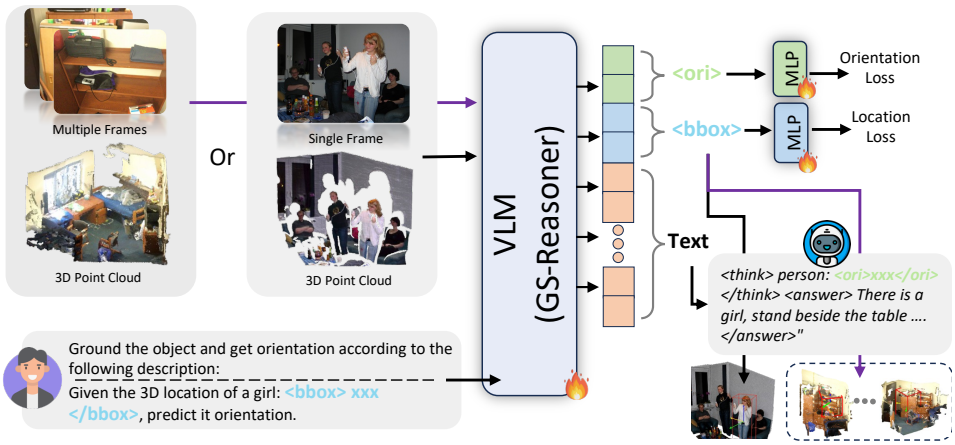


Figure 3: Overview of OG-VLM. Given multi-view frames, a single frame, or a 3D point cloud together with a text or box query, the VLM backbone generates structured grounding outputs with separate <bbox> and <ori> spans. Lightweight heads attached to the corresponding tokens provide localization, orientation, and symmetry supervision through geometry-aware losses.

Reasoner (Chen et al., 2025b), it introduces two extensions: separate <bbox> and <ori> spans with dedicated sign and symmetry tokens (Sec. 3.3.1), and lightweight heads on span-closing hidden states for geometry-aware box, orientation, and symmetry supervision (Sec. 3.3.2).

3.3.1 Special Token Design for Orientation Grounding

GS-Reasoner represents 3D boxes as plain-text coordinates inside a <bbox> span. Extending this format to orientation introduces three issues: interference between box and orientation values, severe errors from sign flips in rotation components, and inconsistent free-form symmetry labels. Following prior LLM-based grounding models (Peng et al., 2023; Chen et al., 2024b), we address these issues with separate structured spans and dedicated sign and symmetry tokens.

To improve the stability of structured spatial prediction, we explicitly disentangle localization and orientation outputs using $Y = [y^{\text{bbox}}, y^{\text{ori}}]$, where $y^{\text{bbox}} =$ <bbox>($\tilde{\mathbf{b}}$)</bbox>, and $y^{\text{ori}} =$ <ori>($\tilde{\mathbf{o}}, \mathbf{s}$)</ori>. Here, $\mathbf{b} \in \mathbb{R}^6$ denotes the 3D bounding box, $R \in SO(3)$ denotes the object orientation, $\mathbf{o} \in \mathbb{R}^6$ is the continuous 6D representation of R used in training, and $\mathbf{s} \in \{0,1\}^3$ denotes the axial symmetry labels.

Rather than generating signed values as free-form text, we factorize each scalar v of $\mathbf{b}$ and $\mathbf{o}$ into a sign token and its magnitude, and we write each symmetry label as a dedicated token:

$$v \mapsto \big(\sigma(v), |v|\big), \quad \sigma(v) = \begin{cases} \texttt{<+>}, & v \geq 0, \\ \texttt{<->}, & v < 0, \end{cases} \qquad s_a \mapsto \begin{cases} \texttt{<s_yes>}, & s_a = 1, \\ \texttt{<s_no>}, & s_a = 0. \end{cases} \tag{1}$$

This design reduces interference between box and orientation prediction while better matching token-based language modeling. Explicit sign tokens separate coarse spatial direction from numeric magnitude, reducing severe errors caused by sign flips. Discrete symmetry tokens likewise avoid unstable free-form binary generation and improve output consistency.

3.3.2 Geometry-Aware Loss for Grounding

The language-modeling loss treats each numeric token as an independent class and therefore does not explicitly reflect geometric distance. For example, predictions with substantially different coordinate errors may receive similar token-level penalties when they differ from the target by a similar number of tokens. Moreover, the language-modeling objective alone does not enforce geometric constraints, such as orthogonality between the predicted orientation axes. To provide explicit geometric supervision, we attach separate lightweight MLP heads to the hidden states of the corresponding tokens for box, orientation, and symmetry prediction. These heads provide task-specific supervision for each spatial component (*i.e.,* $\mathcal{L}_{\text{box}}$, $\mathcal{L}_{\text{ori}}$, $\mathcal{L}_{\text{sym}}$) and augment the language-modeling objective:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{ori}}\mathcal{L}_{\text{ori}} + \lambda_{\text{sym}}\mathcal{L}_{\text{sym}}. \tag{2}$$

Box loss. From the hidden state at the </bbox> token, we regress a 3D box $B = [x_{\min}, y_{\min}, z_{\min}, x_{\max}, y_{\max}, z_{\max}] \in \mathbb{R}^6$, and define

$$\mathcal{L}_{\text{box}} = \alpha_{\text{giou}}\mathcal{L}_{\text{GIoU}} + \alpha_{\text{scale}}\mathcal{L}_{\text{scale}} + \alpha_{\text{center}}\mathcal{L}_{\text{center}}, \tag{3}$$

where $\mathcal{L}_{\text{GIoU}}$, $\mathcal{L}_{\text{scale}}$, and $\mathcal{L}_{\text{center}}$ enforce overlap-aware alignment, relative size consistency, and normalized center consistency, respectively.

Orientation loss. The ground-truth orientation lies in $SO(3)$, while we predict its continuous 6D representation as two 3D axes $\hat{\mathbf{r}}, \hat{\mathbf{f}}$. Let $\mathbf{r}, \mathbf{f}$ denote the ground-truth right and front axes. We define

$$\mathcal{L}_{\text{ori}} = \beta_{\text{align}}\mathcal{L}_{\text{align}} + \beta_{\text{ortho}}\mathcal{L}_{\text{ortho}}, \tag{4}$$

where

$$\mathcal{L}_{\text{align}} = \frac{1}{2}\left[\left(1 - \cos(\hat{\mathbf{r}}, \mathbf{r}) + \epsilon\right)^{\lambda} + \left(1 - \cos(\hat{\mathbf{f}}, \mathbf{f}) + \epsilon\right)^{\lambda}\right], \qquad \mathcal{L}_{\text{ortho}} = |\cos(\hat{\mathbf{r}}, \hat{\mathbf{f}})|. \tag{5}$$

The two terms encourage axis alignment and mutual orthogonality, respectively. The adaptive exponent λ preserves sensitivity in the small-error regime. At inference, Gram–Schmidt orthogonalization converts the predicted axes into a valid rotation matrix.

