arXiv:2606.31148v3 [cs.CV] 16 Oct 2026

PruneGround: Plug-and-play Spatial Pruning for 3D Visual Grounding

Duc Cd [1] **Khai Le-Duc** [1,2,3] **Florent Draye** [4] **Chris Ngo** [1]
Terry Jingchen Zhang [3,6] **Bernhard Schölkopf** [4,5] **Zhijing Jin** [2,3,4]
[1] Knovel Engineering Lab [2] University of Toronto [3] Vector Institute
[4] MPI for Intelligent Systems, Tübingen [5] ELLIS Institute Tübingen
[6] University of Oxford
Co-first authors
duccd@hanyang.ac.kr duckhai.le@mail.utoronto.ca
leduckhai/PruneGround

Abstract

3D Visual Grounding (3DVG) aims to localize target objects in 3D scenes given natural language descriptions. Existing approaches typically perform reasoning over the entire scene, leading to ambiguous predictions and high computational cost, especially in cluttered environments. We observe that many referential expressions rely on local spatial context and often correspond to restricted spatial regions rather than the full scene. Motivated by this insight, we propose PruneGround, an effective plug-and-play framework for 3DVG built upon three key components. First, we introduce *Language-Guided Spatial Pruning* (LGSP), which leverages a frozen Vision Language Model (VLM) to identify language-relevant regions, thereby reducing spatial computation and grounding candidates in the narrower search space. Second, we propose *MultiView-Conditioned Description Reformulation* (MCDR), which decomposes complex expressions into simplified target–anchor relations and augments missing spatial cues through multi-view reasoning. Finally, we propose *LLM-Grounder*, which repurposes a detection-pretrained spatial LLM into a language-conditioned grounding model by aligning point cloud and linguistic representations within the pruned region. Extensive experiments on the three most popular point cloud benchmarks demonstrate that our method achieves **state-of-the-art** results on all three ScanRefer settings and on 9 out of 10 Nr3D/Sr3D settings.



Figure 1: **Intuition of our proposed method, inspired by human spatial cognition** (see Appendix Sec. A). Given a language description, the grounding model, as humans do, should infer the target object by attending *only* to spatially relevant regions guided by anchor objects and spatial relations, rather than reasoning over *all* candidate objects in the scene (e.g., all chairs).

1 Introduction

3D Visual Grounding (3DVG) focuses on linking natural language expressions to their corresponding objects in complex 3D environments, demanding robust language understanding and spatial reasoning, and serving as a key component in applications such as augmented reality (Chen et al., 2020; Ma et al., 2023), embodied navigation (Zhou et al., 2023; Chen et al., 2022b; Huang et al., 2022b), intelligent agents (Calisto et al., 2023; Zhang et al., 2023), robotic perception (Chen et al., 2023; Kong et al., 2023a,b; Lai et al., 2023), and autonomous driving (Cui et al., 2024; Feng et al., 2023). In this task, most existing methods adopt a *segment-then-select* paradigm (Yuan et al., 2021; Mane et al., 2025; Zhang et al., 2024a; Xu et al., 2024a; Chang et al., 2024), where 3D object proposals are first extracted via scene-level segmentation and then matched to the textual query. Despite strong benchmark performance, such *two-stage* approaches suffer from inherent limitations, including *error propagation* from imperfect segmentation and high computational cost, often redundant in cluttered scenes with fine-grained relational cues.

Given a language description for localizing a target object, the contextual cues typically correspond

to only a restricted region of the 3D scene. As illustrated in Fig. 1, instead of reasoning over all candidate objects (e.g., all chairs in the scene), the grounding model should focus *only* on spatially relevant areas guided by the anchor objects and spatial relations described in the utterance. This mirrors human spatial cognition, where people interpret referring expressions by using salient anchor objects and relational cues to narrow attention to the most relevant region of a scene rather than exhaustively comparing every possible object.

Additionally, human utterances are often free-form and exhibit ambiguity, incompleteness, and implicit spatial reasoning. Spatial relations may be partially specified or omitted, relying on assumptions that are intuitive for humans but challenging for models. This variability complicates grounding, as models must resolve ambiguity and infer missing geometric cues.

Consider the red chair in Fig. 1. The utterance may rely on the door to identify the target, yet in point clouds the door can be ambiguous, e.g. sharing color with the wall or partially missing due to occlusion and capture artifacts. Models that depend on the door as an anchor are thus prone to confusion. To improve localization, additional contextual cues are needed. Relations with nearby objects, such as a desk or bookshelf, provide complementary constraints that help disambiguate candidates and yield more stable bounding box estimation in cluttered or imperfect scenes.

Motivated by these problems, we propose PruneGround, a novel plug-and-play, *prune-then-ground* framework that leverages a Vision Language Model (VLM) to prune irrelevant regions of a 3D scene, reformulate descriptions, and optimize a Large Language Model (LLM) for effective grounding. By focusing computation on language-relevant regions, our framework enables a target-centric and more accurate grounding process.

We devise a *three-stage* pipeline for effective 3DVG:

(i) **Language-Guided Spatial Pruning (LGSP)**: We first render global multi-view RGB and depth images of the 3D scene and leverage a VLM to identify regions that are semantically relevant to the input query. This step effectively narrows the spatial search space and reduces unnecessary computation over irrelevant areas (Sec. 3.2).

(ii) **Multi-View-Conditioned Description Reformulation (MCDR)**: We further render side views and cropped top-view regions centered on the pruned areas, and employ the VLM to decompose complex multi-anchor expressions into structured single-anchor statements. In addition, we augment the language descriptions with explicit spatial relations between the target and surrounding anchors that may be implicit or absent in the original utterance (Sec. 3.3).

(iii) **Fine-tuned LLM-based Grounder (LLM-Grounder)**: Finally, we repurpose a detection-pretrained spatial LLM for language-conditioned grounding by aligning point cloud and linguistic representations within the pruned region and adapting the decoding scheme to produce the target prediction in a structured language form (Sec. 3.4).

