PanoGrounder: Bridging 2D and 3D with Panoramic Scene Representations for VLM-based 3D Visual Grounding

Seongmin Jung[1⋆], Seongho Choi[1,2⋆], Gunwoo Jeon[1], Minsu Cho[3], and Jongwoo Lim[1]

[1] Seoul National University, South Korea
{sm18570, sh.choi, gunw0, jongwoo.lim}@snu.ac.kr
[2] Robotics Lab, Hyundai Motor Company, South Korea
[3] Pohang University of Science and Technology (POSTECH), South Korea
mscho@postech.ac.kr
https://choiseongho-h.github.io/PanoGrounder

Abstract. 3D Visual Grounding (3DVG) is a critical bridge from vision-language perception to robotics, requiring both language understanding and 3D scene reasoning. Traditional supervised models leverage explicit 3D geometry but exhibit limited generalization, owing to the scarcity of 3D vision-language datasets and the limited reasoning capabilities compared to modern vision-language models (VLMs). We propose a generalizable 3DVG framework, **PanoGrounder**, that couples multi-modal panoramic representation with pretrained 2D VLMs for strong vision-language reasoning. Panoramic renderings, augmented with 3D semantic and geometric features, serve as an intermediate representation between 2D and 3D, and offer two major benefits: (i) they can be directly fed to VLMs with minimal adaptation and (ii) they retain long-range object-to-object relations thanks to their 360-degree field of view. We devise a three-stage pipeline that places a compact set of panoramic viewpoints considering the scene layout and geometry, grounds a text query on each panoramic rendering with a VLM, and fuses per-view predictions into a single 3D bounding box via lifting. Our approach achieves state-of-the-art results on ScanRefer and Nr3D, and demonstrates strong generalization to unseen 3D datasets and text rephrasings.

Keywords: 3D Visual Grounding · Panoramic Vision · Vision-Language Model · Scene Understanding

1 Introduction

3D Visual Grounding (3DVG) aims to localize a target object in a 3D scene from a free-form natural language query, sitting at the interface of natural language understanding and 3D scene understanding [1, 5]. As a core capability for embodied AI, 3DVG underpins applications in augmented reality, vision-language

⋆ Equal contribution.

(a) this is a wet floor sign. it is leaning against the wall close to the sink.

(b) this picture is on the wall. it is in between two other pictures at a higher elevation.

(c) the couch is between the small table and the pile of pillows. the couch is circular with an indent in the middle.

Fig. 1: PanoGrounder localizes referred objects in 3D scenes by leveraging panoramic renderings as a 2D–3D interface. Compared to 3D-VisTA [47] (red), PanoGrounder (blue) accurately grounds (a) rare objects, (b) small objects with fine-grained spatial relations, and (c) complex queries involving multiple inter-object relationships.

navigation, and robotic perception and manipulation [2]. The task requires *language understanding*, which extracts attributes and relations from the instruction, and *3D scene understanding*, which grounds those cues in metric geometry to find the target object [24].

Traditional 3DVG systems typically employ separate encoders for text and point clouds, followed by cross-modal fusion [24]. By operating directly on the 3D scene, these models have achieved strong performance on standard benchmarks [1, 5]. Nevertheless, this design has limitations in both *language comprehension* and *3D vision generalization*. On the language side, most works rely on BERT [9] or CLIP [31]-style encoders, which offer more limited compositional and spatial reasoning capabilities than modern VLMs [21,26,28,35,41,43], making paraphrases and relation-heavy descriptions difficult to handle. On the 3D vision side, these models are trained from scratch on limited-scale 3DVG datasets [1, 5], which restricts their generalization to scenes, categories, and linguistic expressions outside of the training distribution. To leverage stronger linguistic reasoning, recent work introduces VLMs into the 3DVG pipeline. However, transferring 2D capacity to 3D remains non-trivial. Recent VLM-based methods [22,38] use perspective (pinhole) images as a 2D–3D interface, but the limited field of view fails to capture a holistic spatial context. Moreover, reliance on large proprietary models makes these approaches computationally prohibitive and difficult to fine-tune. This calls for a simpler, more efficient intermediate representation that connects 2D VLMs with 3D reasoning.

In this paper, we introduce **PanoGrounder**, which uses panoramic renderings as an explicit intermediate representation between 2D and 3D modalities. Unlike perspective images, panoramas cover a 360° field of view, capturing holistic spatial context within a single image while remaining fully compatible with VLMs. By operating on renderable representations rather than raw point clouds, this design also relaxes the high-quality 3D input requirement common in tra-

ditional 3DVG pipelines—requiring only casually captured RGB frames and an off-the-shelf SfM/NVS pipeline. PanoGrounder operates in three stages: (i) a compact set of panoramic viewpoints is selected and multi-modal panoramas—RGB, semantic, and range—are rendered for richer contextual cues; (ii) a pretrained VLM processes each panorama to predict a 2D bounding box in pixel coordinates; and (iii) the 2D grounding outputs are lifted into metric 3D space and fused across views. To incorporate semantic and geometric context from the multi-modal renderings, the VLM is augmented with a lightweight adapter, fine-tuned in the panoramic domain.

As shown in Fig. 1, PanoGrounder achieves accurate grounding on challenging cases involving rare objects, fine-grained spatial relations, and complex multi-object queries. Quantitatively, it achieves state-of-the-art performance on ScanRefer [5] and Nr3D [1]. We further assess (i) *scene generalization* on ARKitScenes+SceneVerse [3, 18] and 3RScan+RIORefer [27, 34] and (ii) *text generalization* with a modified version of ScanRefer queries. Across all settings, our method exhibits strong generalization compared to fully supervised baselines.

Our main contributions are as follows:

- We introduce a novel 3DVG approach that treats panoramic renderings as a 2D–3D interface for pretrained 2D VLMs. This design preserves scene-wide context while enabling powerful vision-language reasoning with only minimal modifications around the VLM.
- We propose an effective 3DVG model, PanoGrounder, that injects *semantic* and *geometric* features into the VLM's vision encoder via a multi-modal feature adapter and introduces an Earth Mover's Distance (EMD) loss that provides distance-aware supervision for more accurate localization.
- Our method, PanoGrounder, achieves state-of-the-art performance on ScanRefer and Nr3D. It further demonstrates strong generalization to unseen scenes and diverse text rephrasings.