Symmetry loss. We further apply a cosine-based symmetry regularizer,

$$\mathcal{L}_{\text{sym}} = 1 - \cos(\hat{\mathbf{s}}, \mathbf{s}), \tag{6}$$

where $\hat{\mathbf{s}}$ and $\mathbf{s}$ denote the predicted and ground-truth symmetry vectors, respectively.

Overall, the proposed loss supplements token-level supervision with explicit geometric constraints, improving box consistency, axis alignment, and symmetry-aware grounding. Detailed formulations of $\mathcal{L}_{\text{GIoU}}$, $\mathcal{L}_{\text{scale}}$, $\mathcal{L}_{\text{center}}$, as well as the adaptive schedule of λ, are provided in Appendix E.

4 Experiments

4.1 Experimental Settings

Training Settings. We use the proposed ReferOri as training data, including *387K single-view* and *331K multi-view* queries. We consider three progressively enhanced settings: **Only data**, which fine-tunes the base model on our constructed orientation-grounding data; **+ Special token**, which introduces our special-token representation; and **+ Geometric loss**, which further incorporates our geometry-aware loss. Further details on dataset construction, hyperparameter selection, and implementation can be found in Appendix E.

Benchmarks and Metrics. We evaluate on both single-view and multi-view orientation-grounding benchmarks. The single-view benchmarks include RefCOCO*, RefCOCO+*, RefCOCOg* (Yu et al., 2016), and Ori-COCO (Wang et al., 2024b), while the multi-view benchmarks include ScanRefer* (Chen et al., 2020), Multi3DRefer* (Zhang et al., 2023b), and ARKitScenes (Baruch et al., 2021). Here, $(*)$ denotes benchmarks whose orientation annotations are generated using our annotation pipeline. In total, five benchmarks are annotated by our pipeline, whereas OriCOCO and ARKitScenes provide native orientation annotations. Since neither dataset relies on our annotation process, we use them exclusively for out-of-distribution (OOD) and zero-shot evaluation.

For continuous-orientation benchmarks, we report **Acc@15** and **Acc@30**, *i.e.,* the percentage of predictions whose orientation error is within 15° and 30°, respectively. For fair comparison, all methods' native orientation representations are converted to rotation matrices in a shared right-handed reference frame and scored using the same symmetry-aware $SO(3)$ geodesic metric. Ori-COCO is evaluated using its native eight-sector classification accuracy because it only provides discrete orientation labels. Per-baseline conversion details are provided in Appendix C.3.

4.2 Main Results

As shown in Table 1, current spatial VLMs remain largely dominated by localization-oriented grounding, although grounding has become increasingly central to spatial understanding and reasoning. Existing baselines with partial orientation capability, including SpatialReasoner (Ma et al., 2025b) and VG-LLM (Zheng et al., 2025), perform poorly on benchmarks despite being designed for single-view reasoning and multi-view oriented-box prediction, respectively. This demonstrates our hypothesis that advances in localization grounding and grounding-guided reasoning do not automatically translate

Table 1: Results on multi- and single-view scenes. Qualitative results appear in Fig. 5 in the appendix.

Models	Multi-View Scene						Single-View Scene						
	ARKitScenes		ScanRefer*		Multi3DRefer*		Ori-COCO	RefCOCO*		RefCOCO+*		RefCOCOg*	
	Acc15 ↑	Acc30 ↑	Acc15 ↑	Acc30 ↑	Acc15 ↑	Acc30 ↑	Acc ↑	Acc15 ↑	Acc30 ↑	Acc15 ↑	Acc30 ↑	Acc15 ↑	Acc30 ↑
Spatial Reasoner	–	–	–	–	–	–	14.9	4.5	14.5	4.5	14.0	3.9	13.1
VG-LLM	10.1	17.4	17.8	26.2	21.5	33.5	–	–	–	–	–	–	–
Ours (only data)	43.1	45.4	63.1	66.6	68.1	70.6	60.1	31.1	60.9	31.1	60.8	28.8	57.8
Ours (+special token)	48.2	50.5	65.2	68.0	70.5	73.1	62.2	31.6	63.3	31.7	61.5	29.7	59.1
Ours (+geometric loss)	**50.5**	**51.6**	**66.1**	**70.1**	**71.5**	**74.2**	**64.8**	**32.0**	**63.9**	**32.2**	**62.1**	**30.2**	**60.5**

into reliable object-centric orientation grounding. Predicting oriented boxes or relying on implicit orientation cues is insufficient to recover how an object itself is oriented. Moreover, representing orientation solely with three Euler angles is limited by convention dependence, discontinuities, and axis coupling, which can hinder stable learning and accurate grounding.

In contrast, our method consistently achieves large gains across both single-view and multi-view benchmarks, indicating that orientation grounding requires explicit task formulation, dedicated supervision, and tailored modeling. These results support our central claim that orientation grounding is a natural step beyond localization grounding toward deeper 3D world understanding. Localization results (in appendix Table 12) further show that our designs also improve 3D bounding box grounding.

4.3 Comparison with Orientation Foundation Model

We further compare our OG-VLM with recent orientation foundation models (Table 2). Unlike OG-VLM, which takes a queried multi-view 3D scene as input and performs orientation grounding in a scene-centric setting, existing orientation foundation models (Wang et al., 2024b; 2026) assume access to a cropped target object for object-centric orientation estimation. Since object localization is assumed to be given, the latter setting is substantially less challenging and may require additional detection or segmentation modules in practice.

Table 2: Comparison with orientation foundation models.

Models	ARKitScenes		ScanRefer*		Multi3DRefer*	
	Acc@15	Acc@30	Acc@15	Acc@30	Acc@15	Acc@30
Orientation foundation models						
Orient Anything	18.3	35.8	17.4	34.5	16.9	33.1
Orient Anything V2	19.8	43.3	15.8	37.4	16.9	35.6
Ours (+geometric loss)	**50.5**	**51.6**	**66.1**	**70.1**	**71.5**	**74.2**

Nevertheless, OG-VLM achieves superior performance across both in-domain and OOD benchmarks. Notably, although trained exclusively on ScanNet-based multi-view data, it generalizes well to ARKitScenes, indicating that it learns transferable orientation representations rather than dataset-specific coordinate conventions. We attribute this advantage to OG-VLM's explicit grounding supervision and scene-aware reasoning, which enable it to jointly localize the queried target and infer its orientation from multi-view observations and surrounding scene context.

5 Analysis

5.1 Axis-Wise Error Analysis

As shown in Table 1, progressively adding our designs yields consistent gains, with the combination of special tokens and geometric loss performing best overall. To identify where these gains arise, we analyze axis-wise mean orientation error on RefCOCO+* and ScanRefer* in Table 3. Improvements mainly occur on the X and Y axes, corresponding to the object's right and front directions, which are also the more challenging components of orientation grounding. In contrast, Z-axis error is substantially smaller and varies little across settings, especially on ScanRefer*. This is expected because most real-scene objects are approximately upright, making the up axis strongly constrained by gravity. Thus, our method mainly improves the harder object-centric horizontal directions, while the vertical axis is already relatively easy. The consistent X/Y error reduction from *only data* to *special*

Table 3: Axis-wise mean orientation error (in degrees). X, Y, and Z denote the right, front, and up axes, respectively.

