OMNIGROUNDER: NATIVE-CAMERA 3D GROUNDING FROM DIVERSE DATA AND GEOMETRIC PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied AI is pushing vision-language models from naming what they see toward locating it, and the core capability behind it is 3D grounding: mapping an image and a text query to an oriented 3D box that specifies an object's location, size, and orientation. However, an object's apparent size depends jointly on its metric size, depth, and camera focal length, making it difficult to recover metric 3D boxes across cameras. In this work, we present OmniGrounding, a native-camera dataset, and OmniGrounder, a vision-language grounding model for this task. The OmniGrounding aligns camera annotations, oriented 3D boxes, and language instructions across multiple domains with a newly proposed game-video source, providing 4.5M bounding boxes with 3M question–answer pairs for detection and referring. The OmniGrounder injects geometric priors from pretrained 3D foundation model and learns camera-discriminative features by focal-length estimation. In the mid-training ablation, geometry injection reduces focal-length mean absolute error (MAE) by 23%, from 62.3 px to 47.9 px, compared with the same-backbone baseline. Without training on WildDet3D, our full model achieves 31.48 AP_{3D}^{dist}, outperforming the baseline (20.16 AP_{3D}^{dist}) significantly. Our dataset, model, and pretrained weights will be publicly available.

1 INTRODUCTION

Embodied agents require metric 3D perception to navigate and interact with the physical world. Language-conditioned 3D grounding provides this capability by localizing a queried object with its metric position, size, and orientation from an image or video. Because the predicted box is expressed in the coordinate frame of the camera that captured the input, we call this task *native-camera 3D grounding*. It must hold across the heterogeneous cameras an agent encounters, whose focal length, resolution, and capture geometry vary from one observation to the next.

Native-camera 3D grounding requires four ingredients: visual observations, camera parameters, oriented 3D boxes, and language annotations. No existing dataset provides all four together across domains at scale. Datasets are organized around single domains—driving scenes, indoor rooms, or object-centric clips—and differ in camera conventions, box parameterizations, category vocabularies, and annotation protocols, so their supervision cannot be pooled directly. On the model side, perspective projection entangles object size, depth, and focal length, so metric geometry is ambiguous from a single image, and existing methods recover it only partially. Grounding robustly across diverse, unseen cameras therefore remains open.

We address both gaps with one principle: keep every annotation in the coordinate frame of its source camera. This *native-camera contract* lets heterogeneous sources be described by a single supervision format, so data can be scaled by aligning and enriching existing collections rather than collecting everything anew. On the model side, we complement the vision-language backbone with geometric priors from a frozen reconstruction model, which supply the metric cues that a single image lacks. Together they yield OmniGrounding, a large-scale native-camera 3D grounding corpus, and OmniGrounder, a grounder trained on it.

For data, we avoid the cost of collecting and annotating new real-world 3D data in two ways. First, we convert existing datasets into the native-camera contract, reconciling axis conventions, dimension order, and rotation basis so that indoor, driving, and object-centric sources can be mixed. Second, we mine game-engine videos, which provide diverse scenes and geometrically consistent observations

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.

at no annotation cost, into open-vocabulary boxes with a semi-automatic pipeline: Pi3X recovers metric scale, a VLM with SAM 3 and Grounding DINO discovers and tracks objects, and gravity alignment with multi-view VLM inspection fits and verifies oriented boxes. The result is OmniGrounding, which contains 4.5M oriented 3D boxes and 3M question–answer pairs for detection and referring, together with 10M image–caption pairs that carry focal-length targets.

For the model, OmniGrounder builds on a vision-language backbone and injects geometric features from a frozen reconstruction encoder (VGGT-Omega) through a trainable residual path at selected decoder layers. The backbone provides semantics and language grounding, while the reconstruction features provide the metric geometry that a single view cannot. To work with unknown intrinsics, the model conditions box generation on focal length, using a supplied value when available and otherwise predicting it. It is trained in three stages: mid-training for scene and camera supervision, supervised fine-tuning for object-level boxes and referring, and reinforcement learning from geometric and formatting feedback.

Experiments show that OmniGrounder improves native-camera grounding over a same-backbone baseline across five benchmarks with different cameras and annotation conventions: Recall@0.15 rises from 42.89 to 44.62 on SUN RGB-D, from 58.30 to 64.65 on ARKitScenes, and from 17.21 to 19.77 on Hypersim. On WildDet3D, which is excluded from training, it reaches 31.48 $\mathrm{AP}_{3D}^{\mathrm{dist}}$ against a 20.16 baseline, and on ScanRefer it raises $\mathrm{Acc@}0.25$ from 0.79 to 5.89. In the mid-training ablation, geometry injection lowers focal-length MAE from 62.3 to 47.9 px.

Our contributions are as follows:

- We construct OmniGrounding, a native-camera 3D grounding dataset that aligns cameras, boxes, and language across indoor, driving, object-centric, and game-video sources, with 4.5M oriented 3D boxes, 3M question–answer pairs, and 10M focal-length caption pairs.
- We develop OmniGrounder, a vision-language grounder that injects geometry from a frozen reconstruction encoder into the decoder and conditions box prediction on supplied or estimated focal length, producing native-camera 3D boxes from images and multi-frame clips.
- We show on five benchmarks that OmniGrounder outperforms a same-backbone baseline across domains, transfers to WildDet3D without training on it, and that geometric injection improves focal-length recovery.

2 Related Work

2.1 3D Grounding Datasets

Existing datasets for 3D grounding are typically organized around individual domains: nuScenes (Caesar et al., 2020) and KITTI (Geiger et al., 2012) cover driving scenes; SUN RGB-D (Song et al., 2015), ScanNet (Dai et al., 2017), and ARKitScenes (Baruch et al., 2021) cover indoor environments; and Objectron provides object-centric videos (Ahmadyan et al., 2021). Their complementary coverage is valuable, but differences in camera conventions, box annotations, and category vocabularies prevent direct pooling into consistent grounding supervision. Referring datasets add instance-specific language: ScanRefer (Chen et al., 2020) and Nr3D/Sr3D (Achlioptas et al., 2020) describe indoor objects, while Talk2Car (Deruyttere et al., 2019) provides driving commands. However, this language supervision remains tied to particular scene domains and object collections. Omni3D broadens geometric coverage by unifying detection annotations across datasets (Brazil et al., 2023), but does not itself supply the rich referring language needed for object-specific grounding. Its accompanying Cube R-CNN model uses virtual-camera depth normalization, requiring focal-length-dependent conversion of each predicted depth back to the input camera's metric geometry.

These limitations motivate scaling grounding data through annotation alignment and enrichment rather than virtual-camera normalization. OmniGrounding preserves scene, object, and native-camera diversity while unifying coordinate conventions, metric units, and box orientations, refining and adding 3D boxes, and providing verified spatial and appearance descriptions for broader, less ambiguous supervision.