2 Related Work

2.1 3D Visual Grounding

3D-based 3DVG. Traditional 3D-based methods are fully supervised, using separate encoders for text and 3D point clouds and fusing them via a cross-modal module. Two-stage approaches [1, 4, 6, 14, 47, 48] follow a proposal-and-selection paradigm, where a 3D detector first proposes candidate objects and the model selects the best match. ViewSRD [14] further improves language understanding by decomposing complex queries into simpler clauses via an LLM. One-stage methods [16, 17, 30, 37] directly regress 3D bounding boxes or masks conditioned on the query; BUTD-DETR [16] introduces language and objectness guidance into a DETR-style decoder, while MCLN [30] extends it with two parallel decoders for box-level and mask-level prediction, enhancing overall localization consistency. Despite these advances, most methods rely on BERT or CLIP text encoders and

are trained from scratch on limited-scale 3DVG datasets, constraining both spatial reasoning and generalization to unseen scenes and expressions. Furthermore, their reliance on high-quality, human-cleaned point clouds limits practical applicability, as performance degrades significantly when using raw RGBD-projected point clouds [17]. In contrast, PanoGrounder leverages pretrained VLMs via a panoramic 2D–3D interface, enabling stronger language understanding and more robust generalization without requiring curated 3D inputs.

2D-based 3DVG. Without relying on a global 3D scene point cloud, a line of works performs 3DVG directly from one or a small set of RGB(-D) views. Refer-it-in-RGBD [25] operates on single-view RGB-D and employs a coarse-to-fine framework to recover the full 3D extent of partially observed targets. Mono3DVG [42] tackles single-view monocular RGB in autonomous-driving scenes, leveraging a lightweight depth predictor. Recent zero-shot approaches [22,38–40] prompt an LLM/VLM with multi-turn instructions combining images and text to inject scene context. However, all these methods share a common limitation: the choice of viewing direction critically affects what scene content is captured, making it difficult to capture holistic spatial context across the scene. Zero-shot variants additionally rely on large models not designed for task-specific fine-tuning, making adaptation costly. In contrast, PanoGrounder uses panoramic renderings to capture scene-wide context within a single image, and its adapter-based design enables efficient end-to-end fine-tuning on task-specific data.

2.2 Multi-Modal 3D Perception

A complementary line of work jointly trains models on multiple tasks, such as 3D visual grounding, 3D scene captioning, and Visual Question Answering (VQA), to align 3D and language spaces. Several methods [23, 47, 48] adopt transformer-based alignment where task-specific heads branch from a shared backbone. UniVLG [17] further leverages abundant 2D data via a neural 2D-to-3D lifting model to transfer supervision into 3D. Beyond supervised alignment, 3D-R1 [15] explores RLHF-style policy optimization to refine instruction following in 3D contexts. Recently, Multi-modal Large Language Models (MLLMs) have been investigated by feeding 3D-aware tokens into pretrained VLM backbones [26]. Scene-LLM [11] forms hybrid voxel-point tokens from multi-view features [31] and linearly projects them into the LLM space. Chat-Scene [13] uses object proposals with per-instance Object Identifier tokens to fuse 2D/3D object features for unified object-token reasoning. LLaVA-3D [45] augments 2D patch tokens with 3D position embeddings to create 3D-aware patches for the LLM. These approaches require task-specific 3D token designs and substantial architectural modifications to bridge 3D and language spaces. In contrast, PanoGrounder bridges 3D and language spaces through panoramic renderings, enabling direct reuse of pretrained 2D VLMs with minimal architectural overhead.

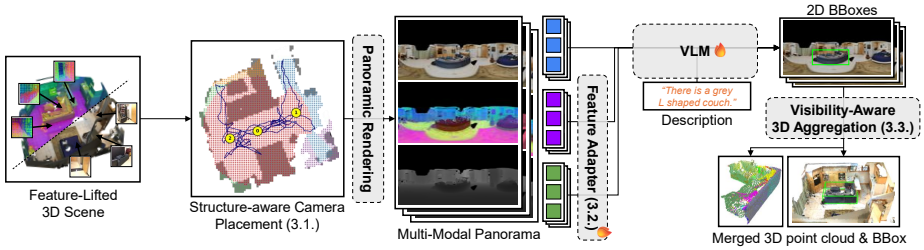

Fig. 2: Overall pipeline of PanoGrounder. We first select a compact set of informative panoramic viewpoints via Structure-aware Camera Placement and render multi-modal panoramas (RGB, lifted semantic features, and range). Each view is processed by an off-the-shelf VLM augmented with our multi-modal feature adapter to produce 2D bounding boxes from the text query. Finally, per-view 2D predictions are lifted and fused with visibility-aware 3D aggregation to yield a single 3D bounding box for the referred object.

3 Method

We assume the 3D scene is given as a renderable representation (e.g., a triangle mesh or 3D Gaussian Splatting [19]). Given such a representation and a text query, our goal is to predict a 3D bounding box of the referred object. Our key idea is to use panoramic renderings as an intermediate representation that bridges 2D vision-language understanding and 3D spatial reasoning.

As illustrated in Fig. 2, we first select a compact set of informative viewpoints (Sec. 3.1) and render multi-modal panoramas—RGB, geometric, and semantic feature maps—at each location (Sec. 3.2). These are fed into a VLM augmented with lightweight adapters that inject geometric and semantic cues (Sec. 3.2), producing per-view 2D bounding box predictions. The predictions are then lifted and fused into a 3D bounding box via visibility-aware 3D aggregation (Sec. 3.3). We train the model with cross-entropy combined with an Earth Mover's Distance loss for distance-aware supervision (Sec. 3.4).