Dataset	Ablation	X (↓)	Y (↓)	Z (↓)
RefCOCO+*	only data	37.9	40.8	17.8
	special token	35.3	38.3	17.6
	geometric loss	**35.1**	**38.0**	**17.3**
ScanRefer*	only data	38.4	38.6	0.1
	special token	38.1	38.2	0.1
	geometric loss	**37.1**	**37.2**	**0.1**

token and then *geometric loss* further shows that both structured output design and geometry-aware supervision improve fine-grained orientation alignment rather than merely benefiting trivial axes.

5.2 What Affects Orientation Grounding?

Beyond the main benchmark results, we further analyze several factors that influence orientation grounding, including input depth, object symmetry, and localization quality. These analyses help characterize how visual and geometric evidence, object properties, and localization accuracy affect orientation prediction. We additionally study the effect of the number of input views in Appendix D.2.

Depth. We conduct the depth ablation in the single-view setting to isolate its effect from multi-view geometric consistency. As shown in Table 4, removing depth causes a dramatic drop in 3D box grounding (*i.e.,* localization) but a smaller degradation in orientation grounding. This suggests that depth is crucial for metric properties such as object scale, extent, and 3D location, whereas orientation can still be partially inferred from appearance, local shape, and semantic priors. Nevertheless, the consistent drops in Acc@15 and Acc@30 show that depth remains important for precise orientation grounding. Overall, 3D box grounding relies more heavily on metric reconstruction, while orientation can additionally exploit visual and semantic cues, which also helps explain why VLMs can acquire partial orientation understanding from image-language supervision.

Table 4: Effect of depth on RefCOCO+*. For localization, we report the accuracy at IoU thresholds of 25% and 50%. For orientation grounding, we report the accuracy of predictions whose orientation error falls within 15° and 30°, as well as the mean orientation error.

Model	Setting	Localization		Orientation		
		Acc 25%	Acc 50%	Mean Err. (°)	Acc@15	Acc@30
only data	**w depth**	68.7	36.3	29.6	31.1	60.8
	w/o depth	1.2	0.0	34.3	24.7	52.3
special token	**w depth**	68.2	35.6	29.5	30.9	61.5
	w/o depth	0.8	0.0	35.0	23.6	52.1
geometric loss	**w depth**	67.8	35.9	30.5	29.0	58.7
	w/o depth	2.5	0.1	33.4	24.4	53.7

Object symmetry. Symmetry directly determines whether multiple orientations should be treated as equivalent. We therefore compare *symmetric* and *non-symmetric* objects across datasets. As shown in Table 5, RefCOCO+ exhibits lower accuracy on symmetric objects, suggesting greater orientation ambiguity under limited single-view geometry. In contrast, symmetric objects achieve slightly higher accuracy on ScanRefer across all settings. We hypothesize that richer multi-view geometry reduces viewpoint ambiguity, while symmetry-aware labels admit a larger set of equivalent orientations. Consequently, symmetric objects benefit more from multi-view reasoning, whereas asymmetric objects remain more sensitive to local structure, occlusion, and grounding noise.

Table 5: Effect of object symmetry on orientation grounding.

Dataset	Model	Symmetric		Non-symmetric	
		Acc@15	Acc@30	Acc@15	Acc@30
RefCOCO+*	only data	24.41	53.76	34.41	66.56
	special token	26.76	53.05	34.44	67.36
	geometric loss	28.87	58.69	35.31	68.79
ScanRefer*	only data	64.23	66.37	59.54	64.31
	special token	65.97	68.40	58.37	63.69
	geometric loss	66.77	69.03	60.50	63.80

Effect of Symmetry and Localization Constraints. To disentangle genuine orientation-grounding ability from the effects of symmetry handling and localization errors, we compare the same model under three evaluation settings that progressively introduce symmetry adjustment and localization constraints. As shown in Table 6, removing symmetry adjustment lowers Acc@15 by 2.7 points on ScanRefer* and 7.2 points on Multi3DRefer*, but by less than 0.2 points on the RefCOCO family. The relatively modest gains indicate that the reported orientation performance is not primarily driven by symmetry adjustment; the model retains substantial orientation-grounding capability without it, while explicit symmetry modeling remains useful for intrinsically ambiguous orientations. Requiring joint correctness with localization causes a much larger drop, particularly on ScanRefer* and Multi3DRefer*, because localization and orientation must succeed simultaneously and errors from either component accumulate. This distinction helps isolate orientation quality from localization failures during evaluation. We therefore treat the joint metric as a diagnostic complement to orientation-only evaluation, while the remaining gap suggests that jointly acquiring multiple grounding capabilities remains challenging for current spatial VLMs.

Table 6: Comparison of three evaluation settings for OG-VLM. *w/o symmetry-adjusted* directly computes orientation accuracy without symmetry handling; *w/ symmetry-adjusted* accounts for symmetry-equivalent orientations when computing Acc@15 and Acc@30; and *Joint* additionally requires the predicted box to satisfy a localization IoU threshold of 0.25.

	ScanRefer*		**Multi3DRefer***		**RefCOCO***		**RefCOCO+***		**RefCOCOg***	
	Acc@15	Acc@30	Acc@15	Acc@30	Acc@15	Acc@30	Acc@15	Acc@30	Acc@15	Acc@30
w/o symmetry-adjusted	63.4	66.9	64.3	71.8	31.9	63.8	32.1	61.1	30.1	55.4
w/ symmetry-adjusted	66.1	70.1	71.5	74.2	32.0	63.9	32.2	62.1	30.2	60.5
Joint w/ IoU≥0.25	45.2	47.8	43.0	45.7	25.8	50.2	24.5	47.7	21.5	39.1

5.3 Does Orientation Grounding Benefit Downstream Spatial Reasoning?

A central question is whether orientation grounding provides information that is useful for downstream spatial reasoning. Importantly, neither model is trained on these downstream benchmarks, allowing us to evaluate whether the learned orientation capability helps downstream tasks without benchmark-specific supervision. We first evaluate on the *Orientation* subset of 3DSRBench (Ma et al., 2025a), which is a benchmark designed to assess the 3D spatial reasoning capabilities of VLMs. As shown in Fig. 4, our orientation-grounded model consistently improves over the baseline model (*i.e.,* GS-Reasoner (Chen et al., 2025b)), increasing the overall accuracy from 30.0% to 40.0%. These results indicate that, even without 3DSRBench-specific training, strengthening the model's orientation-grounding capability can directly improve spatial reasoning that relies on an object's intrinsic reference frame.