2.2 3D Grounding Vision-Language Models

Existing 3D grounding VLMs mainly target either qualitative spatial reasoning or generic 3D box generation. Methods for qualitative spatial relations and 3D scene understanding (Chen et al., 2024; Cheng et al., 2024; Yang et al., 2025; Hong et al., 2023; Huang et al., 2024a; Yuan et al., 2024; Cai et al., 2024) reason about spatial concepts but do not directly predict metric 3D boxes. Metric box generation methods such as Cube-LLM (Cho et al., 2025) and LocateAnything3D (Man et al., 2025) produce 3D boxes autoregressively, but their supervision is not aligned to native camera geometry across image and video sources. SpatialStack aligns multi-level geometric features with the language backbone for spatial reasoning (Zhang et al., 2026), but it does not target or evaluate metric 3D grounding. In contrast, OmniGrounder incorporates reconstruction features into autoregressive native-camera box generation, explicitly conditions the output on supplied or visually estimated focal length, and aligns annotations across image and video sources for consistent metric supervision.

2.3 Camera Calibration and Geometric Priors

Vision-language models (VLMs) learn strong image-text alignment but lack explicit 3D geometric grounding: their representations are not guaranteed to encode camera intrinsics, metric depth, or point-level geometry. Camera calibration and feed-forward reconstruction provide complementary sources of such explicit 3D information. Calibration methods recover imaging parameters from visual observations. Perspective Fields predicts per-pixel up vectors and latitude values that can be converted into calibration parameters (Jin et al., 2023). WildCamera estimates an incidence field and recovers intrinsics using RANSAC (Zhu et al., 2023). For video, DroidCalib incorporates a self-calibrating bundle-adjustment layer into a visual SLAM system (Hagemann et al., 2023). Feed-forward reconstruction goes further by predicting dense geometry: DUSt3R regresses pointmaps from uncalibrated images (Wang et al., 2024), MASt3R adds dense local features and a matching objective to improve correspondences (Leroy et al., 2024), and VGGT predicts camera parameters, point maps, depth maps, and point tracks from one or multiple views (Wang et al., 2025a). VGGT-Omega extends this formulation to process multiple frames jointly (Wang et al., 2026). These representations capture the true 3D structure of a scene, which image-text aligned VLMs cannot directly provide. However, they are not conditioned on language and do not identify which object an instruction refers to. OmniGrounder bridges this gap by incorporating frozen reconstruction features as geometric cues into the vision-language decoder, and supervising camera-conditioned object predictions with aligned annotations.

3 OmniGrounding

OmniGrounding provides images and videos paired with camera parameters, oriented 3D boxes, and language instructions, and it is organized so that a single corpus supports both camera prediction and object-level grounding. This corpus contains 4.5M oriented 3D bounding boxes and 3M question–answer pairs, together with 10M image–caption pairs carrying focal-length targets for camera supervision. A guiding principle is that *every annotation is expressed in the coordinate frame of its source camera*, so that supervision from different domains can be mixed without a virtual-camera detour. We build it in three steps—source selection, native-camera annotation construction, and task supervision construction—and the resulting annotations feed a three-stage training schedule of mid-training, supervised fine-tuning, and reinforcement learning.

3.1 Source Selection and Composition

We select sources that provide the geometry required for native-camera supervision, spanning indoor, driving, object-centric, and game-video domains. Indoor data come from SUN RGB-D, ARKitScenes, and ScanNet with HoliSpatial box annotations (Song et al., 2015; Baruch et al., 2021; Dai et al., 2017; Gao et al., 2026), complemented by synthetic scenes from Hypersim (Roberts et al., 2021). Driving data come from nuScenes and KITTI (Caesar et al., 2020; Geiger et al., 2012), and Objectron supplies object-centric imagery (Ahmadyan et al., 2021). Game videos from OmniWorld-Game (Zhou et al., 2025) extend the corpus to dynamic, open-ended scenes that existing 3D datasets do not cover.

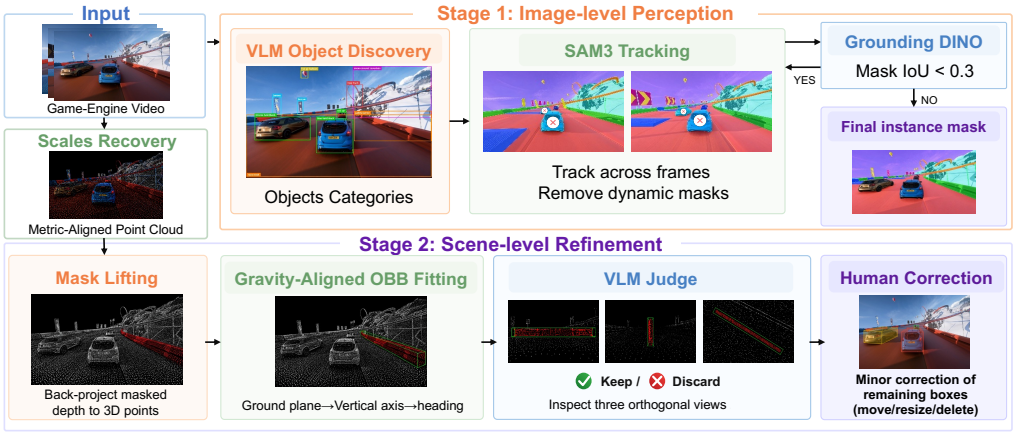

Figure 1: **Game-video annotation pipeline.** Object discovery, tracking, and candidate detection provide regions for reconstruction and box fitting. Pi3X contributes metric-scale estimates; gravity alignment, multi-view inspection, and manual correction support annotation refinement.

These sources are complementary: driving data provide long-range outdoor geometry and strong perspective effects, indoor data provide densely sampled objects at close range, object-centric data cover many categories under controlled viewpoints, and game videos contribute motion and vocabulary diversity. We convert the annotations into a common representation, filter and refine the data, and correct skewed or erroneous boxes, and on the game videos we additionally construct open-vocabulary boxes, where category names are proposed in text rather than selected from a fixed list. The heterogeneity of the sources is also the main obstacle, since camera conventions, box parameterizations, and category vocabularies differ across datasets; we therefore normalize every source to the contract in Section 3.2 rather than training on raw annotations. The resulting corpus provides 3M question–answer pairs and 4.5M bounding boxes.