By simply integrating VLM-based scene pruning and description reformulation with an effective LLM-based grounding framework, our plug-and-play PruneGround achieves **state-of-the-art** performance across the three most popular 3DVG benchmarks - ScanRefer (Chen et al., 2020), Nr3D, and Sr3D from ReferIt3D (Achlioptas et al., 2020). Notably, it demonstrates strong generalization by delivering state-of-the-art results on all three ScanRefer settings (Tab. 1) and on 9 out of 10 Nr3D/Sr3D settings (Tab. 2).

2 Related Works

Two-stage vs. One-stage 3DVG. Pioneering ScanNet-based benchmarks such as ScanRefer (Chen et al., 2020) and ReferIt3D (Achlioptas et al., 2020; Dai et al., 2017) have advanced 3DVG. Early supervised methods typically use separate language and scene encoders followed by cross-modal fusion.

(i) *Two-stage* approaches (Zhao et al., 2021; Achlioptas et al., 2020; Bakr et al., 2022; Chen et al., 2022a; Yang et al., 2021; Yuan et al., 2021; Zhu et al., 2023; Mane et al., 2025; Wang et al., 2025a; Zhang et al., 2024b; Chang et al., 2024; Yang et al., 2023; Abdelrahman et al., 2024; Zhang et al., 2024a) follow a *proposal-and-selection* paradigm: a 3D detector or segmentor first generates candidate proposals, then the proposal best matching the query is selected. By decoupling object localization and language grounding, they leverage strong geometric priors and mature detection pipelines through explicit candidate reasoning.

(ii) In contrast, *one-stage* approaches (Jain et al., 2022, 2025; Luo et al., 2022; Qian et al., 2024; Wang et al., 2024; Wu et al., 2023b; Guo et al., 2025; Yuan et al., 2024; Wu et al., 2023a) directly

localize the target in the 3D scene conditioned on the query. This enables *end-to-end* optimization and tighter language-geometry interaction, while avoiding error propagation from explicit proposal generation.

LLM- and VLM-based 3DVG. To enhance language understanding in 3DVG, recent works incorporate LLMs (Guo et al., 2023; Yin et al., 2023; Zhan et al., 2023; Yuan et al., 2024; Feng et al., 2024; Zhu et al., 2024a; Mi et al., 2025; Liu et al., 2025) and VLMs (Xu et al., 2024b; Wang et al., 2025b; Cao et al., 2025; Li et al., 2025b; Zhu et al., 2025a; Jung et al., 2025; Jin et al., 2025) into the grounding pipelines.

(i) *Zero-shot* approaches exploit pretrained VLMs for open-vocabulary 3DVG without task-specific 3D supervision. ZSVG3D (Yuan et al., 2024) uses LLM-driven visual programming and modular 3D reasoning; VLM-Grounder (Xu et al., 2024b) performs iterative grounding over stitched multi-view images and projects ensembles to 3D boxes; and SeeGround (Li et al., 2025a) converts 3D scenes into query-aligned rendered views with spatial textual descriptions to match VLM inputs.

(ii) Fine-tuned approaches embed VLMs/LLMs into 3D grounding networks with task-specific box decoders. LIBA (Wang et al., 2025b) mitigates point-language granularity mismatch via multi-granularity alignment, dynamic cross-modal adapters, and LLM-guided hierarchical reasoning. ReGround3D (Zhu et al., 2024a) introduces a reasoning-oriented 3D grounding benchmark and a VLM-based chain-of-grounding framework that interleaves reasoning and localization.

Structured language reasoning for 3DVG. Accurate 3DVG requires reasoning over attributes, relations, and viewpoint-dependent expressions. Recent works improve language grounding by converting free-form utterances into structured semantic forms. ViewRefer (Guo et al., 2023) uses LLMs to enrich queries with view-aware templates, while EDA (Wu et al., 2023b) decomposes descriptions into fine-grained semantic elements for dense alignment with point cloud instances. ViewSRD (Huang et al., 2025) restructures multi-anchor expressions into simpler single-anchor statements to disentangle target-anchor relations.

Similarly, we decompose multi-anchor utterances into single-anchor relations. Unlike purely text-based restructuring (Guo et al., 2023), our method renders multi-view images and uses a VLM to decouple relations, remove redundancies, and supplement missing geometric cues, yielding richer perspective-aware descriptions.

3 PruneGround

3.1 Overall Architecture

As shown in Fig. 2, given a 3D scene represented by point coordinates and color features $\mathcal{S} \in \mathbb{R}^{N\times 6}$ and a **natural language query** $\mathcal{Q}$[1], our goal is to localize the referred target object. Instead of grounding over the entire scene, we propose a *three-stage* framework that progressively reduces spatial redundancy and refines language understanding.

(i) We introduce **LGSP** (Sec. 3.2), where multi-view renderings (top and oblique views) are aligned with the query using a pretrained VLM to identify semantically relevant regions. Irrelevant 3D points are removed, yielding a pruned scene that reduces clutter and narrows the search space.

(ii) We present **MCDR** (Sec. 3.3). Using top and side views of the pruned scene, we identify target and anchor objects and reformulate complex expressions into structured *single-target→single-anchor* relations. Redundant statements are filtered, and the query is enriched with geometric cues inferred from multi-view semantics.

(iii) Given the pruned scene and reformulated description, we develop **LLM-Grounder** (Sec. 3.4), a language-conditioned grounding model repurposed from a detection-pretrained spatial LLM, which jointly reasons over language and point cloud representations to predict the target object.

3.2 Language-Guided Spatial Pruning (LGSP)

Let an indoor 3D scene represented as a point cloud with coordinates and color, $\mathcal{S} \in \mathbb{R}^{N\times 6}$, where the x-y plane aligns with the floor and the z-axis points upward, opposite gravity. This canonical alignment enables consistent multi-view rendering and spatial reasoning.