3.1 Structure-Aware Camera Placement

Panoramic cameras capture an omnidirectional view of the scene, bypassing the need to predict a specific viewing orientation, so we only select *locations*. We start by estimating the floor via RANSAC [10] and place a regular grid $\mathcal{G}=\{p_i\}$ with $h=10\,\text{cm}$ spacing across the scene's 2D floor footprint. Each grid point p_i serves as a candidate panoramic camera location, with its height set to the average height of the raw cameras used during scene reconstruction.

Scoring factors. To score each candidate $p \in \mathcal{G}$, we construct three factors:

– **Ray coverage** $A(p) \in [0,1]$: fraction of other grid points visible from p within $r_{\max}=3\,\text{m}$, counted via obstacle-free projection onto the panoramic image plane.

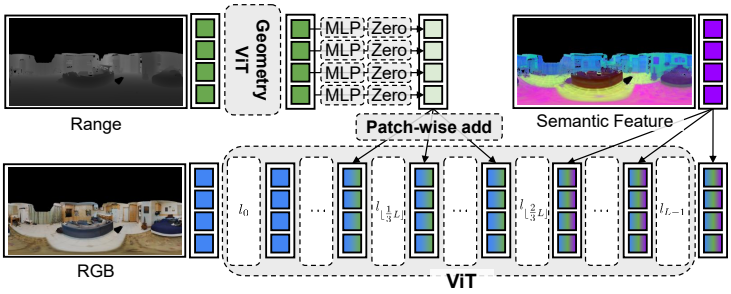

Fig. 3: Multi-modal feature adapter. We inject panoramic *geometric* (range) and *semantic* (multi-view) features into selected ViT layers of the VLM. Each modality is processed by a lightweight adapter (2-layer MLP followed by a 1×1 convolution), and the resulting output is added patch-wise to the ViT tokens. The 1×1 convolution weights and bias are initialized to zero.

- **Distance-to-surface** $D_{\text{surf}}(p)$: Euclidean distance from p to the nearest scene geometry (walls or furniture).
- **Distance-to-trajectory** $D_{\text{traj}}(p)$: Euclidean distance from p to the nearest raw RGB camera center. This term is optional—when no prior camera trajectory is available, it is simply omitted.

Score and selection. We rank candidates using

$$S(p) = \frac{A(p)\, D_{\text{surf}}(p)}{D_{\text{traj}}(p) + \varepsilon}, \quad \varepsilon{=}10^{-3}, \tag{1}$$

which balances area coverage, obstacle clearance, and trajectory proximity. Starting from the highest-scoring point, we greedily select cameras. Once a camera is selected, all grid points visible from it are marked as 'covered', effectively setting their contribution to the ray coverage $A(p)$ to zero for all remaining candidates. This sequential selection is repeated until at least 90% of the scene's grid points are covered.

3.2 Multi-Modal Panoramic VLM

For each selected viewpoint, indexed by k, we render a panoramic RGB image $I_k \in \mathbb{R}^{H\times W\times 3}$ using equirectangular projection. To better handle occlusions in cluttered indoor scenes and to provide explicit 3D information for spatial reasoning, we additionally extract geometric and semantic feature maps from the same panoramic cameras and inject them into the VLM alongside RGB via lightweight adapters (Fig. 3). All modalities share a common token grid of $N_f = H_f \times W_f$ patches (H_f and W_f are the patch counts along the image height and width), so that the injected features align patch-wise with the VLM's RGB tokens.

Geometric feature map. For each selected viewpoint, we render a range (depth) map $D_k \in \mathbb{R}^{H \times W}$. Inspired by [7], D_k is fed to a pretrained ViT [29] to obtain a d_{geo}-dimensional geometric feature map $\mathbf{f}_{\text{geo}} \in \mathbb{R}^{N_f \times d_{\text{geo}}}$ aligned with this token grid.

Multi-view fused semantic feature map. We extract dense semantic patch features from a frozen ViT encoder [29] applied to the original RGB views of the dataset. Using known camera intrinsics and extrinsics, each mesh vertex visible in a view is assigned the d_{sem}-dimensional feature of the patch it projects into. Since a vertex is typically visible from multiple views, the features it receives are averaged and re-rendered onto the selected panoramic cameras to produce a semantic feature map $\mathbf{f}_{\text{sem}} \in \mathbb{R}^{N_f \times d_{\text{sem}}}$.

Feature adapter. We inject both geometric and semantic features into the VLM's vision encoder via a shared adapter design: a 2-layer MLP followed by a 1×1 convolution whose weights and bias are initialized to zero, following Zero-Convolution [44]. At initialization the adapter output is exactly zero, so the VLM behaves identically to its pretrained form; fine-tuning then lets the adapters encode task-relevant signals without distorting the original representation space. Geometric features are injected into *mid-level* layers ($l \in [L/3,\, 2L/3)$) to provide a spatial scaffold, whereas semantic features are injected into *later* layers ($l \in [2L/3,\, L]$) to supply high-level contextual cues:

$$\mathbf{X}_{l,n} \leftarrow \mathbf{X}_{l,n} + \text{ZeroConv}_m(\text{MLP}_m(\mathbf{f}_{m,n}))\,, \tag{2}$$

where $m \in \{\text{geo}, \text{sem}\}$ indexes the modality, L is the number of VLM encoder layers, l and n are the layer and token indices, and $\mathbf{f}_{m,n}$ is the n-th token of $\mathbf{f}_m$.

Training supervision. To obtain ground-truth 2D bounding boxes, we render a per-pixel instance-ID map at each selected camera via the same equirectangular projection used for RGB. Each 3DVG dataset provides (text query, target object ID) pairs; we compute the tightest 2D bounding box enclosing all pixels of the target instance, yielding *image–text–box* triplets without any manual labeling.

3.3 Visibility-Aware 3D Aggregation

Let b_k denote the 2D bounding box predicted by the VLM for the k-th viewpoint. This stage leverages cross-view consistency to filter out erroneous individual 2D predictions and accurately localize the target object in 3D.

Mask-to-3D lifting. Given I_k and b_k, we employ an off-the-shelf segmentation model (e.g., SAM [20]) to extract a precise object mask M_k. The masked pixels are then unprojected into world coordinates using D_k and the camera extrinsics of view k, yielding a per-view 3D point set $\mathcal{P}_k$.