3.2 Native-Camera Annotation Construction

Data processing. Each object is represented by an oriented box $[x, y, z, w, h, d, \text{roll}, \text{pitch}, \text{yaw}]$, with center coordinates and dimensions in meters. We adopt the OpenCV convention: x points right, y down, and z forward, with $z > 0$ in front of the camera. Euler angles are normalized by $180°$, and focal length is expressed in pixels. Single-image boxes are defined in the input camera frame. For video, boxes are expressed in the first camera frame, while the observation that determines the object is referenced by a frame tag, so a clip provides both multi-view context and a well-defined coordinate frame.

For sources with existing boxes, we transform annotations into the target camera frame, refine loose boxes, and discard heavily occluded objects. ScanNet/HoliSpatial annotations are used for training after rotation refinement. Appendix A provides annotation provenance.

Geme-engine videos. For game videos, depth is scale-ambiguous. We estimate metric scale with Pi3X (Wang et al., 2025b) before box fitting. A VLM proposes objects, SAM 3 tracks them across frames (Carion et al., 2025), and Grounding DINO supplies additional candidate detections (Liu et al., 2024b). After track filtering, we back-project depth within retained masks to object point clouds, separating open-vocabulary category discovery from geometric label construction.

We then fit and screen gravity-aligned boxes (Figure 1). A RANSAC ground plane determines the vertical axis, which remains fixed during heading search. Following 3DAxisPrompt (Liu et al., 2024a), we render each candidate from three orthogonal directions for VLM inspection. We reject stretched fits and fragments contained within complete boxes, followed by a light manual correction pass. Appendix A.4 provides qualitative examples.

Annotation alignment. In parallel with the game-video construction, we refine the inherited annotations under the same contract. We convert each annotation into this representation by applying

the dimension-order and rotation-basis changes jointly, and we verify the result with signed axis directions rather than corner overlap alone; for objects without a defined front, no orientation is inferred. The remaining repairs are source-specific. nuScenes and KITTI are re-derived from their native coordinate frames, and ARKitScenes, Hypersim, and SUN RGB-D correct angle serialization while isolating records whose transforms are non-rigid or degenerate. HoliSpatial boxes are re-fitted from multi-view observations under depth and gravity constraints, which corrects skewed heading, dimensions, and orientation. WildDet3D restores official valid native boxes and excludes boxes with weak 2D–3D agreement. Refinement targets geometric validity in the native camera rather than exhaustive recall: the retained annotations form a reliable subset, and disputed or degenerate boxes are excluded rather than approximated. Appendix A records the corresponding quality-control notes and summarizes representative before-and-after refinements (Figure 3).

3.3 Task and Supervision Construction

We build referring and detection queries from the object annotations. For referring examples, a VLM generates a question from an image annotated with the target box and nearby distractor objects. Queries cover six attribute groups: color and tone, material and shape, state and pose, relative size, a named landmark, and scene location. We then verify each query–object correspondence, so that the referring language is grounded in the actual geometry rather than in the caption text alone.

Each example pairs a visual observation with a task instruction. Outputs follow the autoregressive order in Section 4: scene description, focal length if absent from the input, and object descriptions with native-camera boxes. Geometric fields, including focal length, depth, dimensions, and heading, are derived from camera and box annotations, so the same annotation supports both language supervision and a numeric target. Video observations use frame tags that link each object to its observed frame, while the associated boxes remain in the first camera frame.

3.4 Training Stages and Supervision

Training proceeds in three stages: mid-training, supervised fine-tuning (SFT), and reinforcement learning (RL). These stages move from image-level supervision to object-level grounding and then to feedback-based refinement, so that camera geometry is learned before object localization and object localization is learned before response quality is optimized.

Mid-training pairs images with annotated scene captions and focal-length targets to form 10M examples, providing image-level scene and camera supervision before object-level grounding. Because focal length is the quantity that links image coordinates to metric geometry, this stage aligns the backbone's visual representation with camera geometry before any object is localized.

We use cropping and padding to cover an effective focal range of 500–3000 px, updating focal targets accordingly. A central crop retaining a linear fraction s_c, followed by resizing to the original dimensions, yields $f' = f/s_c$; padding followed by resizing with scale s_p yields $f' = fs_p$. These augmentations leave camera pose unchanged and are described in detail in Appendix D.

SFT adds object-level supervision through detection and referring question–answer pairs. The model learns to emit scene and object descriptions together with native-camera boxes, and to condition those boxes on the focal length. RL then refines grounding responses to referring queries using geometric and formatting feedback.

4 OmniGrounder

As shown in Figure 2, OmniGrounder aims to learn camera-discriminative geometric representations for native-camera 3D grounding. We introduce two key designs for this setting, *i.e.* geometric prior injection (§4.1) and focal-length token reconstruction (§4.2). Here we will introduce the details of these two designs.

4.1 Geometry Injection and Fusion

The training data for 3D grounding remain limited, making it difficult to learn reliable metric geometry solely from task-specific supervision. We therefore leverage a pretrained 3D foundation model

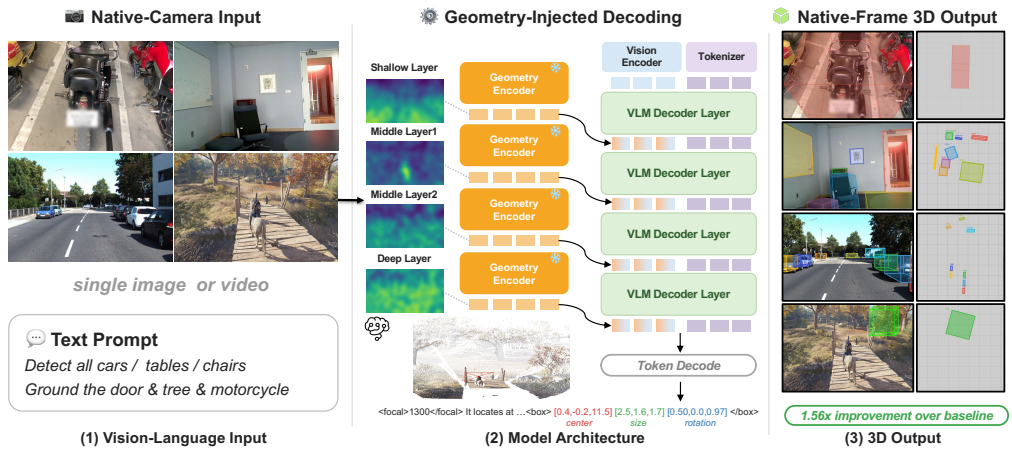

Figure 2: **Overview of OmniGrounder**. Given an image or video clip and a language instruction, the vision-language backbone predicts native-camera 3D boxes. Its decoder integrates geometric features from a frozen reconstruction encoder through a trainable fusion path. When the focal length is unavailable, the model estimates it before generating the boxes.

as an external source of geometric priors. Specifically, we extract geometric features using a frozen VGGT-Omega encoder (Wang et al., 2026), which jointly processes all frames of a video to capture cross-frame geometry. Let h_ℓ denote the decoder's visual-token features at layer ℓ, and $g_{k(\ell)}$ the aligned reconstruction features from the paired encoder layer. We apply the residual update at visual-token positions:

$$h_\ell \leftarrow h_\ell + W_o \,\mathrm{MLP}\big(\mathrm{LN}(g_{k(\ell)})\big)\,, \tag{1}$$

where LN denotes layer normalization, and the MLP followed by output projection W_o maps reconstruction features to the decoder width.