Global Multi-View Rendering. The **LGSP** module first renders a set of structured multi-view observations from $\mathcal{S}$. Specifically, we generate one top-view image that serves as a canonical spatial layout of the scene, together with four oblique views. The top view captures the global spatial arrangement along the horizontal plane but may lack

[1]The colors of the *variables* and *components* in Sec. 3 correspond to their colors in Fig. 2

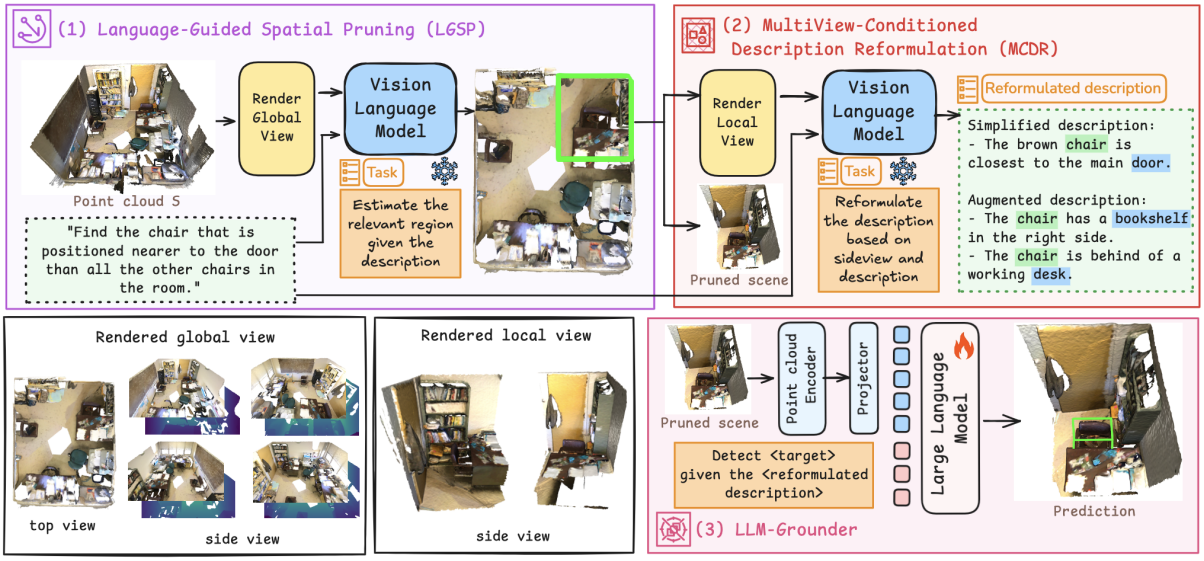


Figure 2: **Overview of our PruneGround pipeline**, consisting of three *plug-and-play* components: ⓘ LGSP to remove irrelevant regions (Sec. 3.2), ⓘⓘ MCDR to simplify the description while adding relational spatial information (Sec. 3.3), and ⓘⓘⓘ LLM-Grounder that repurposes a detection-pretrained spatial LLM for language-conditioned grounding on the pruned scene and reformulated description (Sec. 3.4).

semantic details of vertically structured objects. To complement this, each oblique view provides an RGB image encoding object semantics and a corresponding depth map encoding geometric structure.

To construct the oblique views, we compute the axis-aligned bounding box (AABB) of the scene to define its spatial extent. The camera centers are placed at the midpoints of the four upper edges of the AABB. Each camera is oriented toward the center of the opposite vertical face, ensuring that the field of view covers the corresponding side of the scene (see Appendix Fig. 4). RGB and depth images are rendered simultaneously under this configuration. Implementation details of the camera intrinsics and rendering settings are provided in the Appendix Sec. B. The final multi-view representation thus consists of one top-view image, four oblique RGB images, and four corresponding oblique depth maps, as shown in Fig. 2.

Language-Guided Region Selection. The rendered multi-view images, together with the **language query** $\mathcal{Q}$, are fed into a **frozen VLM**. Conditioned on the textual query, the VLM analyzes the semantic and spatial correspondences across views and predicts a relevant **2D bounding region** on the top-view image that is most consistent with the description.

$$\begin{aligned} \text{bbox}^{2D} &= (x_{\min}, x_{\max}, y_{\min}, y_{\max}) \\ &= \text{VLM}(\mathcal{V}_{\text{global}}, \mathcal{Q}) \end{aligned} \tag{1}$$

where $\mathcal{V}_{\text{global}}$ denotes the *global multi-view* input, and $(x_{\min}, x_{\max}, y_{\min}, y_{\max})$ define the boundaries of the retained region.

Details of the structured prompting for the VLM are provided in Fig. 6 of Appendix Sec. C.

Post-processing and 3D Projection. To ensure spatial stability, we apply post-processing to the predicted **2D bounding box** in the top view by enforcing a *minimum area constraint*. If the box falls below this threshold, it is proportionally expanded to maintain sufficient coverage. The refined box is then projected back into 3D space. Since the top view lies on the x-y plane (with a default resolution of 512×512), we retain the full extent along the z-axis and select points whose (x, y) coordinates lie within the box, discarding the rest to obtain the pruned point cloud S'.

For compact scenes where the area or pruned region is smaller than $1\,\text{m} \times 1\,\text{m}$, we retain the entire scene or expand the region accordingly to avoid unnecessary information loss.

3.3 Multi-View-Conditioned Description Reformulation (MCDR)

Natural language descriptions in 3DVG are inherently human-centric and often reflect intuitive reasoning rather than complete spatial specifications. As a result, relevant objects or spatial relations may be omitted or only implicitly expressed. While ViewSRD (Huang et al., 2025) decomposes ex-

pressions into target-anchor pairs to facilitate relational learning, it is purely text-driven and does not condition on visual semantics. Consequently, it may miss spatial cues that are visually evident but linguistically absent. To address this, we propose an **MCDR** module that restructures descriptions into target-anchor relations under multiview visual conditioning. By incorporating semantic cues from top and side views of the pruned region, our method simplifies queries into *single-target→single-anchor* statements while recovering missing spatial information from scene geometry.

Local Multi-View Rendering. For the pruned scene $\mathcal{S}'$ represented by a *3D bounding box*, we render two side-view RGB images from the x- and y-axis directions of the *selected top-view image's bounding box*. For the camera aligned with the x-axis, the viewing direction is set to pass through the centers of the two faces parallel to the y-axis. The camera is then gradually elevated to reduce occlusions from surrounding furniture. To further mitigate occlusions from structural elements (e.g., walls), the camera orientation is constrained to face outward from the center of the original scene. The camera along the y-axis is configured using the same procedure. In this step, depth maps are discarded to simplify the input and emphasize semantic content. A visualization of the camera configuration and rendering pipeline is provided in Fig. 5 of Appendix Sec. B.