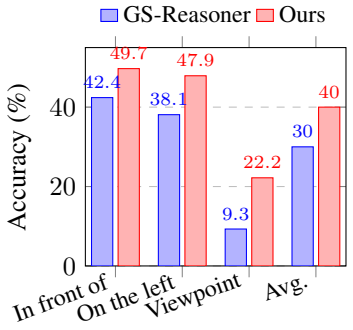


Figure 4: Results on three **Orientation** subsets of 3DSRBench (*In front of*, *On the left*, and *Viewpoint*). *Avg.* denotes the aggregate performance.

We further evaluate on MSQA (Linghu et al., 2024), which provides a complementary setting for **situated 3D spatial reasoning**, where directional relations must be interpreted under a spatial reference frame rather than as isolated geometric attributes. We follow the official evaluation protocol and report Exact Match (EM)@1 and EM@1-Strict under the same benchmark evaluation setting (Table 7). Although the improvements are less directly tied to object-centric relations than on 3DSRBench, our model still improves both EM@1 and EM@1-Strict, suggesting that directional understanding also transfers to situated spatial reasoning.

Table 7: Downstream situated spatial reasoning results on MSQA.

Model	**EM@1** ↑	**EM@1-Strict** ↑
GS-Reasoner	29.02	18.05
Ours	**32.64**	**28.18**

Together, these results suggest that orientation grounding serves as a complementary geometric cue for spatial reasoning across diverse downstream reasoning scenarios, particularly when object orientation is part of the spatial context. By providing a query-conditioned, object-centric frame of reference, it captures information that localization alone cannot recover, especially for symmetric objects and orientation-dependent referring expressions where valid orientations depend on the query. More downstream evaluation protocols are provided in Appendix C.4.

6 Conclusion

We introduced *orientation grounding* as a referring 3D grounding task that extends localization to object-centric orientation and axial symmetry, and constructed ReferOri, a scalable single- and multi-view benchmark and training resource. We show that existing spatial VLMs can acquire strong orientation-grounding capability through explicit supervision, structured outputs, and lightweight geometry-aware losses. Experiments demonstrate substantial gains over orientation-aware VLMs and object-level orientation foundation models, while downstream evaluations further show benefits for orientation-related spatial reasoning. We hope this work encourages more complete spatial grounding beyond object location alone. Future work can build richer spatial reasoning datasets upon comprehensive spatial grounding representations to further study grounding-to-reasoning transfer. We also discuss the limitations of the current formulation and evaluation in Appendix A.

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.

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7822–7831, 2021.

Sergio Arnaud, Paul McVay, Ada Martin, Arjun Majumdar, Krishna Murthy Jatavallabhula, Phillip Thomas, Ruslan Partsey, Daniel Dugas, Abha Gejji, Alexander Sax, et al. Locate 3d: Real-world object localization via self-supervised learning in 3d. *arXiv preprint arXiv:2504.14151*, 2025.

Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 2614–2623, 2019.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19129–19139, 2022.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

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.

Jianqiu Chen, Zikun Zhou, Mingshan Sun, Rui Zhao, Liwei Wu, Tianpeng Bao, and Zhenyu He. Zeropose: Cad-prompted zero-shot object 6d pose estimation in cluttered scenes. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(2):1251–1264, 2024a.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025a.

Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024b.

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

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21126–21136, 2022.

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 conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Zhiwen Fan, Panwang Pan, Peihao Wang, Yifan Jiang, Dejia Xu, and Zhangyang Wang. Pope: 6-dof promptable pose estimation of any object in any scene with one reference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7771–7781, 2024.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. Onepose++: Keypoint-free one-shot object pose estimation without cad models. *Advances in Neural Information Processing Systems*, 35: 35103–35115, 2022.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36: 20482–20494, 2023.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3707–3717, 2025.

Xiongkun Linghu, Jiangyong Huang, Xuesong Niu, Xiaojian Ma, Baoxiong Jia, and Siyuan Huang. Multi-modal situated reasoning in 3d scenes. *Advances in Neural Information Processing Systems*, 37:140903–140936, 2024.

Xiongkun Linghu, Jiangyong Huang, Baoxiong Jia, and Siyuan Huang. 3d-rft: Reinforcement fine-tuning for video-based 3d scene understanding. *arXiv preprint arXiv:2603.04976*, 2026.

Disheng Liu, Tuo Liang, Zhe Hu, Jierui Peng, Yiren Lu, Yi Xu, Yun Fu, and Yu Yin. Deconstructing spatial intelligence in vision-language models. 2025a.

Mengya Liu, Siyuan Li, Ajad Chhatkuli, Prune Truong, Luc Van Gool, and Federico Tombari. One2any: One-reference 6d pose estimation for any object. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6457–6467, 2025b.

Wufei Ma, Guofeng Zhang, Qihao Liu, Guanning Zeng, Adam Kortylewski, Yaoyao Liu, and Alan Yuille. Imagenet3d: Towards general-purpose object-level 3d understanding. *Advances in Neural Information Processing Systems*, 37:96127–96149, 2024.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso De Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6924–6934. IEEE, 2025a.

Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. *arXiv preprint arXiv:2504.20024*, 2025b.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinqiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, et al. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. *arXiv preprint arXiv:2502.13143*, 2025.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10901–10911, 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.

Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6825–6834, 2022.

Jiacong Wang, Zijian Kang, Haochen Wang, Haiyong Jiang, Jiawen Li, Bohong Wu, Ya Wang, Jiao Ran, Xiao Liang, Chao Feng, et al. Vgr: Visual grounded reasoning. *arXiv preprint arXiv:2506.11991*, 2025.

Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19757–19767, 2024a.

Zehan Wang, Ziang Zhang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient anything: Learning robust object orientation estimation from rendering 3d models. *arXiv preprint arXiv:2412.18605*, 2024b.

Zehan Wang, Ziang Zhang, Jiayang Xu, Jialei Wang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient anything v2: Unifying orientation and rotation understanding. *arXiv preprint arXiv:2601.05573*, 2026.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 803–814, 2023.

Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pp. 75–82. IEEE, 2014.

Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *European conference on computer vision*, pp. 160–176. Springer, 2016.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arxiv 2017. *arXiv preprint arXiv:1711.00199*, 10, 2017.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pp. 69–85. Springer, 2016.

Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Leizhang, Chunyuan Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Jiyao Zhang, Mingdong Wu, and Hao Dong. Genpose: Generative category-level object pose estimation via diffusion models. *arXiv preprint arXiv:2306.10531*, 2023a.

Jiyao Zhang, Weiyao Huang, Bo Peng, Mingdong Wu, Fei Hu, Zijian Chen, Bo Zhao, and Hao Dong. Omni6dpose: A benchmark and model for universal 6d object pose estimation and tracking. In *European Conference on Computer Vision*, pp. 199–216. Springer, 2024b.

Mingxu Zhang, Xiaoqi Li, Jiahui Xu, Kaichen Zhou, Hojin Bae, Yan Shen, Chuyan Xiong, and Hao Dong. Sr3d: Unleashing single-view 3d reconstruction for transparent and specular object grasping. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 15816–15823. IEEE, 2025.

Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15225–15236, 2023b.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *arXiv preprint arXiv:2505.24625*, 2025.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5745–5753, 2019.