We zero-initialize W_o to preserve the backbone features at the start of training. The encoder has approximately 908M parameters, and the fusion path adds approximately 201M trainable parameters.

4.2 Training Procedure and Objectives

Mid-training and SFT. Mid-training and SFT share the same autoregressive objective. Given target tokens $y_{1:T}$ and their prefix $y_{<t}$, the language loss is the standard token-level cross-entropy:

$$\mathcal{L}_{\text{text}} = -\sum_{t=1}^{T} \log p_\theta(y_t \mid y_{<t}, x), \tag{2}$$

where x denotes the visual input and instruction. When focal-length supervision is available, we additionally predict this item and supervise it with L_2 loss:

$$\mathcal{L}_{\text{foc}} = \left(\hat{f} - f^\star\right)^2, \tag{3}$$

where $\hat{f}$ and $f^\star$ denote the predicted and ground-truth focal lengths in pixels. The overall objective is:

$$\mathcal{L} = \mathcal{L}_{\text{text}} + \lambda_{\text{foc}}\,\mathcal{L}_{\text{foc}}, \tag{4}$$

with a scalar weight λ_{foc}. Mid-training applies this objective to scene descriptions and focal-length targets, whereas SFT uses detection and referring question–answer pairs containing object descriptions and oriented 3D boxes.

GRPO and reward design. After SFT, we apply GRPO to refine referring predictions. For each prompt, GRPO samples multiple responses and normalizes their rewards within the group, so that the model is optimized toward responses better than its own alternatives. In the focal-estimation setting, the reward is:

$$R = \lambda\, r_{\text{box}} + \gamma\, r_{\text{foc}} + \beta\, r_{\text{fmt}}, \tag{5}$$

where λ, γ, and β are scalar weights. The three reward terms measure box quality, focal accuracy, and output format:

$$r_{\text{box}} = \text{IoU} \cdot \text{prec}^{1/2} \cdot \text{recall} \cdot g_{\text{proj}} \cdot g_{\text{depth}}. \tag{6}$$

Precision penalizes duplicate predictions, recall penalizes missed referents, and g_{proj} penalizes off-screen centers.

To reward both depth accuracy and focal–depth consistency, we define log residuals $d_z = \log(\hat{z}/z^\star)$ and $d_f = \log(\hat{f}/f^\star)$, where hats denote predictions and stars denote targets, and use

$$g_{\text{depth}} = \underbrace{\exp(-|d_z|)}_{\text{depth accuracy}} \cdot \underbrace{\exp(-|d_z - d_f|)}_{\text{focal–depth consistency}}. \tag{7}$$

This factor is maximized only when both residuals vanish, penalizing depth errors even when they match focal errors. The consistency term is important because a box can have the right image-space position yet the wrong metric depth if the focal length is misestimated; requiring the two log residuals to agree prevents such errors from cancelling.

Inference. At test time the model generates a response with the same structure used during training. If the focal length is supplied, it is injected as a token and the model skips focal prediction; otherwise the predicted focal length is used to interpret the generated boxes. Boxes are read from the object descriptions and are already in the native camera frame, so no virtual-to-native conversion is required before evaluation.

5 Experiments

5.1 Experimental Setup

Benchmarks and baselines. We evaluate OmniGrounder on SUN RGB-D, ARKitScenes, Hypersim, WildDet3D, and ScanRefer, using a same-backbone baseline as the primary comparison and external methods in Table 1 for context. External methods fall into two groups: direct box predictors (*Grd.*) and embodied VLMs (*Emb.*). OmniGrounder uses Qwen3.5-9B (Qwen Team, 2026) and adds 201M trainable fusion parameters. Focal-length ablations use one epoch of mid-training on the 10M focal-caption corpus without supplied focal length. We organize the results into two dataset studies, which test whether the aligned supervision transfers across domains and held-out sources, and two model studies, which isolate the geometric prior and the reasoning trace.

Metrics. We measure focal-length recovery using mean absolute error (MAE) in pixels and mean relative error (MRE). Grounding is evaluated using mean per-image Recall at $\text{IoU}_{3D} = 0.15$ on the indoor benchmarks, $\text{Acc@}0.25$ on ScanRefer, and center-distance $\text{AP}_{3D}^{\text{dist}}$ on WildDet3D (Huang et al., 2026). Grounding results are reported on each benchmark's held-out evaluation split; the source datasets provide their own test splits, and we do not reuse training images for evaluation. Comparisons are restricted to results obtained under the same metric and protocol.

Evaluation protocols. All evaluations follow the same task protocol. SUN RGB-D, ARKitScenes, Hypersim, and WildDet3D are category-conditioned detection benchmarks, where the instruction names a target category and the model predicts a box for every matching object; ScanRefer is a referring benchmark, where the instruction identifies a single target object. Ground-truth focal length is never supplied at test time, so every reported number uses the focal length estimated by the model. Video inputs are provided as 10 sampled frames, and the grounding results in Tables 1, 2, and the trace ablation in Table 4 come from the checkpoint after RL.

5.2 Main Results

Grounding results across domains. Table 1 shows that OmniGrounder without WildDet3D training outperforms the same-backbone baseline across all five benchmarks. The largest gain among the three indoor benchmarks occurs on ARKitScenes, where Recall rises from 58.30 to 64.65 (+6.35 points), compared with an increase from 42.89 to 44.62 (+1.73 points) on SUN RGB-D. Recall also improves from 17.21 to 19.77 on Hypersim, extending the observed gains to synthetic indoor imagery.

Table 1: 3D grounding results. Best results are bold; second-best are underlined. *Grd.*: direct box predictors; *Emb.*: embodied VLMs. External methods are contextual comparisons, not training- or capacity-matched controls. Our baseline and primary model exclude WildDet3D from training; the final row reports a training-inclusive setting. Δ is computed between the baseline and OmniGrounder without WildDet3D training.