Description Simplification and Augmentation. Given rendered local multi-view images $\mathcal{V}_{\text{local}}$ and the **original description** $\mathcal{Q}$, we continue to leverage a **frozen VLM** to perform *structured* query reformulation:

$$(t, \mathcal{T}, \mathcal{A}, \hat{\mathcal{S}}) = \text{VLM}(\mathcal{V}_{\text{local}}, \mathcal{Q}). \tag{2}$$

The **VLM** decomposes the **query** $\mathcal{Q}$ into four components: (i) the target object name t, (ii) a set of attributes of the target object $\mathcal{T}$, which describe properties such as color, material, and shape, (iii) a set of anchor object names $\mathcal{A}$, and (iv) a set of target–anchor relational statements $\hat{\mathcal{S}}$, each expressed as: $\langle$target, target attributes$\rangle$ + $\langle$spatial relation$\rangle$ + $\langle$anchor, anchor attributes$\rangle$.

By prompting the **VLM** to augment relational information using the local multi-view images, the resulting relational statements incorporate not only reformulated content from the **original description** but also additional target–anchor pairs. Specifically, anchors that do not explicitly appear in the original description but are observable in the local multi-view images may be introduced, leading to enriched relational supervision grounded in visual context. Details of the structured prompting strategy for the VLM query are provided in Fig. 7 of Appendix Sec. C.

3.4 LLM-based Grounder (LLM-Grounder)

Given a reformulated referring expression and a pruned 3D point cloud as input, we build upon the SpatialLM framework (Mao et al., 2025), originally developed for unconditional 3D object detection, and repurpose it for language-conditioned grounding. The architecture follows an *Encoder–Projector–LLM* design.

A point cloud encoder (Sonata (Wu et al., 2025), a PTv3 variant with 108.8M parameters (Wu et al., 2024)) first extracts geometric features from the pruned scene. Unlike the original SpatialLM, which processes point clouds without linguistic conditioning, we inject the referring expression into the multimodal sequence to enable target-aware reasoning. Specifically, the geometric features are projected via a Multilayer Perceptron (MLP) into the language embedding space and concatenated with the text token embeddings of the referring expression, forming a unified multimodal input sequence. The LLM (Qwen2.5-0.5B (Yang et al., 2024)) processes this sequence and predicts a single target object matching the referring expression, in a structured language format:

$$\texttt{<}c,\ p_x, p_y, p_z,\ s_x, s_y, s_z\texttt{>} \tag{3}$$

where c denotes the object name, (p_x, p_y, p_z) the center position, and (s_x, s_y, s_z) the size of the bounding box along each axis. This contrasts with the enumerative output of the original SpatialLM, which produces structured tokens for all detected objects in the scene. Finally, the predicted bounding box parameters are mapped back to the original scene coordinate space to obtain the final grounding result.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our framework on three widely used 3DVG benchmarks. (i) **ScanRefer** contains 51,583 human-written referring expressions for 800 ScanNet scenes (Dai et al., 2017). Following the official split, we use 36,665 samples for training and 9,508 for validation. Based

on object uniqueness within a scene, the dataset is divided into *unique* (the target class appears once) and *multiple* (the target class appears multiple times) subsets. (ii) **Nr3D** consists of 37,842 human-written referring expressions grounded to annotated objects across 641 ScanNet scenes (511 for training and 130 for validation), covering 76 object classes. Each expression refers to a target object among same-class distractors. For evaluation, the dataset is partitioned into *easy* (one distractor) and *hard* (multiple distractors) subsets. Expressions are further categorized as *view-dependent* or *view-independent*, depending on whether grounding requires a specific viewpoint. (iii) **Sr3D** contains 83,570 automatically generated referring expressions from 1,018 training and 255 validation ScanNet scenes. Unlike Nr3D, Sr3D uses predefined templates and relies purely on spatial relations to distinguish same-class objects. We follow the standard evaluation protocol consistent with Nr3D.

Implementation Details. We use *Qwen2.5-VL-3B-Instruct* (Bai et al., 2025) as the frozen VLM for both LGSP and MCDR. The top-view, oblique-view, and side-view images are rendered at a resolution of 512×512 pixels with a field of view (FoV) of 90° (see Fig. 4 and Fig. 5 of Appendix Sec. B). LLM-Grounder adopts the Qwen-based SpatialLM (Mao et al., 2025) (Qwen2.5-0.5B) and initialize from its pre-trained checkpoint. Following Mao et al. (2025), we shift the point cloud and target bounding box coordinates to ensure non-negative values, then quantize them into 1,280 bins at 2.5cm resolution. The LLM predicts integer values, which are mapped back to continuous coordinates via inverse transformation. All experiments are implemented in PyTorch and run on an NVIDIA A100 80 GB GPU. Further details of the pre-trained model initialization and training procedure are provided in Appendix Sec. E.

4.2 Experimental Results

ScanRefer dataset. Tab. 1 compares our method with prior approaches on the ScanRefer benchmark. On the challenging *Multiple* subset, where multiple candidates share the same class as the target, our pruning strategy effectively removes objects in irrelevant regions, achieving 57.6% Acc@0.25 and 49.4% Acc@0.5. As a result, our method achieves 63.8% Acc@0.25 and 56.1% Acc@0.5 overall, establishing a new **state of the art in all three settings**: *Unique*, *Multiple*, *Overall*.

Nr3D and Sr3D datasets. As shown in Tab. 2, on the challenging *Hard* subset - where the target object is distracted by other objects of the same class - our approach achieves 68.8% and 76.5% on Nr3D and Sr3D respectively. As a result, we establish a new **state of the art in 9 out of 10 settings**, with *Overall* accuracies of 75.1% and 81.5%, respectively.