AI USE STATEMENT

Large language models were used only for language polishing and improving writing clarity. They were not used for research design, experiments, results, or scientific conclusions. All assisted text was reviewed by the authors.

A LIMITATIONS

Our annotation pipeline may introduce noise through imperfect segmentation, reconstruction ambiguity, and coordinate transformation. Manual verification and an independent annotation audit indicate high annotation consistency, although residual errors remain, particularly for ambiguous or weakly directional objects. In addition, a noticeable gap remains between our model and reconstruction-based object-level orientation estimates, suggesting that the current VLM has not yet fully absorbed the geometric cues available from reconstruction.

Although our model performs strongly on multi-view orientation grounding, dedicated orientation foundation models remain stronger in object-centric single-view orientation estimation. The two settings are also not directly equivalent: our model reasons over queried scenes, whereas orientation foundation models typically operate on cropped objects or masks. Our goal is therefore not to replace these models, but to extend orientation understanding from isolated objects to referring grounding in single- and multi-view scenes. Bridging these complementary capabilities remains an important direction for future work.

Our model is obtained by fine-tuning an existing 3D VLM, and we do not comprehensively evaluate whether this adaptation preserves all capabilities of the base model. Incorporating orientation grounding during pretraining may provide a more unified way to learn localization, orientation, and reasoning. Our robustness analyses also reveal residual reliance on upright-scene and language priors, motivating rotation-aware training, explicit gravity conditioning, and visually unanswerable examples.

Finally, the current downstream evaluation focuses on orientation-related spatial reasoning rather than full embodied systems such as manipulation or navigation. Evaluating whether improved orientation grounding translates to broader embodied performance remains future work.

B DATA CONSTRUCTION DETAILS

This appendix provides additional details for the construction of our orientation grounding data, including the single-view and multi-view pipelines, symmetry estimation, coordinate alignment, annotation-module validation, human verification, and failure cases.

B.1 SINGLE-VIEW CONSTRUCTION PIPELINE

Our single-view construction pipeline starts from existing referring grounding data with object-level bounding boxes. Given an image I and a referred object box b, we first filter out severely blurred images to avoid unstable reconstruction and ambiguous geometry.

We then apply a metric-scale monocular 3D scene reconstruction model to estimate scene depth and camera parameters. This yields a camera-to-world transformation $T_{\text{world}\leftarrow\text{cam}}$ together with a coarse 3D scene representation in a unified metric frame.

Next, we refine the object region using a 2D segmentation model. Specifically, we use the referred box b as input to obtain an object mask M, which provides a more precise object boundary than the original bounding box. The masked object, together with the original image, is fed into an object-level 3D reconstruction model, which produces:

- an object point cloud $\mathcal{P}$,
- an object canonical coordinate system,
- and an object-to-camera transformation $T_{\text{cam}\leftarrow\text{obj}}$.

We treat the canonical coordinate system of the reconstructed object as an orientation proxy for the object itself. Let $R_{\text{cam}\leftarrow\text{obj}}$ denote the rotation component of $T_{\text{cam}\leftarrow\text{obj}}$, and let $R_{\text{world}\leftarrow\text{cam}}$ denote the rotation component of $T_{\text{world}\leftarrow\text{cam}}$. The object orientation in the world frame is computed as

$$R_{\text{obj}}^{\text{world}} = R_{\text{world}\leftarrow\text{cam}}\, R_{\text{cam}\leftarrow\text{obj}}.$$

The final rotation is stored as the ground-truth object orientation in $SO(3)$, and converted into a continuous 6D representation during training.

Based on the reconstructed point cloud $\mathcal{P}$, we further estimate the object's axial symmetry labels along the x, y, and z axes. The final annotation tuple for each referred object is

$$\mathcal{A} = (B, R, S),$$

where B is the 3D bounding box, $R \in SO(3)$ is the world-frame object orientation, and $S \in \{0,1\}^3$ denotes the symmetry labels.

B.2 Multi-View Construction Pipeline

For multi-view scenes, we exploit view consistency to improve annotation reliability. Given a referred object instance in a multi-view scene, we first identify the two most informative views. In practice, we approximate informativeness by the visible object area and select the top two frames with the largest mask area or bounding-box area.

For each selected frame, we estimate the object orientation independently using the same object-level reconstruction pipeline as in the single-view setting. Each orientation estimate is then transformed into the common scene world frame using the native camera parameters provided by the multi-view scene.

Let the resulting orientations be R_1^{world} and R_2^{world}. We compute their angular discrepancy

$$\Delta(R_1, R_2) = \arccos\left(\frac{\text{tr}(R_1 R_2^\top) - 1}{2}\right).$$

If $\Delta(R_1, R_2) < \tau$, where $\tau = 5^\circ$, we directly accept the annotation. Otherwise, we retain one prediction as a coarse annotation and ask a human annotator to correct it.

This strategy greatly reduces manual effort. Instead of labeling orientation from scratch, annotators only need to adjust a coarse but already plausible initialization.

B.3 Symmetry Estimation

We estimate axial symmetry labels directly from the reconstructed point cloud $\mathcal{P}$. For each canonical axis $a \in \{x, y, z\}$, we reflect the point cloud across the corresponding symmetry plane and measure the geometric consistency between the reflected point set and the original point cloud. If the discrepancy falls below a predefined threshold, we assign the corresponding axis symmetry label as positive.

Formally, let $\text{Ref}_a(\mathcal{P})$ denote the reflected point cloud across axis a, and let $d(\cdot,\cdot)$ denote a bidirectional point-set discrepancy. We mark axis a as symmetric if

$$d\big(\mathcal{P}, \text{Ref}_a(\mathcal{P})\big) < \epsilon_a.$$

This produces the final symmetry label vector

$$S = (s_x, s_y, s_z), \quad s_a \in \{0, 1\}.$$

In practice, symmetry estimation is reliable for most rigid and structurally regular objects, but can become noisy under severe reconstruction artifacts, heavy occlusion, or near-symmetric shapes.

B.4 Coordinate Alignment and Sanity Checks

Our pipeline combines outputs from scene-level and object-level reconstruction modules, which may adopt different coordinate conventions. To ensure consistency, we explicitly verify:

- coordinate convention compatibility across modules,
- global axis flips,
- handedness mismatch,
- and world-frame alignment after transformation.

These checks are not conducted only in image space. During verification, we inspect both the original 2D image and the final reconstructed 3D scene with the transformed object orientation. This allows us to detect systematic issues that may not be obvious from a 2D projection alone, such as left-right inversion or incorrect up-axis assignment.

We find that such sanity checks are important in practice, especially when combining independently trained reconstruction modules.

B.5 Annotation Module Validation

Since SAM3D (Chen et al., 2025a) provides the object-level orientation proposals used in our annotation pipeline, we evaluate its orientation estimates on Ori-COCO as an external reference. As shown in Table 8, SAM3D achieves 96.3% accuracy, compared with 72.4% and 86.4% for Orient Anything V1 and V2, respectively. This evaluation provides an external sanity check on the orientation proposals used for large-scale annotation.