	Model	SUN RGB-D	ARKitScenes	Hypersim	WildDet3D	ScanRefer
		Recall (IoU$_{3D}$=0.15) ↑			AP_{3D}^{dist} ↑	Acc@0.25 ↑
Grd.	OVM3D-Det (Huang et al., 2024b)	38.46	24.81	–	–	–
Grd.	Qwen3-VL-8B (Bai et al., 2025)	40.01	54.36	16.4	26.21	0.02
Emb.	Pelican-VL-7B (Zhang et al., 2025b)	3.10	3.13	0.13	9.74	0.02
Emb.	MIMO-Embodied-8B (Hao et al., 2025)	2.05	3.98	1.98	10.13	0.04
Emb.	RoboBrain 2.5-8B (Tan et al., 2026)	34.95	5.30	7.71	29.35	1.32
Emb.	ACE-Brain-0-8B (Gong et al., 2026)	18.78	33.56	7.78	17.45	0.94
Emb.	Rynnbrain-8B (Dang et al., 2026)	26.12	48.19	16.07	11.74	0.54
	Baseline (no geometric prior)	42.89	58.30	17.21	20.16	0.79
	OmniGrounder (no WildDet3D training)	<u>44.62</u>	<u>64.65</u>	<u>19.77</u>	<u>31.48</u>	<u>5.89</u>
	Δ (vs. baseline)	+1.73	+6.35	+2.56	+11.32	+5.10
	OmniGrounder (+ WildDet3D training)	**48.95**	**68.32**	**22.74**	**37.38**	**5.92**

Table 2: Calibrated-image evaluation. "Used" identifies a training-source dataset, not confirmed image-level overlap. Grounding uses a distance-based metric.

Dataset	Training status	Focal MAE (px) ↓	Focal MRE (%) ↓	AP_{3D}^{dist} ↑
SUN RGB-D	used	0.21	0.04	56.38
ARKitScenes	used	6.36	0.40	26.37
3RScan	held out	55.48	7.27	9.22

Beyond the indoor benchmarks, OmniGrounder achieves 31.48 $\mathrm{AP}_{3D}^{\mathrm{dist}}$ on the unseen WildDet3D dataset, compared with 20.16 for the baseline. On ScanRefer, $\mathrm{Acc@0.25}$ improves from 0.79 to 5.89, although the low absolute accuracy shows that reliable multi-frame grounding remains challenging. Consistent gains across five benchmarks demonstrate that aligned native-camera supervision transfers across scene types and annotation conventions.

Grounding results across domains. To evaluate cross-source transfer, Table 2 compares SUN RGB-D and ARKitScenes, which contribute training data, with 3RScan (Wald et al., 2019), which is excluded from all training stages. On 3RScan, OmniGrounder achieves 9.22 $\mathrm{AP}_{3D}^{\mathrm{dist}}$, 7.27% focal MRE, and 55.48 px focal MAE, demonstrating native-camera grounding on an unseen source. Its lower performance than on the in-domain datasets reflects the remaining domain gap in sensor characteristics, depth ranges, and object distributions.

Table 1 shows a similar source-level effect: excluding WildDet3D from training yields 31.48 $\mathrm{AP}_{3D}^{\mathrm{dist}}$, compared with 20.16 for the baseline, while including it increases performance to 37.38. The held-out gain demonstrates cross-domain transfer from aligned supervision, whereas the in-domain result indicates the additional benefit of source-specific data.

5.3 Ablations and Discussions

Geometry injection and fusion. We compare fusion operators and injection depths for focal-length recovery. Four-layer additive fusion achieves the lowest focal error in this sweep: 47.9 px MAE versus 62.3 px for the baseline, a reduction of approximately 23%, and 3.11% MRE versus 4.10% (Table 3). All focal MAEs are measured without a supplied focal length.

The comparison also separates language-model loss from focal accuracy. Two- and three-layer concatenation lower held-out loss relative to the 0.6703 baseline, but their focal MAEs rise to 70.9 and 87.3 px. A better sequence-prediction loss therefore does not guarantee a better numerical camera estimate. This distinction matters for the focal-caption supervision: its effectiveness must be examined through the predicted camera quantity, not inferred from language loss alone.

Table 3: Fusion and layer-count comparison after mid-training. Parameter counts refer to the fusion modules, not the full model. All focal MAEs are measured without a supplied focal length.

Fusion	Layers	Fusion params.	Held-out loss ↓	Focal MAE (px) ↓	MRE (%) ↓
Baseline (no geometry)	–	–	0.6703	62.3	4.10
Concat	2	201M	0.6337	70.9	4.58
Concat	3	201M	0.6350	87.3	5.74
Concat	4	201M	0.6352	50.5	3.22
Cross-attention	4	1007M	0.6339	50.5	3.27
Additive	2	201M	0.6342	52.3	3.44
Additive	3	201M	0.6330	49.8	3.15
Additive	**4**	**201M**	**0.6319**	**47.9**	**3.11**

Table 4: Reasoning-trace ablation on SUN RGB-D and WildDet3D at the RL checkpoint. "w/o <think>" is the model with the reconstruction prior but without the trace, which coincides with the Geo-only row of Table 5.

Reasoning trace	SUN RGB-D Recall (IoU$_{3D}$=0.15) ↑	WildDet3D AP_{3D}^{dist} ↑
w/o <think>	38.92	30.68
w/ <think>	**44.62**	**31.48**
Δ	+5.70	+0.80

Table 5: Geo–Think decomposition on SUN RGB-D and WildDet3D. Each row toggles the reconstruction prior and the reasoning trace. Geo+Think is the full configuration and coincides with the RL model in Tables 1 and 4; "–" marks entries the variant does not estimate.

Variant	SUN Focal MAE (px) ↓	Wild Focal MAE (px) ↓	SUN Recall@0.15 ↑	Wild AP_{3D}^{dist} ↑
Think-only	0.38	150.19	42.89	20.16
Geo-only	–	–	38.92	30.68
Geo+Think	**0.22**	**97.86**	**44.62**	**31.48**

Additive fusion has lower focal error than concatenation at every tested layer count. Four-layer additive fusion also achieves lower MAE than four-layer cross-attention, 47.9 versus 50.5 px, with reported fusion-module counts of 201M and 1007M parameters, respectively. We select it on this configuration-level accuracy and parameter comparison; these results do not constitute a repeated-trial uncertainty estimate.

Reasoning trace and geometric features. We next isolate how the output structure and the geometric prior interact. Adding a <think> reasoning trace before the boxes raises grounding by 5.70 points on SUN RGB-D (38.92 to 44.62) and 0.80 points on WildDet3D (30.68 to 31.48), as shown in Table 4. The trace gives the model a place to commit to intermediate quantities, such as the focal length and coarse object positions, before emitting the final boxes.

Table 5 disentangles the effects of geometric priors and reasoning traces. Geo-only improves WildDet3D grounding but reduces SUN RGB-D recall, whereas Think-only retains a high WildDet3D focal MAE of 150.19 px. Their combination achieves the best overall results: 0.22 px focal MAE and 44.62 Recall@0.15 on SUN RGB-D, together with 97.86 px focal MAE and 31.48 AP_{3D}^{dist} on WildDet3D. These configuration-level comparisons indicate that the two components address complementary failure modes.