Qualitative analysis. As shown in the qualitative examples (Fig. 3), LGSP effectively prunes irrelevant regions, reducing candidate objects and directing attention to language-relevant areas. Meanwhile, MCDR rewrites descriptions into a simpler, more structured form, enriching them with additional attributes and spatial relations-even beyond the original input-while maintaining consistent length and semantic density. Together, these components enable LLM-Grounder to better align language with the target object for accurate bounding box prediction without being distracted by irrelevant candidates or missing spatial relationships in the original utterance.

4.3 Ablation Study

Impact of each proposal module. As shown in Tab. 3, incorporating LGSP consistently improves performance across both *Unique* and *Multiple* subsets by removing irrelevant regions, yielding a **+3.2%** Acc@0.5 gain in overall accuracy. Building on this, MCDR further boosts results through description reformulation: simplification alone provides noticeable improvements across both subsets (*Overall* **+1.7%** Acc@0.5), while combining simplification and augmentation delivers the best performance, bringing an additional **+2.1%** Acc@0.5 gain in overall.

Candidate Reduction Rate. Next, we conduct an ablation study to validate the effectiveness of the proposed LGSP module. We first evaluate it on the Sr3D Hard subset, where each query contains at least two distractor objects from the same category, providing a challenging setting for candidate reduction before grounding. A candidate is considered removed if the pruned scene contains no spatial region belonging to that object. As shown in Table 4, our module reduces grounding candidates from 2.43 to 1.28 on Sr3D Hard and from 5.72 to 1.62 on the ScanRefer Multiple split, effectively narrowing the grounding search space.

Method	Venue	Unique		Multiple		Overall	
		Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
ScanRefer (Chen et al., 2020)	ECCV	67.6	46.2	32.1	21.3	40.0	26.1
MVT (Huang et al., 2022a)	CVPR	77.7	66.5	31.9	25.3	40.8	33.3
3D-SPS (Luo et al., 2022)	CVPR	84.1	66.7	40.3	29.8	48.8	37.0
EDA (Wu et al., 2023b)	CVPR	85.8	68.6	49.1	37.6	54.6	42.3
ConcreteNet (Unal et al., 2023)	ECCV	86.4	82.1	42.4	38.4	50.6	46.5
Chat-Scene (Huang et al., 2023)	NeurIPS	89.6	82.5	47.8	42.9	55.5	50.2
VPP-Net (Shi et al., 2024)	CVPR	86.1	67.1	50.3	39.0	55.7	43.3
G^3-LQ (Wang et al., 2024)	CVPR	88.6	73.3	50.2	39.7	56.0	44.7
MA^2TransVG (Xu et al., 2024a)	CVPR	86.3	74.1	53.8	41.4	57.9	45.7
MCLN (Qian et al., 2024)	ECCV	86.9	72.7	52.0	40.8	57.2	45.5
PQ3D (Zhu et al., 2024b)	ECCV	86.7	78.3	51.5	46.2	57.0	51.2
VGMamba (Zhu et al., 2025b)	ICCV	**91.9**	79.6	54.8	47.6	60.0	53.9
PanoGrounder (Jung et al., 2025)	arXiv	84.3	–	55.3	–	61.0	–
PruneGround **(Ours)**		90.2	**84.7**	**57.6**	**49.4**	**63.8**	**56.1**

Table 1: Grounding accuracy (%) on **ScanRefer** (Acc@0.25 / 0.5, IoU). **Bold**: best, underline: second-best.

Method	Venue	Nr3D					Sr3D				
		Easy	Hard	View-D	View-I	Overall	Easy	Hard	View-D	View-I	Overall
ReferIt3D (Achlioptas et al., 2020)	ECCV	43.6	27.9	32.5	37.1	35.6	44.7	31.5	39.2	40.8	40.8
ViL3DRel (Chen et al., 2022a)	NeurIPS	70.2	57.4	62.0	64.5	64.4	74.9	67.9	63.8	73.2	72.8
3D-VisTA (Zhu et al., 2023)	ICCV	65.9	49.4	53.7	59.4	57.5	72.1	63.6	57.9	70.1	69.6
MIKASA (Chang et al., 2024)	CVPR	69.7	59.4	65.4	64.0	64.4	78.6	67.3	**70.4**	75.4	75.2
G^3-LQ (Wang et al., 2024)	CVPR	–	50.7	–	–	58.4	–	66.3	–	–	73.1
MA^2TransVG (Xu et al., 2024a)	CVPR	–	57.6	–	–	65.2	–	69.3	–	–	73.9
PQ3D (Zhu et al., 2024b)	ICCV	73.3	56.7	60.7	67.0	64.9	78.8	68.2	51.5	76.7	75.6
LIBA (Wang et al., 2025b)	AAAI	–	57.2	60.3	–	64.5	–	70.2	61.7	–	75.8
ViewSRD (Huang et al., 2025)	ICCV	75.3	64.8	68.6	70.6	69.9	78.3	70.6	69.0	76.2	76.0
VGMamba (Zhu et al., 2025b)	ICCV	–	61.4	–	–	68.3	–	74.4	–	–	81.3
PanoGrounder (Jung et al., 2025)	arXiv	82.2	67.2	70.5	76.3	74.6	81.3	74.2	60.5	80.0	79.1
PruneGround **(Ours)**		**84.6**	**68.8**	**71.2**	**76.6**	**75.1**	**83.7**	**76.5**	61.7	**82.4**	**81.5**

Table 2: Grounding accuracy (%) on **Nr3D** and **Sr3D** (Acc@0.5, IoU). **Bold**: best, underline: second-best

Method	Unique		Multiple		Overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
LLM-Grounder	83.1	75.3	51.2	42.9	57.3	49.1
+ LGSP	85.7 +2.6	79.9 +4.6	53.6 +2.4	45.8 +2.9	59.2 +1.9	52.3 +3.2
+ MCDR (S)	88.6 +2.9	82.5 +2.6	56.0 +2.4	47.3 +1.5	62.4 +3.2	54.0 +1.7
+ MCDR (S + A)	**90.2** +1.6	**84.7** +2.2	**57.6** +1.6	**49.4** +2.1	**63.8** +1.4	**56.1** +2.1

Table 3: **Impact of each proposal module on ScanRefer**. Baseline is LLM-Grounder trained on *original* descriptions and *unpruned* scenes, then incrementally add LGSP for scene pruning and MCDR for description reformulation (via simplification (S) or simplification + augmentation (S + A)). Consistent improvements are also seen on Nr3D and Sr3D (see Appendix Tab. 9).