B.6 Human Verification Protocol

For single-view annotations, we evaluate reliability through repeated random human inspection. We conduct five rounds, each with 100 randomly sampled examples. During verification, annotators inspect:

Table 8: Validation of the annotation module on Ori-COCO (Acc).

Model	Acc
Orient Anything V1 (Wang et al., 2024b)	72.4
Orient Anything V2 (Wang et al., 2026)	86.4
SAM3D (Chen et al., 2025a)	96.3

- the original 2D image,
- the target object region, and
- the reconstructed 3D scene with the projected object orientation.

A sample is counted as correct only when the referred object is correctly identified and its annotated orientation is visually consistent with both the source image and reconstructed geometry. Annotators additionally check coordinate consistency, global axis flips, and handedness mismatches. Across the five rounds, 96% of the sampled annotations are judged correct on average, which we report as the *human-verified correctness rate* rather than accuracy against ground-truth annotations.

Multi-view annotation protocol. For ScanNet multi-view data, where reconstruction is less reliable, reconstruction outputs are treated only as annotation proposals. All object annotations are manually inspected. Annotators check whether the proposed canonical frame and symmetry label are consistent with the available multi-view evidence and reconstructed geometry. Incorrect proposals are corrected, while ambiguous or low-confidence cases are independently re-annotated by a second annotator. Remaining disagreements are resolved through joint inspection, and samples without reliable consensus are discarded. For weakly directional objects, symmetry is assigned only when alternative directions are genuinely equivalent; if a unique direction exists but cannot be reliably determined, the annotation is corrected or removed.

Independent audit. We further audit 200 images from each of ScanNet, RefCOCO, RefCOCO+, and RefCOCOg, for a total of 800 images. Two evaluators independently judge whether each full annotation—including the referent, orientation frame, and symmetry label—is correct, yielding 1,600 binary judgments. Table 9 reports percentage agreement and Fleiss' κ. Agreement ranges from 96.0% to 98.8%, with 97.5% overall. The lower κ values despite high raw agreement are consistent with the prevalence effect under the strong imbalance toward annotations judged as correct.

B.7 Advantages of the Pipeline

Our construction pipeline has three main advantages.

Table 9: Independent annotation audit: agreement between two evaluators.

Dataset	Agreement	Fleiss' κ
ScanNet	96.00%	0.479
RefCOCO	96.27%	0.409
RefCOCO+	98.79%	0.436
RefCOCOg	98.68%	0.493
Overall	97.45%	0.454

First, it is **scalable**. It builds upon existing referring data, segmentation, and reconstruction modules, allowing orientation annotations to be generated at large scale without exhaustive manual labeling.

Second, it supports both **single-view** and **multi-view** settings. This enables us to cover both natural-image scenarios and richer 3D scene understanding benchmarks within a unified formulation.

Third, it is **extensible**. Since the pipeline is modular, future advances in scene reconstruction and object reconstruction can be directly incorporated to improve annotation quality. In this sense, the quality of the benchmark can continue to improve with stronger underlying 3D perception models.

B.8 Failure Cases and Limitations

Although effective in practice, the pipeline still has several limitations.

Reconstruction ambiguity. The canonical coordinate system produced by object-level reconstruction is used as an orientation proxy, but it is not guaranteed to perfectly match human-defined intrinsic orientation in all cases. This issue is most pronounced for highly symmetric objects or objects with weak directional cues.

Segmentation errors. Errors in the 2D object mask may propagate to the object reconstruction stage, which can in turn affect both orientation and symmetry estimation.

Occlusion and incomplete geometry. Heavy occlusion may lead to incomplete or distorted reconstructions, making symmetry estimation and orientation inference less reliable.

Cross-module noise. The final annotation quality depends on the consistency of scene reconstruction, object reconstruction, and coordinate transformation. Although we explicitly check for major mismatches, residual noise may still remain.

We therefore regard the constructed annotations as scalable supervision with bounded noise, rather than perfectly noise-free labels.

C Evaluation Protocol Details

C.1 Unified Rotation Conversion

All continuous-orientation benchmarks are evaluated after converting each method's native output into a 3×3 rotation matrix in the same right-handed reference frame (x right, y forward, z up):

- **SpatialReasoner** (Ma et al., 2025b) predicts a front direction under an assumed world-up direction; we complete it to a full rotation matrix and map it into the shared frame.
- **VG-LLM** (Zheng et al., 2025) predicts Euler angles; we compose them into a rotation matrix following its published convention.
- **Orient Anything V1/V2** (Wang et al., 2024b; 2026) predict azimuth–polar–rotation angles; we convert them into rotation matrices following the Orient Anything convention.
- **OG-VLM** predicts a continuous 6D rotation representation; we orthogonalize it via Gram–Schmidt and map it into the same reference frame.

After conversion, all methods are evaluated using the same $SO(3)$ geodesic error, so differences in native orientation parameterization do not introduce representation-specific evaluation metrics.

C.2 Symmetry-Aware Evaluation

The symmetry-aware evaluation in Sec. 3.1 uses the model-predicted axial symmetry labels rather than ground-truth symmetry. For OG-VLM, the predicted labels $\hat{S} = (\hat{s}_x, \hat{s}_y, \hat{s}_z)$ determine the set of symmetry-equivalent transformations $\mathcal{G}(\hat{S})$, and the orientation error is computed as the minimum $SO(3)$ geodesic distance over this set.

For baselines that do not provide compatible axial symmetry predictions, including SpatialReasoner and VG-LLM, we use the identity equivalence set $\mathcal{G} = \{I\}$, corresponding to no symmetry adjustment. Thus, ground-truth symmetry labels are never used to relax orientation errors during evaluation.

Our axis-specific symmetry representation differs from the rotational-order-k representation used by Orient Anything V2 (Wang et al., 2026). As shown in Table 6, symmetry adjustment produces dataset-dependent improvements, while the relatively modest changes on most benchmarks indicate that the reported orientation performance is not primarily driven by symmetry handling.

C.3 Benchmark Splits and Metrics

RefCOCO/RefCOCO+/RefCOCOg, ScanRefer, and Multi3DRefer are constructed by converting the test splits of the corresponding referring datasets into orientation-grounding test sets while preserving their original referring structure. Ori-COCO is adopted from Orient Anything (Wang et al., 2024b). For ARKitScenes (Baruch et al., 2021), which already provides oriented 3D bounding-box annotations, we directly use its validation split following common practice.

For all continuous benchmarks, every method's native output is converted into a rotation matrix in the same right-handed reference frame before evaluation. Specifically, we convert Euler angles for VG-LLM, a front direction under an assumed world-up for SpatialReasoner, azimuth–polar–rotation angles for Orient Anything, and the continuous 6D rotation representation used by our method. All predictions are then evaluated using the same symmetry-aware $SO(3)$ geodesic metric.