6 Conclusion

OmniGrounding combines aligned camera, box, and language supervision across indoor, driving, object-centric, and game-video sources, and OmniGrounder uses that supervision together with a geometry-augmented VLM for native-camera grounding from images and videos. The dataset is central to the result: aligning annotations under a native-camera contract is what lets one model transfer across domains without virtual-camera conversion, and the supplied-or-estimated focal-length interface makes the supervision usable when intrinsics are missing. The selected fusion variant lowers focal error, while the full model improves grounding over the baseline, including on WildDet3D without training on it. Scale still relies on learned priors, and multi-frame grounding remains challenging. Further work should isolate data-source and annotation effects and test generalization under controlled camera and scene splits.

AI USE STATEMENT

Generative AI was used only for language polishing and grammar checking.

ETHICS STATEMENT

This work uses publicly available datasets and does not involve human subjects or personally identifiable information. All source datasets and game content are used under their respective licenses and terms of use.

REPRODUCIBILITY STATEMENT

Full data manifests and configurations will be provided with the released code, subject to the source dataset licenses.

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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7818–7827. IEEE, 2021.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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.

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 *CVPR*, 2023.

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 *CVPR*, 2020.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *arXiv preprint arXiv:2406.13642*, 2024.

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

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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.

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

An-Chieh Cheng, Yang Fu, Yatai Ji, Ligeng Zhu, Guanqi Zhan, Zhuoyang Zhang, Zhaojing Yang, Song Han, Yao Lu, Pavlo Molchanov, Vidya Nariyambut Murali, Jan Kautz, Xiaolong Wang, Hongxu Yin, and Sifei Liu. Grounded 3d-aware spatial vision-language modeling. *arXiv preprint arXiv:2605.30307*, 2026.

Jang Hyun Cho, Boris Ivanovic, Yulong Cao, Edward Schmerling, Yue Wang, Xinshuo Weng, Boyi Li, Yurong You, Philipp Krähenbühl, Yan Wang, and Marco Pavone. Language-image models with 3d understanding. In *International Conference on Learning Representations (ICLR)*, 2025.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2432–2443. IEEE, 2017.

Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangpin Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, et al. Rynnbrain: Open embodied foundation models. *arXiv preprint arXiv:2602.14979*, 2026.

Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie Francine Moens. Talk2car: Taking control of your self-driving car. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2088–2098, 2019.

Yuanyuan Gao, Hao Li, Yifei Liu, Xinhao Ji, Yuning Gong, Yuanjun Liao, Fangfu Liu, Manyuan Zhang, Yuchen Yang, Dan Xu, Xue Yang, Huaxi Huang, and Hongjie Zhang. Holi-spatial: Evolving video streams into holistic 3d spatial intelligence. *arXiv preprint arXiv:2603.07660*, 2026.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

Ziyang Gong, Zehang Luo, Anke Tang, Zhe Liu, Shi Fu, Zhi Hou, Ganlin Yang, Weiyun Wang, Xiaofeng Wang, Jianbo Liu, et al. Ace-brain-0: Spatial intelligence as a shared scaffold for universal embodiments. *arXiv preprint arXiv:2603.03198*, 2026.

Annika Hagemann, Moritz Knorr, and Christoph Stiller. Deep geometry-aware camera self-calibration from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Xiaoshuai Hao, Lei Zhou, Zhijian Huang, Zhiwen Hou, Yingbo Tang, Lingfeng Zhang, Guang Li, Zheng Lu, Shuhuai Ren, Xianhui Meng, et al. Mimo-embodied: X-embodied foundation model technical report. *arXiv preprint arXiv:2511.16518*, 2025.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024a.

Rui Huang, Henry Zheng, Yan Wang, Zhuofan Xia, Marco Pavone, and Gao Huang. Training an open-vocabulary monocular 3d object detection model without 3d data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Weikai Huang, Jieyu Zhang, Sijun Li, Taoyang Jia, Jiafei Duan, Yunqian Cheng, Jaemin Cho, Matthew Wallingford, Rustin Soraki, Chris Dongjoo Kim, Shuo Liu, Donovan Clay, Taira Anderson, Winson Han, Ali Farhadi, Bharath Hariharan, Zhongzheng Ren, and Ranjay Krishna. Wilddet3d: Scaling promptable 3d detection in the wild. *arXiv preprint arXiv:2604.08626*, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision (ECCV)*, 2024.

Dingning Liu et al. 3daxisprompt: Promoting the 3d grounding and reasoning in gpt-4o. In *Neurocomputing*, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024b.

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.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

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 *ICCV*, 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.

Huajie Tan, Enshen Zhou, Zhiyu Li, Yijie Xu, Yuheng Ji, Xiansheng Chen, Cheng Chi, Pengwei Wang, Huizhu Jia, Yulong Ao, et al. Robobrain 2.5: Depth in sight, time in mind. *arXiv preprint arXiv:2601.14352*, 2026.

Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-$omega$. *arXiv preprint arXiv:2605.15195*, 2026.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025b.

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

Jin Yao, Hao Gu, Xuweiyi Chen, Jiayun Wang, and Zezhou Cheng. Open vocabulary monocular 3d object detection. In *International Conference on 3D Vision (3DV)*, 2026.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. In *Conference on Robot Learning (CoRL)*, 2024.

Hanxue Zhang, Haoran Jiang, Qingsong Yao, Yanan Sun, Renrui Zhang, Hao Zhao, Hongyang Li, Hongzi Zhu, and Zetong Yang. Detect anything 3d in the wild. *arXiv preprint arXiv:2504.07958*, 2025a.

Jian Zhang, Shijie Zhou, Bangya Liu, Achuta Kadambi, and Zhiwen Fan. Spatialstack: Layered geometry-language fusion for 3d vlm spatial reasoning. *arXiv preprint arXiv:2603.27437*, 2026.

Yi Zhang, Che Liu, Xiancong Ren, Hanchu Ni, Shuai Zhang, Zeyuan Ding, Jiayu Hu, Hanzhe Shan, Zhenwei Niu, Zhaoyang Liu, et al. Pelican-vl 1.0: A foundation brain model for embodied intelligence. *arXiv preprint arXiv:2511.00108*, 2025b.

Yang Zhou, Haoyi Zhu, Wenzheng Chang, Jianjun Zhou, Yifan Wang, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. *arXiv preprint arXiv:2509.12201*, 2025.

Shengjie Zhu, Abhinav Kumar, Masa Hu, and Xiaoming Liu. Tame a wild camera: In-the-wild monocular camera calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Appendix

This appendix provides additional details and analyses that complement the main paper.