Anchor view selection. To identify the most reliable viewpoint, we project each candidate point set $\mathcal{P}_k$ onto all other views $t \neq k$ and compute the tight 2D bounding box $\hat{b}_{k\to t}$ enclosing the projected points. We then formulate a cross-view consistency score:

$$\text{score}(k) = \sum_{t \in \mathcal{K}\setminus\{k\}} \text{IoU}\Big(\hat{b}_{k\to t}, b_t\Big), \tag{3}$$

where $\mathcal{K}$ is the set of all selected viewpoints, and the *anchor view* is the one maximizing this score, $k^* = \arg\max_k \text{score}(k)$. This formulation explicitly favors predictions that exhibit strong geometric consensus across multiple perspectives.

Multi-view fusion. We aggregate the individual point sets into a global cloud $\mathcal{P} = \bigcup_{k\in\mathcal{K}} \mathcal{P}_k$ and apply statistical outlier removal to mitigate noise. The denoised points are then re-projected onto the anchor view k^*. Points whose projections fall outside the anchor bounding box b_{k^*} are discarded, resulting in a visibility-filtered set $\mathcal{P}_{\text{vis}}$. Finally, we fit an axis-aligned bounding box to $\mathcal{P}_{\text{vis}}$ to yield the final 3D localization.

For evaluations requiring pre-defined candidate boxes (e.g., ReferIt3D), we introduce a *two-stage* variant of our framework to ensure fair comparison. This variant preserves the core algorithmic modules, with comprehensive details provided in the supplementary material.

3.4 Training Objective

Following CogVLM [35], each bounding box coordinate is normalized to $[000, 999]$, discretized into 3 digits, and predicted as a sequence of digit tokens $\{0, \ldots, 9\}$. We supervise the autoregressive decoder with token-level cross-entropy under teacher forcing. However, standard cross-entropy treats each digit as an independent category, ignoring numerical proximity. Inspired by [32], we add an auxiliary EMD (1-Wasserstein distance) loss to inject awareness of the underlying numerical ordering.

For the i-th digit, let $X_i \in \{0, \ldots, 9\}$ be the ground-truth digit and $P^{(i)}(x)$ the predicted probability for digit x. The per-digit CE and EMD losses are:

$$\mathcal{L}_{\text{CE}}^{(i)} = -\log P^{(i)}(X_i), \qquad \mathcal{L}_{\text{EMD}}^{(i)} = \omega_i \sum_{x=0}^{9} P^{(i)}(x)\,|X_i - x|, \tag{4}$$

where ω_i is a place-value weight (e.g., $10^2, 10^1, 10^0$ for hundreds/tens/ones). The EMD term penalizes larger deviations more heavily, encouraging probability mass to concentrate near the correct digit. The final objective combines both over all 3 digits:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{3} \Big(\mathcal{L}_{\text{CE}}^{(i)} + \lambda\, \mathcal{L}_{\text{EMD}}^{(i)}\Big), \tag{5}$$

where λ balances the two terms.

To better encourage the model to leverage the injected geometric features, we augment the training data with auxiliary geometric QA pairs, inspired by [7]. Our QA samples are generated on-the-fly by randomly sampling two pixels on a panorama and asking the model to identify which has a larger 3D coordinate along a given axis. These auxiliary samples are supervised with cross-entropy only (details in the supplementary material).

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on two widely used 3D visual grounding benchmarks: ScanRefer [5] and ReferIt3D (Nr3D, Sr3D) [1]. ScanRefer annotates 51,583 human-written referring expressions across 800 ScanNet [8] scenes. Nr3D contains 41,503 human utterances over 707 ScanNet scenes spanning 76 object classes, while Sr3D consists of 83,572 template-based spatial descriptions, also referring to 76 object classes.

To assess cross-dataset generalization on unseen settings, we use ARKitScenes [3] scenes paired with SceneVerse [18] human-written referring expressions only (we exclude automatically generated texts), and 3RScan [34] scenes paired with RIORefer [27] human-written referring expressions. ARKitScenes provides 5,047 captures over 1,661 unique indoor scenes, and SceneVerse includes ARKitScenes among its sources with both human and generated texts. 3RScan comprises 1,482 scans of 478 indoor environments with instance-level annotations, and RIORefer contributes 63,602 human descriptions for 1,380 3RScan scans.

Evaluation Metrics. On ScanRefer, we report Acc@0.25 overall and on the *Unique*/*Multiple* subsets [5]. *Unique* denotes queries with no same-class distractor, while *Multiple* denotes the presence of same-class distractors. On ReferIt3D (Nr3D, Sr3D), we report Top-1 accuracy overall and on four subsets: *Easy*/*Hard* (exactly one vs. two or more same-class distractors) and *View-Dependent*/*View-Independent* [1], indicating whether resolving the description requires a specific viewpoint (e.g., "left/right"). For the generalization experiments (Sec. 4.3), we report Top-1 accuracy given ground-truth object instances and report *Unique* and *Multiple* splits using the same criterion as ScanRefer.

Implementation Details. We adopt CogVLM-17B [35] as our 2D grounding backbone, pretrained on large-scale image-text data. We fine-tune the model with LoRA, optimized by Adam with a batch size of 64 and a cosine-decay learning rate starting at 1×10^{-4}; all remaining hyperparameters follow the official CogVLM implementation. We train for 5 epochs counted scene-centrically (all panoramic views of a scene form a single pass); since each referring expression is seen from ≈ 2.4 viewpoints on average, this corresponds to roughly 12 text-centric epochs. For data augmentation, we apply random in-place camera yaw rotations, implemented as horizontal circular shifts (wrap-around) of the

Table 1: Evaluation on Nr3D, Sr3D [1], and ScanRefer [5]. To ensure a fair comparison, we group methods by training regime: models trained on a single benchmark versus those trained on mixed datasets (e.g., ScanRefer + ReferIt3D). *S+R* denotes our model trained jointly on ScanRefer and ReferIt3D (Nr3D/Sr3D). For ScanRefer, we report Accuracy@0.25 (IoU). †Results trained on that single dataset only.