Dataset	No pruning	With LGSP
ScanRefer Multiple Eval	5.72	1.62
Sr3D Hard set	2.43	1.28

Table 4: **Impact of LGSP on removing noisy candidates in cluttered environments.**. The average number of candidates is compared against no-pruning method.

Target recall of LGSP under varying numbers of oblique views. In Tab. 5, we evaluate how the number of oblique views affects relevant-region detection, where success is defined as covering >75% of the target. Performance consistently improves with more views and further benefits from depth. Notably, using two opposing oblique views with depth already yields strong gains, while four views achieve the best result of 94.7% (**+22.1%** over the 1-view RGB baseline). Based on this, we adopt four oblique views with depth as a good trade-off between accuracy and efficiency, with optional box expansion to increase target coverage in practice.

VLM selection justification While larger VLMs such as Qwen2.5-VL-72B, used in SeeGround (Li et al., 2025a), and frontier APIs such as ChatGPT-5 and Gemini 3 Pro offer stronger general-purpose reasoning, they incur substantial latency (>10s/sample) and cost, limiting large-scale evaluation. As shown in Tab. 6, we compare four VLMs

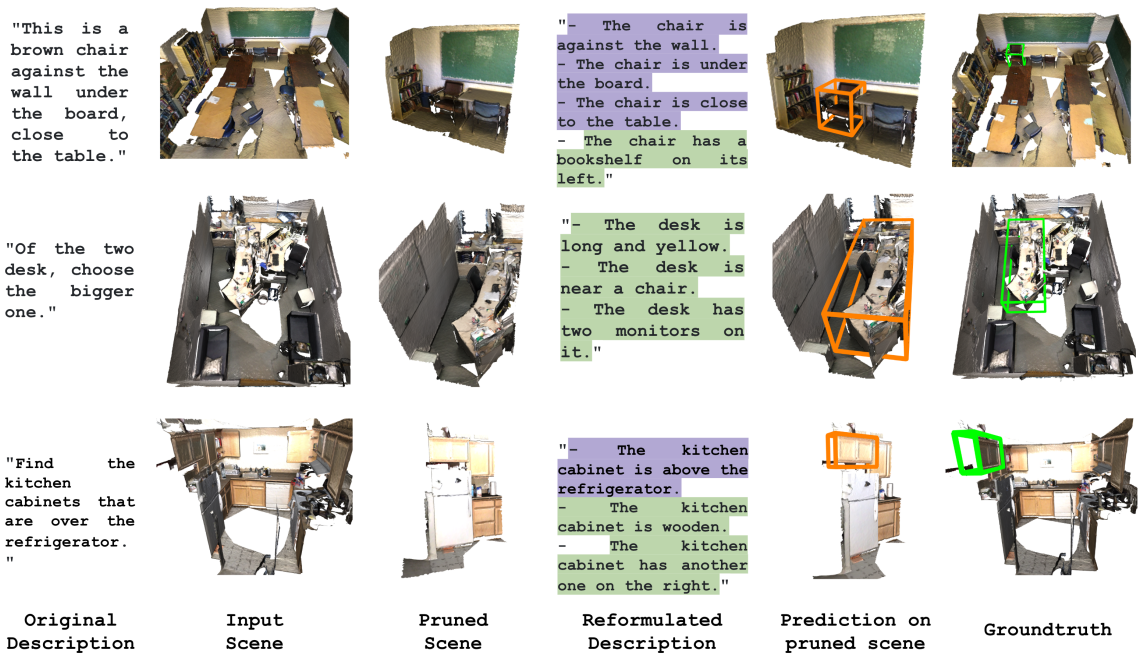


Figure 3: **Qualitative results on ScanRefer, Nr3D, and Sr3D.** The first column shows original utterances with varying complexity. The third column presents scenes pruned by LGSP. The fourth column shows reformulated descriptions from MCDR, where **purple** masks denote *simplified* text and **green** text indicates *augmented* descriptions. The fifth column shows **3D bounding boxes** predicted by LLM-Grounder. Visualization of each step is provided in Appendix Fig. 8.

# Oblique views	RGB	RGB + Depth
1	72.6	74.3 + 1.7
2	84.8 + 12.2	88.9 + 16.3
4	93.1 + 20.5	**94.7** + 22.1

Table 5: **Target recall of LGSP under varying numbers of oblique views** on the ScanRefer dataset. A sample is counted as successful when the pruned region retains >75% of the target, and target recall is reported as the percentage of successful samples. Improvements are measured against the 1-view RGB baseline (72.6)

Frozen VLMs	Area retained (%)	Target recall (%)	Run time (second)
Gemini 3 Pro	12.1	**95.7**	13.1
ChatGPT-5	11.7	95.3	11.2
Qwen2.5-VL-72B-Instruct	16.3	95.1	12.6
Qwen2.5-VL-3B-Instruct	15.1	94.7 - 1.0	**1.4** - 11.7

Table 6: **Analysis of different VLMs on the pruning task**, used in LGSP and MCDR. Box sizes are measured in pixels with respect to a 512×512 top-view image. Deviations are measured against the best results. The entire framework using Qwen2.5-VL-3B-Instruct needs 4.1s for inference (see Appendix Tab. 7).

with a *no-pruning* baseline, which retains the full 512×512 top-view image, across pruned area, target recall, and runtime. All four VLMs achieve similar recall (94.7–95.7%) while removing 83.7–88.3% of irrelevant regions, suggesting that the pruning task is largely saturated and gains little from frontier-scale reasoning. Qwen2.5-VL-3B-Instruct (Bai et al., 2025) runs in 1.4s per stage, about 8–9× faster than the alternatives, while remaining within 1% recall of the best model.