- Appendix A shows the details of the proposed dataset.
- Appendix B presents more implementation details.
- Appendix C includes more experimental results.
- Appendix D provides additional analyses.

A Dataset Details

A.1 Source Composition and Dataset Interfaces

The corpus covers indoor, driving, object-centric, and game-video observations, as described in Section 3.1. The 10M mid-training count refers to image-caption pairs, not object boxes. Table 6 gives a source-level comparison of dataset interfaces.

Table 6: Comparison of dataset interfaces. "Monocular RGB" denotes the visual input; "Detect-all" denotes category-conditioned multi-object labeling, not verified exhaustive instance coverage; "Native-frame" denotes the coordinate representation; "Referring" denotes language identifying a target.

Dataset	Monocular RGB	Detect-all	Native-frame	Referring	Open-vocab.
Omni3D	✓	✓	×	×	×
Objectron	✓	×	✓	×	×
Talk2Car	✓	×	×	✓	×
HoliSpatial	×	✓	✓	×	×
ScanRefer	×	×	✓	✓	×
Nr3D / Sr3D	×	×	✓	✓	×
SUN RGB-D	✓	✓	✓	×	×
nuScenes	✓	✓	✓	×	×
KITTI	✓	✓	✓	×	×
ARKitScenes	✓	✓	✓	×	×
WildDet3D	✓	×	×	✓	✓
OmniGrounding	✓	✓	✓	✓	✓

A.2 Annotation and Quality-Control Details

Section 3.2 distinguishes inherited annotations, refined source boxes, and newly constructed game-video boxes, together with their sources of uncertainty. Figure 3 summarizes representative before-and-after refinements for four sources, and Figure 4 provides complementary views of the retained outputs. Both galleries are qualitative: visual agreement across views is not a measurement of annotation acceptance rates or independent metric calibration.

A.3 Instruction and Output Semantics

Section 3.3 describes the scene, camera, object-description, and box supervision, including the attribute groups used for referring queries. For video, `<objects frame="i">` identifies the observation frame, whereas the associated boxes use the first camera frame. Thus, an observation's time and the coordinate frame of its box are distinct. This notation specifies semantic roles, not a complete parser or serialization contract.

A.4 Additional Game-Annotation Visualizations

The examples cover smoke and overlapping people in the bar, a wide depth range and truncated objects in the corridor, and small distant instances in the street. RGB views show the annotations in image space, while 3DGS and point-cloud views illustrate their spatial extent.

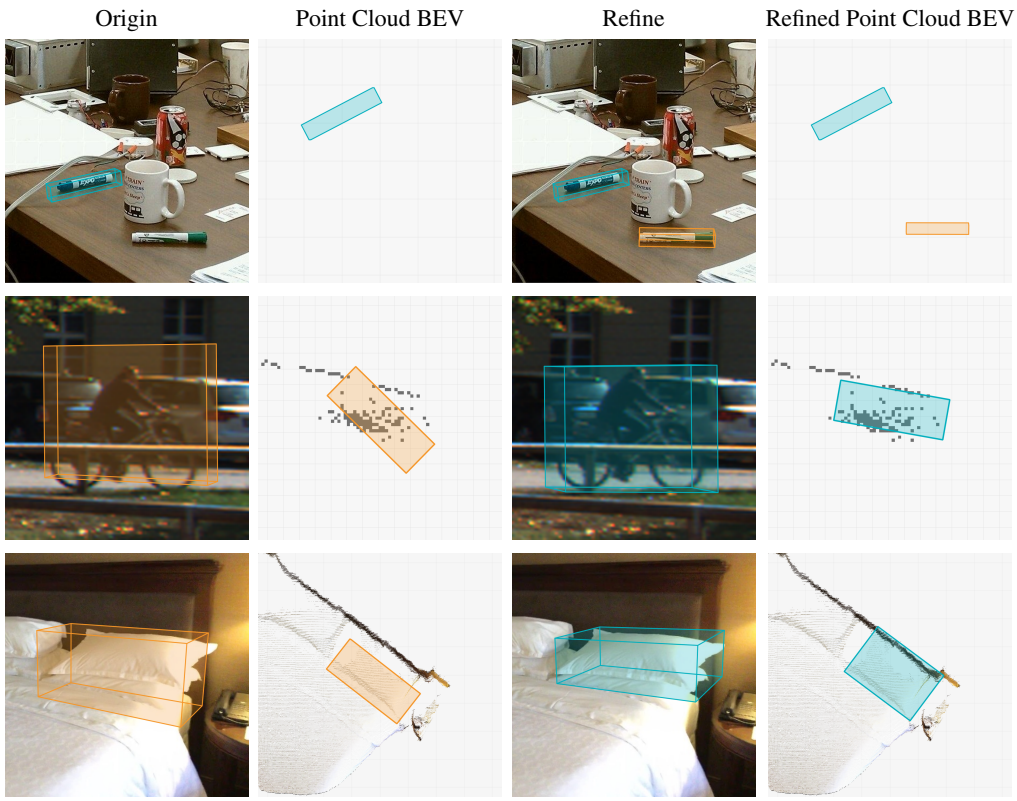

Figure 3: Representative box refinements. Columns show the original and refined annotations in RGB and bird's-eye (BEV) views under the same input and camera calibration. From top to bottom: restored official native boxes on WildDet3D, box re-fitting on a KITTI cyclist, and RGB-D re-fitting on HoliSpatial objects.

B Implementation and Training Details

B.1 Architecture and Training Configuration

We use Qwen3.5-9B (Qwen Team, 2026) with DeepSpeed ZeRO and a 64K context. Reconstruction layers $\{4, 11, 17, 23\}$ are paired with decoder layers $\{0, 1, 2, 3\}$. The reconstruction encoder has approximately 908M frozen parameters, and the fusion path approximately 201M trainable parameters. The latter is the fusion-path count, not the total trainable model size.

Mid-training runs for one epoch on 10M focal-caption pairs. Section 3.4 describes the crop/padding transformations and focal-target updates used to form these examples; Appendix D uses the same transformation convention.

B.2 Reward Definitions

Equations 5 and 6 give the focal-estimation reward and its box component. The box term combines best-matched box overlap with precision and recall, penalizing duplicate predictions and missed referents. The projection factor g_{proj} penalizes off-screen centers, while g_{depth} measures depth accuracy and focal–depth consistency (Eq. 7). The focal term is a bounded function of the log focal ratio; the format term checks the reasoning, focal, and box structure. Small factor floors prevent a single weak factor from forcing the entire box product to zero.