	Method	Nr3D					Sr3D					ScanRefer		
		Easy	Hard	VD	VID	Overall	Easy	Hard	VD	VID	Overall	Unique	Multiple	Overall
Single Dataset	BUTD-DETR [16]	60.7	48.4	46.0	58.0	54.6	68.6	63.2	53.0	67.6	67.0	84.2	46.6	52.2
	ViL3DRel [6]	70.2	57.4	62.0	64.5	64.4	74.9	67.9	63.8	73.2	72.8	81.6	40.3	47.9
	3D-VisTA† [47]	65.9	49.4	53.7	59.4	57.5	72.1	63.6	57.9	70.1	69.6	77.4	38.7	45.9
	MIKASA [4]	69.7	59.4	65.4	64.0	64.4	78.6	67.3	**70.4**	75.4	75.2	–	–	–
	GPS† [18]	67.0	50.9	55.8	59.8	58.7	70.5	63.4	53.1	69.0	68.4	–	–	–
	MCLN [30]	–	–	–	–	59.8	–	–	–	–	68.4	86.9	52.0	57.2
	PQ3D† [48]	73.3	56.7	60.7	67.0	64.9	78.8	68.2	51.5	76.7	75.6	85.2	46.8	52.8
	LIBA [36]	–	57.2	60.3	–	64.5	–	70.2	61.7	–	75.8	88.8	54.4	59.6
	VGMamba [46]	–	61.4	–	–	68.3	–	**74.4**	–	–	**81.3**	**91.9**	54.8	60.0
	ViewSRD [14]	75.3	64.8	68.6	70.6	69.9	78.3	70.6	69.0	76.2	76.0	82.1	37.4	45.4
	Ours	**82.2**	**67.2**	**70.5**	**76.3**	**74.6**	**81.3**	74.2	60.5	**80.0**	79.1	84.3	**55.3**	**61.0**
Multi Dataset	3D-VisTA [47]	72.1	56.7	61.5	65.1	64.2	78.8	71.3	58.9	77.3	76.4	81.6	43.7	50.6
	GPS [18]	72.5	57.8	56.9	67.9	64.9	80.1	71.6	62.8	78.2	77.5	–	–	–
	PQ3D [48]	75.0	58.7	62.8	68.6	66.7	82.7	72.8	62.9	80.5	79.7	86.7	51.5	57.0
	Chat-Scene [13]	–	–	–	–	–	–	–	–	–	–	**89.6**	47.8	55.5
	LLaVA-3D [45]	–	–	–	–	–	–	–	–	–	–	–	–	50.1
	UniVLG [17]	73.3	57.0	55.1	69.9	65.2	**84.4**	**75.2**	66.2	**82.4**	**81.7**	–	–	60.7
	Ours (*S+R*)	**84.1**	**68.4**	**72.9**	**77.5**	**76.1**	82.3	74.5	**66.6**	80.6	79.9	85.0	**56.4**	**62.0**

panorama. For test time augmentation, we generate four augmented views by rotating the camera in-place by 90° increments and perform independent inference on each. For a fair comparison with prior work, all results in the main paper are obtained from mesh-rendered panoramas. However, to better reflect practical use cases, results utilizing 3DGS-rendered panoramas are provided in the supplementary material.

4.2 3D Visual Grounding Results

As summarized in Tab. 1, PanoGrounder achieves state-of-the-art or top-2 results on most splits, with clear gains on the human-annotated Nr3D and ScanRefer benchmarks. We observe that mixed training on ScanRefer and ReferIt3D (*S+R*) consistently improves performance, as also seen in 3D-VisTA [47], GPS [18], and PQ3D [48] (marked with † in Tab. 1); we therefore group rows into *Single Dataset* and *Multi Dataset* sections for a fair comparison.

On **Nr3D** [1], our model achieves the best scores across all splits under single-dataset training, significantly outperforming prior art ViewSRD by +4.7%. Joint training (S+R) further improves Nr3D to new bests. On **Sr3D** [1], our method yields Overall 79.1 (single) and 79.9 (S+R), ranking at or near the top across most splits. Notably, our model's stronger gains on Nr3D, which consists of human-generated referring expressions, highlight its robustness in handling diverse, natural language distributions compared to template-based datasets like Sr3D.

Table 2: Zero-shot comparison on Nr3D. Without any fine-tuning, PanoGrounder outperforms methods that use significantly larger backbones.

Method	Backbone	Easy	Hard	VD	VID	Overall
ZSVG3D [40]	GPT-3.5-turbo	46.5	31.7	36.8	40.0	39.0
SeeGround [22]	Qwen2-VL-72b	54.5	38.3	42.3	48.2	46.1
VLM-Grounder [38]	GPT-4V	55.2	39.5	45.8	49.4	48.0
Ours	CogVLM-17b	**64.2**	**42.7**	**49.8**	**54.7**	**53.2**

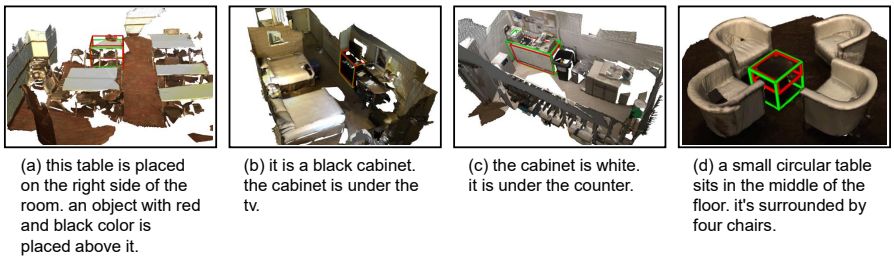

Fig. 4: Qualitative results on ScanRefer [5]. Green boxes denote ground-truth objects, and red boxes denote predictions from PanoGrounder. Across diverse scenes and query types, PanoGrounder produces accurate and spatially consistent grounding results.