Ori-COCO is treated separately because it does not provide fine-grained $SO(3)$ annotations; instead, its labels partition the horizontal orientation space into eight 45° sectors. Following its native protocol, we bin each predicted orientation into one of these sectors and report classification accuracy. Using Acc@15/Acc@30 on this benchmark would primarily reflect label granularity rather than method quality. Accordingly, Ori-COCO results in Table 1 are reported under its native metric. Table 8 uses Ori-COCO to validate SAM3D as our annotation module under the original Orient Anything protocol, whereas Table 1 uses it as an independently annotated external benchmark.

C.4 Downstream Evaluation Protocol

3DSRBench. For 3DSRBench (Ma et al., 2025a), we evaluate the **Orientation** subset, including *In front of*, *On the left*, and *Viewpoint*, following the benchmark's official evaluation protocol. The benchmark uses CircularEval, which evaluates each question under multiple answer-choice permutations and counts it as correct only when the model remains correct across the permutations. Its test set also includes FlipEval-augmented examples to reduce left/right response bias. We report accuracy for each relation type and the aggregate accuracy over the Orientation subset.

MSQA. For MSQA (Linghu et al., 2024), we follow the MSR3D evaluation protocol and report **EM@1** and **EM@1-Strict**. Predictions and reference answers are normalized by lowercasing, whitespace normalization, punctuation removal, spelling correction, number-word conversion, and article removal where applicable. EM@1-Strict requires an exact match with at least one normalized reference answer, whereas EM@1 additionally accepts containment matches between the normalized prediction and reference answers. Empty predictions receive zero under both metrics, and dataset-level scores are averaged over all evaluated questions.

Neither model receives benchmark-specific QA supervision or task-specific fine-tuning on 3DSR-Bench or MSQA.

D Additional Results and Analyses

D.1 Sensitivity to Image-Plane Rotation

We evaluate OG-VLM on inputs rotated in the image plane by $90°$, $180°$, and $270°$. As shown in Table 10, $90°$ and $270°$ rotations break the alignment between image-top and world-up and increase the Z-axis error to roughly $84°$, collapsing Acc@15; a $180°$ rotation preserves the vertical axis and mainly increases horizontal facing errors. This reveals a reliance on upright-scene priors; rotation-aware augmentation or explicit gravity conditioning are natural remedies, which we leave to future.

Table 10: Sensitivity of OG-VLM to image-plane rotation of the input view. Axis-wise and overall mean orientation errors are in degrees; Acc@15/Acc@30 are reported in %.

Rotation	X Err.	Y Err.	Z Err.	Mean Err.	Acc@15	Acc@30
$90°$	69.0	51.3	83.9	60.2	0.05	0.20
$180°$	40.1	44.5	18.7	42.3	16.71	37.32
$270°$	68.4	52.1	83.6	60.3	0.05	0.25

D.2 Input Frames Ablation

We study how the number of input frames affects orientation grounding on Multi3DRefer. As shown in Table 11, performance drops from multi-view to single-view, and further to language-only input, showing that orientation grounding depends strongly on visual and geometric evidence. Multi-view input provides complementary cues for resolving occlusion and structural ambiguity, while single-view input offers only limited local appearance and shape information.

Notably, the model still outputs orientation predictions even without visual input. Such non-zero performance is expected for pretrained VLMs, which retain category-level and scene-level language priors. Our goal in this ablation is therefore not to eliminate these priors, but to quantify the additional benefit of visual evidence, particularly multi-view observations, for orientation estimation. The language-only performance mainly reflects residual priors rather than genuine grounding ability; while they may support coarse guesses from object categories or common scene configurations, they are insufficient for robust orientation grounding in ambiguous or non-canonical cases.

Table 11: Results on Multi3DRefer* under different settings.

Model	Settings	Acc@15	Acc@30
Ours (only data)	multi-view	68.08	70.64
	single-view	58.43	61.46
	language	20.00	40.00
Ours (+special token)	multi-view	70.50	73.06
	single-view	56.88	60.33
	language	8.04	10.80
Ours (+geometric loss)	multi-view	71.50	74.20
	single-view	64.25	66.92
	language	22.22	22.22

D.3 Downstream Spatial Reasoning Analysis

The three orientation relations in 3DSRBench probe complementary uses of the grounded object-centric frame. *In front of* primarily depends on the intrinsic front axis, while *On the left* depends on the intrinsic right/left axis. *Viewpoint* additionally requires relating the object's intrinsic frame to the camera frame, making it a more challenging reference-frame transformation. The consistent gains across all three relations therefore suggest that improved orientation grounding benefits multiple forms of orientation-dependent spatial reasoning rather than a single relation type.

D.4 Localization Results

We additionally report localization grounding results to examine whether adding explicit orientation supervision preserves the base model's localization capability. These results complement the orientation-only evaluation in the main paper by showing the behavior of the localization component under the same training variants.

D.5 Qualitative Comparison with Baselines

Figure 5 shows that our predictions better align with ground-truth orientations than SpatialReasoner on RefCOCO+ and VG-LLM on ScanNet across both single-view and multi-view settings.

Table 12: Localization grounding results at Acc25% on ScanRefer, Multi3DRefer, RefCOCO, RefCOCO+, and RefCOCOg.

Model	ScanRefer Acc 25%	Multi3DRefer Acc 25%	RefCOCO Acc 25%	RefCOCO+ Acc 25%	RefCOCOg Acc 25%
Ours (only data)	55.68	55.43	71.54	68.91	62.37
Ours (+special token)	58.95	57.38	71.84	69.26	62.42
Ours (+geometric loss)	**61.37**	**60.08**	**72.67**	**70.31**	**63.38**



Figure 5: Qualitative orientation-grounding results on RefCOCO+ and ScanNet. For RefCOCO+, gray arrows denote ground truth, while orange and red arrows denote predictions from Ours and SpatialReasoner, respectively. For ScanNet, solid axes denote ground truth and dashed axes denote predictions from Ours and VG-LLM, respectively.

E TRAINING AND GEOMETRY-AWARE LOSS DETAILS

This appendix provides the detailed formulation of the geometry-aware losses used in Sec. 3.3.2, including the box loss, orientation loss, symmetry loss, and the implementation details of the corresponding prediction heads.

E.1 DATASET CONSTRUCTION AND TRAINING DETAILS

For multi-view scenes, we construct orientation-grounding data from Locate3D (Arnaud et al., 2025), SR3D (Zhang et al., 2025), NR3D (Achlioptas et al., 2020), ScanRefer (Chen et al., 2020), and Multi3DRefer (Zhang et al., 2023b). For single-view scenes, we use RefCOCO (Yu et al., 2016). For each referred object, we construct a language query from the original referring expression and a box query from the corresponding bounding box. The resulting dataset contains 331K multi-view queries and 387K single-view queries, with the two query types approximately balanced at a 1:1 ratio.

We use GS-Reasoner (Chen et al., 2025b) as the base model. Training is performed on 4×H200 GPUs, with each full training run taking approximately 120 hours. Loss coefficients and additional optimization details are summarized in Sec. E.7.