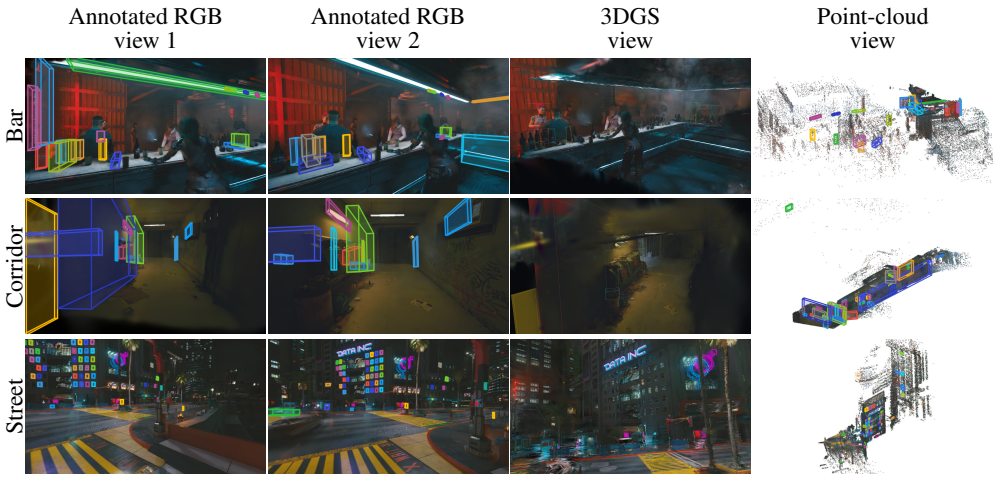

Figure 4: Qualitative game-video annotations. Each row shows two annotated RGB views, a 3D Gaussian splatting (3DGS) rendering, and a point cloud. Colors distinguish object instances.

B.3 Coordinate and Evaluation Conventions

Section 3.2 defines the camera axes, metric dimensions, and normalized Euler-angle fields. Angular-error metrics use physical units rather than this target normalization: the convention check below reports geodesic errors in degrees.

All reported OmniGrounder results use the corrected Euler-order convention. An earlier convention mismatch produced a geodesic rotation error of $98.6°$; correcting the convention reduced it to $6.8°$. This was an evaluation correction, not a model improvement.

C Additional Experimental Results

C.1 Method Capability Comparison

Table 7: Deployment-interface comparison. The columns concern focal prediction without supplied intrinsics, native-frame boxes, multi-frame inputs, and support for referring and category-conditioned detection. $\sim$ denotes partial or indirect support.

Method	Recovers focal length from an image	Native-frame 3D boxes	Single- and multi-frame	Grounding and detection
Cube R-CNN / Omni3D (Brazil et al., 2023)	×	×	×	∼
OVMono3D (Yao et al., 2026)	×	✓	×	∼
OVM3D-Det (Huang et al., 2024b)	×	×	×	∼
DetAny3D (Zhang et al., 2025a)	✓	✓	×	✓
SpatialVLM (Chen et al., 2024)	×	×	×	×
SpatialRGPT (Cheng et al., 2024)	×	×	×	∼
3D-LLM (Hong et al., 2023)	×	×	×	∼
LEO (Huang et al., 2024a)	×	×	×	∼
Cube-LLM (Cho et al., 2025)	×	×	×	∼
LocateAnything3D (Man et al., 2025)	×	✓	×	✓
GR3D (Cheng et al., 2026)	∼	✓	✓	✓
Qwen3-VL (Bai et al., 2025)	×	✓	∼	✓
OmniGrounder (ours)	✓	✓	✓	✓

Table 7 compares deployment interfaces, not prediction accuracy. DetAny3D combines intrinsic estimation with promptable monocular 3D detection. OmniGrounder uses an autoregressive vision-language interface for images and multi-frame clips. Interface coverage alone does not establish a performance ranking.

C.2 Additional SFT-to-RL Scores

Table 8: Auxiliary SFT-to-RL scores on the three indoor detection benchmarks.

Task score, Recall@0.15 ↑	SFT	+RL
SUN RGB-D (indoor)	40.78	**44.62**
ARKitScenes (indoor)	58.94	**64.65**
Hypersim (indoor, synth.)	18.92	**19.77**

Table 8 reports the SFT and +RL scores. The +RL configuration exceeds SFT by 3.84 points on SUN RGB-D and 5.71 points on ARKitScenes, and Hypersim increases from 18.92 to 19.77. These auxiliary scores are reported separately from the main grounding results.

D Calibration and Input-Transformation Analysis

D.1 Calibrated and Held-Out Datasets

Table 2 reports focal and grounding measurements on calibrated real-image datasets (Song et al., 2015; Baruch et al., 2021; Wald et al., 2019). SUN RGB-D and ARKitScenes contribute training data, whereas 3RScan is excluded from all training stages, so the held-out row evaluates transfer to an unseen source and the in-domain rows serve as references under the same protocol and metric.

D.2 Crop/Padding Transformations

The reported settings are labelled by fields of view (FOVs) of $35°$, $50°$, $69°$ (native), and $90°$. A central crop retaining linear fraction $s_c < 1$ gives $f' = f/s_c$ after resizing to the original dimensions, while padding followed by resize scale $s_p < 1$ gives $f' = fs_p$. These are digital input transformations, not newly acquired camera views. Cropping removes visual content, while padding adds a border rather than observations of a wider scene. Both change the effective FOV while preserving the source camera pose, so the transformations probe how the model responds to camera-intrinsic shifts rather than to new scene content. Table 9 reports the resulting focal and grounding metrics.

Table 9: Sensitivity to digital crop/resize and pad/resize transformations of calibrated images. No new scene content is observed through padding.

FOV	Focal MAE (px) ↓	Focal MRE (%) ↓	AP_{3D}^{dist} ↑	Recall@0.15 ↑	Depth MAE (m) ↓	Size MAE (m) ↓	Reprojection (px) ↓
$35°$	77.01	7.35	21.69	10.56	0.4687	**0.2659**	264.25
$50°$	**48.43**	**7.21**	25.03	28.26	0.3467	0.2871	131.94
$69°$ (native)	66.77	15.62	**29.97**	**30.20**	**0.3061**	0.2925	**125.61**
$90°$	182.70	56.22	26.58	10.96	0.4444	0.2907	176.05

Grounding is strongest at the native $69°$ view, which attains the highest AP_{3D}^{dist} (29.97) and Recall@0.15 (30.20) together with the lowest depth MAE (0.3061 m) and reprojection error (125.61 px); focal MAE is lowest at $50°$ (48.43 px). Because the transformations are digital approximations of a focal-length change, their effect on grounding varies with the effective FOV, and cross-FOV readings depend on the induced focal distribution and image resolution. We therefore treat this table as a sensitivity check for the crop/padding convention used in mid-training: the augmentations widen the focal coverage seen during training, and the native camera remains the operating point for which the model is calibrated.