On **ScanRefer** [5], our model attains the best Overall (61.0) and Multiple (55.3) scores under single-dataset training, outperforming VGMamba (60.0) and LIBA (59.6), with particularly strong gains in multi-object scenes. Joint training yields further improvements to Overall 62.0 and Multiple 56.4, both best. We note that the performance gains on Nr3D are larger than on ScanRefer: ScanRefer's *Unique* subset can often be resolved by simple attribute matching, whereas Nr3D consistently requires fine-grained relational disambiguation, where our stronger language understanding provides the greatest benefit.

Zero-Shot Comparison with Larger VLMs. Recent zero-shot 3DVG methods rely on proprietary or very large VLMs such as GPT-4V and Qwen2-VL-72B. For a fair comparison, we evaluate PanoGrounder without any fine-tuning. As shown in Tab. 2, our zero-shot pipeline with CogVLM-17B outperforms all baselines despite using a significantly smaller backbone, achieving 53.2 Overall on Nr3D compared to VLM-Grounder's 48.0 with GPT-4V. This suggests that our panoramic multi-modal representation and structured pipeline are effective, enabling strong zero-shot performance even with a smaller backbone.

Qualitative Results Fig. 4 shows qualitative grounding results on ScanRefer [5]. PanoGrounder accurately localizes the referred objects and produces

Table 3: Cross-dataset evaluation on ScanNetV2 [8], ARKitScenes [3], and 3RScan [34]. All models are trained on ScanRefer [5] and directly evaluated on unseen 3D scenes from ARKitScenes and 3RScan with corresponding text annotations. To ensure fair comparison focused on 3D scene generalization, we use GT object segmentation for all methods; * additionally leverages ground-truth semantic labels. All baseline results are re-run by us.

Method	ScanRefer			ARKitScenes			3RScan		
	unique	multiple	overall	unique	multiple	overall	unique	multiple	overall
ViL3DRel [6]	92.0	51.8	59.6	57.2	21.1	28.3	71.8	31.3	36.8
3D-VisTA [47]	89.5	49.9	57.2	59.7	26.6	32.9	<u>74.1</u>	<u>32.0</u>	<u>37.7</u>
BUTD-DETR* [16]	<u>92.5</u>	52.6	58.5	<u>66.3</u>	<u>30.6</u>	<u>36.1</u>	–	–	–
MCLN* [30]	**93.4**	<u>54.9</u>	<u>60.6</u>	61.2	30.0	35.3	–	–	–
Ours	91.7	**58.5**	**64.9**	**74.2**	**48.0**	**53.5**	**80.4**	**37.7**	**43.8**

spatially consistent 3D boxes across diverse scenes and query types. In Fig. 4(a), PanoGrounder correctly distinguishes the target table from nearby distractors using relational cues. Similarly, in Fig. 4(b–d), it resolves challenging references in cluttered scenes by combining attribute cues with spatial and relational reasoning. We also observe three common failure modes—ambiguous references satisfied by multiple objects, small or heavily occluded targets, and partial localization of multi-part objects—and provide representative examples and a detailed analysis in the supplementary material.

4.3 Generalization Analysis

To fairly evaluate scene and language generalization independently of proposal quality, we fix proposals to ground-truth object segmentations and re-run all baselines under the same setting in the following experiments.

Scene Generalization. We evaluate scene-level generalization by testing on two unseen 3D datasets: ARKitScenes [3] paired with human-written expressions from SceneVerse [18], and 3RScan [34] paired with RIORefer [27]. As shown in Tab. 3, PanoGrounder shows notably smaller performance degradation than existing baselines when transferred to unseen scenes. This indicates that our approach generalizes effectively to novel environments and diverse 3D layouts beyond the training distribution. BUTD-DETR [16] and MCLN [30] could not be evaluated on 3RScan+RIORefer because their public implementations are restricted to the ScanNetV2 object set.

Text Generalization. We assess the linguistic robustness of 3D visual grounding models on ScanRefer [5] by generating four variants of each query with LLaMA 3.3 [12]: **Para.** (paraphrased), **+Aff.** (affordance/functional description added), **+Aff.−N** (**+Aff.** with target noun removed), and **Mask** (target noun replaced with "object").

Table 4: Robustness to ScanRefer [5] text variants. Top-1 accuracy on original queries and four text modifications (see text for details). All models use GT object segmentation; * additionally uses GT semantic labels. All baseline results are re-run by us.

Method	Org.	Para.	+Aff.	+Aff.−N	Mask
ViL3DRel [6]	59.6	41.5	9.7	6.5	22.9
3D-VisTA [47]	57.2	52.9	50.4	28.3	28.5
BUTD-DETR* [16]	58.5	53.4	55.1	35.5	27.3
MCLN* [30]	60.6	54.6	55.7	35.7	31.5
PQ3D [48]	**66.0**	**63.3**	58.6	35.8	33.3
Ours	64.9	61.5	**61.3**	**51.7**	**41.8**
Δ (Ours − PQ3D)	**-1.1**	**-1.8**	**+2.7**	**+15.9**	**+8.5**

Table 5: Ablation study. We evaluate the contribution of each input modality (RGB, semantic, geometric) and auxiliary training signals (EMD loss and geometric QA).

	Input			Train		Acc@0.25
	RGB	Sem.	Geo.	EMD	Geo QA	
(A)	✓	✗	✗	✗	✗	57.5
(B)	✓	✗	✗	✓	✗	58.4
(C)	✓	✓	✗	✓	✗	60.4
(D)	✓	✓	✓	✗	✗	59.3
(E)	✓	✓	✓	✓	✗	60.2
(F)	✓	✓	✗	✓	✓	60.7
Ours	✓	✓	✓	✓	✓	**61.0**

As shown in Tab. 4, PanoGrounder performs on par with PQ3D on **Org.** and **Para.**, and surpasses it on **+Aff.** (+2.7). The gains become particularly pronounced once explicit class cues are removed, with improvements of +15.9 on **+Aff.−N** and +8.5 on **Mask**. In these two variants, where the target object name is absent, the model must reason over spatial and contextual cues rather than simple lexical matching. PanoGrounder remains accurate under such implicit or underspecified language, demonstrating strong reasoning and robust performance across diverse textual perturbations.