E.2 PREDICTION HEADS

To explicitly disentangle localization and orientation learning, we adopt multiple lightweight prediction heads on top of the hidden states, following the spirit of VGR (Wang et al., 2025), which shows that geometric supervision can be effectively introduced through small MLP heads over language model representations.

Concretely, we use:

- a **box head** on the hidden state at the `</bbox>` token to regress a 3D box,
- an **orientation head** on the hidden state at the `</ori>` token to regress the continuous 6D orientation representation,
- and a **symmetry head** to predict symmetry-related geometry.

This decoupling reduces interference between localization and orientation prediction, while allowing each sub-task to receive task-specific geometric supervision.

E.3 DETAILED BOX LOSS

For 3D grounding, the model predicts an axis-aligned box

$$B = [x_{\min}, y_{\min}, z_{\min}, x_{\max}, y_{\max}, z_{\max}] \in \mathbb{R}^6. \tag{7}$$

Let B_p and B_g denote the predicted and ground-truth boxes, respectively. The box loss in Eq. (3) consists of three terms:

$$\mathcal{L}_{\text{box}} = \alpha_{\text{giou}}\mathcal{L}_{\text{GIoU}} + \alpha_{\text{scale}}\mathcal{L}_{\text{scale}} + \alpha_{\text{center}}\mathcal{L}_{\text{center}}. \tag{8}$$

GIoU loss. We first define the 3D IoU as

$$\text{IoU}(B_p, B_g) = \frac{|B_p \cap B_g|}{|B_p \cup B_g|}. \tag{9}$$

Let C denote the smallest enclosing box covering both B_p and B_g. The generalized IoU is

$$\text{GIoU}(B_p, B_g) = \text{IoU}(B_p, B_g) - \frac{|C| - |B_p \cup B_g|}{|C|}, \tag{10}$$

and the corresponding loss is

$$\mathcal{L}_{\text{GIoU}} = 1 - \text{GIoU}(B_p, B_g). \tag{11}$$

Scale loss. Let

$$\mathbf{s}_p = \mathbf{b}^p_{\max} - \mathbf{b}^p_{\min}, \qquad \mathbf{s}_g = \mathbf{b}^g_{\max} - \mathbf{b}^g_{\min} \tag{12}$$

denote the predicted and ground-truth box sizes. We measure the relative scale discrepancy along each axis as

$$\mathcal{L}_{\text{scale}} = \frac{1}{3} \sum_{k \in \{x,y,z\}} \frac{|s_{p,k} - s_{g,k}|}{\max(s_{p,k}, s_{g,k}) + \epsilon}. \tag{13}$$

Center loss. Let

$$\mathbf{c}_p = \frac{\mathbf{b}^p_{\min} + \mathbf{b}^p_{\max}}{2}, \qquad \mathbf{c}_g = \frac{\mathbf{b}^g_{\min} + \mathbf{b}^g_{\max}}{2} \tag{14}$$

denote the predicted and ground-truth box centers, and let

$$d_g = \|\mathbf{b}^g_{\max} - \mathbf{b}^g_{\min}\|_2 \tag{15}$$

be the diagonal length of the ground-truth box. We define the normalized center loss as

$$\mathcal{L}_{\text{center}} = \min\left(\frac{\|\mathbf{c}_p - \mathbf{c}_g\|_2}{d_g + \epsilon}, 1\right). \tag{16}$$

E.4 Detailed Orientation Loss

The ground-truth orientation is represented as a rotation $R \in SO(3)$, while training adopts its continuous 6D representation. From the hidden state at the `</ori>` token, the orientation head regresses two 3D vectors

$$\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^3. \tag{17}$$

We then apply Gram–Schmidt orthogonalization to recover two orthonormal axes $\hat{\mathbf{r}}$ and $\hat{\mathbf{f}}$, corresponding to the predicted right and front directions.

Let $\mathbf{r}, \mathbf{f}$ denote the ground-truth right and front axes. As defined in Eq. (4), the orientation loss is

$$\mathcal{L}_{\text{ori}} = \beta_{\text{align}}\mathcal{L}_{\text{align}} + \beta_{\text{ortho}}\mathcal{L}_{\text{ortho}}. \tag{18}$$

Axis alignment loss. We define

$$\mathcal{L}_{\text{align}} = \frac{1}{2}\left[\left(1 - \cos(\hat{\mathbf{r}}, \mathbf{r}) + \epsilon\right)^{\lambda} + \left(1 - \cos(\hat{\mathbf{f}}, \mathbf{f}) + \epsilon\right)^{\lambda}\right]. \tag{19}$$

Compared with a plain cosine loss, the exponent λ allows us to adapt the sensitivity of the alignment term during training. When the alignment error becomes sufficiently small, decreasing λ prevents the loss from saturating too early and preserves optimization sensitivity in the low-error regime.

Orthogonality loss. To ensure that the predicted axes remain geometrically valid, we use

$$\mathcal{L}_{\text{ortho}} = |\cos(\hat{\mathbf{r}}, \hat{\mathbf{f}})|. \tag{20}$$

This term penalizes non-orthogonal predictions and stabilizes the recovered orientation basis.

E.5 Adaptive Exponent Schedule

In practice, the exponent λ in Eq. (19) is adaptively reduced during training:

$$\lambda = \begin{cases} 1, & \mathcal{L}_{\text{align}} > \tau_1, \\ 1/2, & \tau_2 < \mathcal{L}_{\text{align}} \le \tau_1, \\ 1/4, & \mathcal{L}_{\text{align}} \le \tau_2, \end{cases} \tag{21}$$

where τ_1 and τ_2 are predefined thresholds. Intuitively, this strategy makes the optimization initially focus on coarse orientation alignment, and later remain sensitive to fine-grained directional refinement.

E.6 Detailed Symmetry Loss

To preserve axis-level structural regularity, we further supervise the predicted symmetry representation. Let $\hat{\mathbf{s}}$ and $\mathbf{s}$ denote the predicted and ground-truth symmetry vectors, respectively. We use a cosine-based loss:

$$\mathcal{L}_{\text{sym}} = 1 - \cos(\hat{\mathbf{s}}, \mathbf{s}). \tag{22}$$

This loss complements the token-level symmetry prediction by encouraging the model to preserve geometric symmetry structure in a continuous feature space. In this sense, token-level decoding captures the discrete output format, while the auxiliary symmetry loss provides additional geometric regularization.

E.7 Hyperparameters and Implementation Details

The default coefficients are set as

$$\alpha_{\text{giou}} = 0.5, \quad \alpha_{\text{scale}} = 0.25, \quad \alpha_{\text{center}} = 0.25, \quad \beta_{\text{align}} = 0.8, \quad \beta_{\text{ortho}} = 0.2.$$

For the overall objective, $\lambda_{\text{box}}, \lambda_{\text{ori}}, \lambda_{\text{sym}}$ are linearly warmed up during training. In addition, after step 2000, span-aware weighting is applied to the language modeling loss, with weight 3 for tokens inside the `<bbox>` and `<ori>` spans and weight 1 elsewhere.