5 Conclusion

In this paper, we introduce PruneGround, the first framework to leverage spatial pruning and detection-pretrained spatial LLM repurposing for 3DVG. It integrates three key components: (i) LGSP to remove irrelevant regions using a frozen VLM with semantic and geometric cues, (ii) MCDR to reformulate the description into a simpler form while augmenting it with semantic and relational spatial information, and (iii) LLM-Grounder, an LLM-based grounding model that aligns point cloud features with linguistic representations for target-selective prediction. Our *plug-and-play* framework achieves **state-of-the-art** results on three most popular point cloud benchmarks (ScanRefer, Nr3D and Sr3D). Notably, it demonstrates strong generalization by delivering state-of-

the-art results on all three ScanRefer settings and on 9 out of 10 Nr3D/Sr3D settings.

Limitations

Our pipeline is limited by *error propagation* across its three stages. Errors in **LGSP** can produce incomplete or noisy candidates, while mistakes in language understanding in **MCDR** may misinterpret attributes or spatial relations. Since **LLM-Grounder** relies on these intermediate outputs, early-stage errors can accumulate and amplify, reducing robustness in cluttered scenes, ambiguous descriptions, and fine-grained spatial reasoning tasks (see Fig. 9 of Appendix Sec. G).

Furthermore, future work should focus on developing more lightweight and adaptable approaches to improve generalization across diverse real-world environments.

Acknowledgement

This work was supported in part by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B; by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645; by Coefficient Giving; by the SwissAI Initiative; by Schmidt Sciences; by the Frontier Model Forum and AI Safety Fund;by the Canadian AI Safety Institute Research Program at CIFAR; by the Canadian AI Safety Institute Research Program at CIFAR through a Catalyst Award; by the Survival and Flourishing Fund; by the Cooperative AI Foundation; and by the Division of Engineering Science at the University of Toronto. Resources used in preparing this research project were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. F.D. acknowledges support through a fellowship from the Hector Fellow Academy.

References

Eslam Abdelrahman, Mohamed Ayman Mohamed, Mahmoud Ahmed, and Mohamed Elhoseiny. 2024. Cot3dref: Chain-of-thoughts data-efficient 3d visual grounding. In *International Conference on Learning Representations*, volume 2024, pages 11871–11896.

Panos Achlioptas, Ahmed Abdelreheem, F. Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. 2020. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European Conference on Computer Vision*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923.

Eslam Bakr, Yasmeen Alsaedy, and Mohamed Elhoseiny. 2022. Look around and refer: 2d synthetic semantics knowledge distillation for 3d visual grounding. *Advances in neural information processing systems*, 35:37146–37158.

Francisco Maria Calisto, João Gabriel de Matos Fernandes, Margarida Morais, Carlos Santiago, João Maria Abrantes, Nuno Jardim Nunes, and Jacinto C. Nascimento. 2023. Assertiveness-based agent communication for a personalized medicine on medical imaging diagnosis. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*.

Ang Cao, Sergio Arnaud, Oleksandr Maksymets, Jianing Yang, Ayush Jain, Sriram Yenamandra, Ada Martin, Vincent-Pierre Berges, Paul Mcvay, Ruslan Partsey, Aravind Rajeswaran, Franziska Meier, Justin Johnson, Jeong Joon Park, and Alexander Sax. 2025. From thousands to billions: 3d visual language grounding via render-supervised distillation from 2d vlms. In *International Conference on Machine Learning*.

Laura A Carlson and Gordon D Logan. 2001. Using spatial terms to select an object. *Memory & Cognition*, 29(6):883–892.

Chun-Peng Chang, Shaoxiang Wang, Alain Pagani, and Didier Stricker. 2024. Mikasa: Multi-key-anchor & scene-aware transformer for 3d visual grounding. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14131–14140.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. 2020. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 202–221. Springer.

Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Y. Qiao, and Wenping Wang. 2023. Clip2scene: Towards label-efficient 3d scene understanding by clip. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7020–7030.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022a. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in neural information processing systems*, 35:20522–20535.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022b. Think global, act local: Dual-scale graph transformer for

vision-and-language navigation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16516–16526.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. 2024. Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 902–909.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11.

Chun Feng, Joy Hsu, Weiyu Liu, and Jiajun Wu. 2024. Naturally supervised 3d visual grounding with language-regularized concept learners. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shuo Feng, Haowei Sun, Xintao Yan, Hao Cheng Zhu, Zhengxia Zou, Shengyin Shen, and Henry X. Liu. 2023. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615:620–627.

Wenxuan Guo, Xiuwei Xu, Ziwei Wang, Jianjiang Feng, Jie Zhou, and Jiwen Lu. 2025. Text-guided sparse voxel pruning for efficient 3d visual grounding. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3666–3675.

Ziyu Guo, Yiwen Tang, Renrui Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. 2023. Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15326–15337.

Haifeng Huang, Zehan Wang, Rongjie Huang, Luping Liu, Xize Cheng, Yang Zhao, Tao Jin, and Zhou Zhao. 2023. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems 37*.

Ronggang Huang, Haoxin Yang, Yan Cai, Xuemiao Xu, Huaidong Zhang, and Shengfeng He. 2025. Viewsrd: 3d visual grounding via structured multi-view decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. 2022a. Multi-view transformer for 3d visual grounding. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15503–15512.

Zanming Huang, Zhongkai Shangguan, Jimuyang Zhang, Gilad Bar, Matthew Boyd, and Eshed Ohn-Bar. 2022b. Assister: Assistive navigation via conditional instruction generation. In *European Conference on Computer Vision*.

Falk Huettig, Joost Rommers, and Antje S Meyer. 2011. Using the visual world paradigm to study language processing: A review and critical evaluation. *Acta psychologica*, 137(2):151–171.

Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. 2022. Bottom up top down detection transformers for language grounding in images and point clouds. In *European Conference on Computer Vision*, pages 417–433. Springer.

Ayush Jain, Alexander Swerdlow, Yuzhou Wang, Sergio Arnaud, Ada Martin, Alexander Sax, Franziska Meier, and Katerina Fragkiadaki. 2025. Unifying 2d and 3d vision-language understanding. In *International Conference on Machine Learning*, pages 26717–26739. PMLR.