4.4 Ablation Study

Tab. 5 analyzes how input modalities and auxiliary objectives contribute to performance. The EMD loss consistently improves accuracy regardless of input configuration: (A)→(B) (57.5→58.4) and (D)→(E) (59.3→60.2), confirming its value as a distance-aware regularization signal. Incorporating semantic features in (C) further boosts accuracy to 60.4, indicating that multi-view fused semantic context is highly beneficial even without geometric input. Row (F) shows that geometric QA without geometric input (60.7) also falls short of the full model. Our final configuration (**Ours**), which combines RGB, semantic, and geometric features with both EMD and geometric QA, achieves the best accuracy (61.0). Notably, masking the geometric features of **Ours** at inference drops its accuracy to 60.5, confirming their utility. Together, these results suggest that geometric input and geometric QA act synergistically rather than as interchangeable components.

Panorama vs. Pinhole Cameras. Panoramic views offer two primary advantages over pinhole cameras: (i) they eliminate the need for viewing direction prediction, thereby simplifying the pipeline; and (ii) they capture the maximum number of text-referenced objects within a single view, strengthening context and relation modeling.

Table 6: Comparison of panoramic and pinhole view settings. We report accuracy by view configuration (rows) and by the number of objects referenced in the text (columns). Our panoramic representation substantially outperforms all pinhole baselines without oracle knowledge, and the performance gap widens as the query mentions more surrounding objects. The oracle GT target setting provides an upper bound for pinhole-based approaches.

View setting	**Overall**	**# of objects mentioned in the text**				
		0 (0.6%)	1 (23.7%)	2 (46.6%)	3 (21.9%)	4+ (7.2%)
pinhole @ 4 views	41.3	41.1	38.7	42.1	42.0	42.2
pinhole @ 16 views	51.0	50.0	47.3	52.4	51.0	54.5
pinhole @ Semantic [22]	43.6	37.5	42.3	44.3	43.6	44.0
Ours	**61.0**	**49.9**	**54.7**	**61.3**	**64.3**	**71.0**
pinhole @ GT target	67.5	58.9	65.5	69.3	66.5	66.0

We compare our panoramic representation against several pinhole-based view selection strategies, all using a 120° horizontal FoV and a fixed camera center as in Sec. 3.1. (1) **Fixed rotation (4 / 16 views):** cameras are rotated at fixed horizontal angular intervals to uniformly cover 360°. (2) **Semantic-based view selection (4 views):** following SeeGround [22], instance masks and predicted labels of Mask3D [33] are used to find text-mentioned objects, up to four of which are chosen by projected size on the panorama, and the camera is oriented toward each; if fewer than four objects are mentioned, the remaining directions are sampled randomly. (3) **GT target (1 view, oracle):** a pinhole camera that always points directly at the ground-truth target.

As shown in Tab. 6, panoramic views substantially outperform all non-oracle pinhole baselines (41.3–51.0 vs. 61.0 overall), indicating that our panoramic representation captures scene context more effectively. The gap further widens as the number of referenced objects increases: in the "4+ objects" category, our approach (71.0) even surpasses the pinhole oracle (66.0). This suggests that pinhole views are fundamentally limited by their narrow field of view, which observes only a subset of objects at once, whereas panoramic views preserve the global spatial relations of the scene.

5 Conclusion

The proposed 3DVG method, PanoGrounder, has demonstrated that panoramic renderings are an effective intermediate representation for 3D visual grounding, preserving long-range spatial relations within a single view while enabling direct use of powerful 2D VLMs. Extensive experiments also have shown state-of-the-art results on ScanRefer and Nr3D, and strong robustness to unseen scenes and text rephrasings, indicating that panorama-driven VLM grounding is a practical path toward generalizable 3D visual grounding.

Limitations and Future Work. PanoGrounder requires an off-the-shelf 3D reconstruction as a preprocessing step, and severe reconstruction noise and artifacts can degrade 2D inference. It is also slower than fully feed-forward 3D methods due to VLM processing, though this can be mitigated by using smaller VLM backbones. Looking forward, three promising directions are: (i) handling queries with *no target object* or *multiple targets*; (ii) generalizing PanoGrounder to *multi-task* settings such as 3D captioning and 3D VQA; and (iii) extending it to building-level or outdoor settings via more flexible camera placement beyond the floor-based grid.

Acknowledgments

We thank Chunghyun Park for his helpful comments and technical advice during the development of the model. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)] and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) [No. RS-2024-00359718].

References

1. Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In: European Conference on Computer Vision. pp. 422–440 (2020)
2. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3674–3683 (2018)
3. Baruch, G., Chen, Z., Dehghan, A., Dimry, T., Feigin, Y., Fu, P., Gebauer, T., Joffe, B., Kurz, D., Schwartz, A., Shulman, E.: ARKitScenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1) (2021)
4. Chang, C.P., Wang, S., Pagani, A., Stricker, D.: Mikasa: Multi-key-anchor & scene-aware transformer for 3d visual grounding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14131–14140 (2024)
5. Chen, D.Z., Chang, A.X., Nießner, M.: Scanrefer: 3d object localization in rgb-d scans using natural language. In: European Conference on Computer Vision. pp. 202–221 (2020)
6. Chen, S., Guhur, P.L., Tapaswi, M., Schmid, C., Laptev, I.: Language conditioned spatial relation reasoning for 3d object grounding. Advances in Neural Information Processing Systems **35**, 20522–20535 (2022)
7. Cheng, A.C., Yin, H., Fu, Y., Guo, Q., Yang, R., Kautz, J., Wang, X., Liu, S.: Spatialrgpt: Grounded spatial reasoning in vision-language models. Advances in Neural Information Processing Systems **37**, 135062–135093 (2024)

8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5828–5839 (2017)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). pp. 4171–4186 (2019)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
11. Fu, R., Liu, J., Chen, X., Nie, Y., Xiong, W.: Scene-llm: Extending language model for 3d visual reasoning. In: IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2195–2206 (2025)
12. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
13. Huang, H., Chen, Y., Wang, Z., Huang, R., Xu, R., Wang, T., Liu, L., Cheng, X., Zhao, Y., Pang, J., et al.: Chat-scene: Bridging 3d scene and large language models with object identifiers. Advances in Neural Information Processing Systems **37**, 113991–114017 (2024)
14. Huang, R., Yang, H., Cai, Y., Xu, X., Zhang, H., He, S.: Viewsrd: 3d visual grounding via structured multi-view decomposition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9726–9736 (2025)
15. Huang, T., Zhang, Z., Tang, H.: 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. arXiv preprint arXiv:2507.23478 (2025)
16. Jain, A., Gkanatsios, N., Mediratta, I., Fragkiadaki, K.: Bottom up top down detection transformers for language grounding in images and point clouds. In: European Conference on Computer Vision. pp. 417–433 (2022)
17. Jain, A., Swerdlow, A., Wang, Y., Arnaud, S., Martin, A., Sax, A., Meier, F., Fragkiadaki, K.: Unifying 2d and 3d vision-language understanding. In: Proceedings of the 42nd International Conference on Machine Learning. pp. 26717–26739 (2025)
18. Jia, B., Chen, Y., Yu, H., Wang, Y., Niu, X., Liu, T., Li, Q., Huang, S.: Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In: European Conference on Computer Vision. pp. 289–310 (2024)
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139:1–139:14 (2023)
20. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026 (2023)
21. Lewis, M., Nayak, N., Yu, P., Merullo, J., Yu, Q., Bach, S., Pavlick, E.: Does clip bind concepts? probing compositionality in large image models. In: Findings of the Association for Computational Linguistics: EACL 2024. pp. 1487–1500 (2024)
22. Li, R., Li, S., Kong, L., Yang, X., Liang, J.: Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)
23. Li, X., Ding, J., Chen, Z., Elhoseiny, M.: Uni3dl: A unified model for 3d vision-language understanding. In: European Conference on Computer Vision. pp. 74–92 (2024)

24. Liu, D., Liu, Y., Huang, W., Hu, W.: A survey on text-guided 3-d visual grounding: Elements, recent advances, and future directions. IEEE Transactions on Neural Networks and Learning Systems **36**, 17717–17737 (2025)
25. Liu, H., Lin, A., Han, X., Yang, L., Yu, Y., Cui, S.: Refer-it-in-rgbd: A bottom-up approach for 3d visual grounding in rgbd images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6032–6041 (2021)
26. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in Neural Information Processing Systems **36**, 34892–34916 (2023)
27. Miyanishi, T., Azuma, D., Kurita, S., Kawanabe, M.: Cross3dvg: Cross-dataset 3d visual grounding on different rgb-d scans. In: International Conference on 3D Vision. pp. 717–727 (2024)
28. Niven, T., Kao, H.Y.: Probing neural network comprehension of natural language arguments. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4658–4664 (2019)
29. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2024)
30. Qian, Z., Ma, Y., Lin, Z., Ji, J., Zheng, X., Sun, X., Ji, R.: Multi-branch collaborative learning network for 3d visual grounding. In: European Conference on Computer Vision. pp. 381–398 (2024)
31. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763 (2021)
32. Ren, S., Wu, Z., Zhu, K.Q.: Emo: Earth mover distance optimization for auto-regressive language modeling. In: International Conference on Learning Representations (2024)
33. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., Leibe, B.: Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. In: IEEE International Conference on Robotics and Automation. pp. 8216–8223 (2023)
34. Wald, J., Avetisyan, A., Navab, N., Tombari, F., Nießner, M.: Rio: 3d object instance re-localization in changing indoor environments. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7658–7667 (2019)
35. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., XiXuan, S., et al.: Cogvlm: Visual expert for pretrained language models. Advances in Neural Information Processing Systems **37**, 121475–121499 (2024)
36. Wang, Y., Li, Y.L., Wu, E.Z.Y., Wang, S.: Liba: Language instructed multi-granularity bridge assistant for 3d visual grounding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 8114–8122 (2025)
37. Wu, Y., Cheng, X., Zhang, R., Cheng, Z., Zhang, J.: Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19231–19242 (2023)
38. Xu, R., Huang, Z., Wang, T., Chen, Y., Pang, J., Lin, D.: Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. In: CoRL (2024)
39. Yang, J., Chen, X., Qian, S., Madaan, N., Iyengar, M., Fouhey, D.F., Chai, J.: Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. In: IEEE International Conference on Robotics and Automation. pp. 7694–7701 (2024)

40. Yuan, Z., Ren, J., Feng, C.M., Zhao, H., Cui, S., Li, Z.: Visual programming for zero-shot open-vocabulary 3d visual grounding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20623–20633 (2024)
41. Yuksekgonul, M., Bianchi, F., Kalluri, P., Jurafsky, D., Zou, J.: When and why vision-language models behave like bags-of-words, and what to do about it? In: International Conference on Learning Representations (2023)
42. Zhan, Y., Yuan, Y., Xiong, Z.: Mono3dvg: 3d visual grounding in monocular images. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 6988–6996 (2024)
43. Zhang, H., Li, L.H., Meng, T., Chang, K.W., Van den Broeck, G.: On the paradox of learning to reason from data. In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence. pp. 3365–3373 (2023)
44. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)
45. Zhu, C., Wang, T., Zhang, W., Pang, J., Liu, X.: Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4295–4305 (2025)
46. Zhu, Y., Zhang, J., Wang, Y., Wu, A., Deng, C.: Vgmamba: Attribute-to-location clue reasoning for quantity-agnostic 3d visual grounding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5295–5304 (2025)
47. Zhu, Z., Ma, X., Chen, Y., Deng, Z., Huang, S., Li, Q.: 3d-vista: Pre-trained transformer for 3d vision and text alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2911–2921 (2023)
48. Zhu, Z., Zhang, Z., Ma, X., Niu, X., Chen, Y., Jia, B., Deng, Z., Huang, S., Li, Q.: Unifying 3d vision-language understanding via promptable queries. In: European Conference on Computer Vision. pp. 188–206 (2024)