Zhao Jin, Rong-Cheng Tu, Jingyi Liao, Wenhao Sun, Xiao Luo, Shunyu Liu, and Dacheng Tao. 2025. Spazer: Spatial-semantic progressive reasoning agent for zero-shot 3d visual grounding. In *Advances in Neural Information Processing Systems*.

Seongmin Jung, Seongho Choi, Gunwoo Jeon, Minsu Cho, and Jongwoo Lim. 2025. Panogrounder: Bridging 2d and 3d with panoramic scene representations for vlm-based 3d visual grounding. *ArXiv*, abs/2512.20907.

Lingdong Kong, You-Chen Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Y. Qiao, and Ziwei Liu. 2023a. Rethinking range view representation for lidar segmentation. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 228–240.

Lingdong Kong, You-Chen Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kaili Chen, and Ziwei Liu. 2023b. Robo3d: Towards robust and reliable 3d perception against corruptions. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19937–19949.

Lei Lai, Zhongkai Shangguan, Jimuyang Zhang, and Eshed Ohn-Bar. 2023. Xvo: Generalized visual odometry via cross-modal self-training. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10060–10071.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. 2025a. 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 (CVPR)*.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. 2025b. Zero-shot 3d visual grounding from vision-language models. *ArXiv*, abs/2505.22429.

Zhenyang Liu, Yikai Wang, Sixiao Zheng, Tongying Pan, Longfei Liang, Yanwei Fu, and Xiangyang Xue. 2025. Reasongrounder: Lvlm-guided hierarchical feature splatting for open-vocabulary 3d visual grounding and reasoning. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3718–3727.

Gordon D Logan. 1994. Spatial attention and the apprehension of spatial relations. *Journal of Experimental Psychology: Human Perception and Performance*, 20(5):1015.

Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 2022. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16433–16442.

Teli Ma, Rong Li, and Junwei Liang. 2023. An examination of the compositionality of large generative vision-language models. In *North American Chapter of the Association for Computational Linguistics*.

Atharv Mahesh Mane, Dulanga Weerakoon, Vigneshwaran Subbaraju, Sougata Sen, Sanjay E. Sarma, and Archan Misra. 2025. Ges3vig: Incorporating pointing gestures into language-based 3d visual grounding for embodied reference understanding. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9017–9026.

Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. 2025. Spatiallm: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*.

Boyu Mi, Hanqing Wang, Tai Wang, Yilun Chen, and Jiangmiao Pang. 2025. Language-to-space programming for training-free 3d visual grounding. In *Conference on Empirical Methods in Natural Language Processing*.

Zhipeng Qian, Yiwei Ma, Zhekai Lin, Jiayi Ji, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. 2024. Multi-branch collaborative learning network for 3d visual grounding. In *European Conference on Computer Vision*, pages 381–398. Springer.

Xiangxi Shi, Zhonghua Wu, and Stefan Lee. 2024. Viewpoint-aware visual grounding in 3d scenes. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14056–14065.

Michael K Tanenhaus, Michael J Spivey-Knowlton, Kathleen M Eberhard, and Julie C Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268(5217):1632–1634.

Anne M Treisman and Garry Gelade. 1980. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136.

Ozan Unal, Christos Sakaridis, Suman Saha, Fisher Yu, and Luc Van Gool. 2023. Four ways to improve verbo-visual fusion for dense 3d visual grounding. In *European Conference on Computer Vision*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xinyi Wang, Na Zhao, Zhiyuan Han, Dan Guo, and Xun Yang. 2025a. Augrefer: Advancing 3d visual grounding via cross-modal augmentation and spatial relation-based referring. In *AAAI Conference on Artificial Intelligence*.

Yuan Wang, Yali Li, and Shen Wang. 2024. G3-lq: Marrying hyperbolic alignment with explicit semantic-geometric modeling for 3d visual grounding. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13917–13926.

Yuan Wang, Yali Li, Eastman Z. Y. Wu, and Shen Wang. 2025b. Liba: Language instructed multi-granularity bridge assistant for 3d visual grounding. In *AAAI Conference on Artificial Intelligence*.

Jeremy M Wolfe. 1994. Guided search 2.0 a revised model of visual search. *Psychonomic bulletin & review*, 1(2):202–238.

Changli Wu, Yiwei Ma, Qi Chen, Haowei Wang, Gen Luo, Jiayi Ji, and Xiaoshuai Sun. 2023a. 3d-stmn: Dependency-driven superpoint-text matching network for end-to-end 3d referring expression segmentation. In *AAAI Conference on Artificial Intelligence*.

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. 2025. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22193–22204.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. 2024. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851.

Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. 2023b. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Can Xu, Yuehui Han, Rui Xu, Le Hui, Jin Xie, and Jian Yang. 2024a. Multi-attribute interactions matter for 3d visual grounding. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17253–17262.

Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. 2024b. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. In *Conference on Robot Learning*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Li Yang, Chunfen Yuan, Ziqi Zhang, Zhongang Qi, Yan Xu, Wei Liu, Ying Shan, Bing Li, Weiping Yang, Peng Li, Yan Wang, and Weiming Hu. 2023. Exploiting contextual objects and relations for 3d visual grounding. In *Neural Information Processing Systems*.

Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. 2021. Sat: 2d semantics assisted training for 3d visual grounding. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1836–1846.

Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Xiaoshui Huang, Zhiyong Wang, Lu Sheng, Lei Bai, et al. 2023. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *Advances in Neural Information Processing Systems*, 36:26650–26685.

Zhihao Yuan, Jinke Ren, Chun-Mei Feng, Hengshuang Zhao, Shuguang Cui, and Zhen Li. 2024. Visual programming for zero-shot open-vocabulary 3d visual grounding. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20623–20633.

Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Zhen Li, and Shuguang Cui. 2021. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1771–1780.

Yangfan Zhan, Yuan Yuan, and Zhitong Xiong. 2023. Mono3dvg: 3d visual grounding in monocular images. In *AAAI Conference on Artificial Intelligence*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in neural information processing systems*, 32.

China. Xiaoyan Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2023. Appagent: Multimodal agents as smartphone users. *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*.

Haomeng Zhang, Chiao-An Yang, and Raymond A. Yeh. 2024a. Multi-object 3d grounding with dynamic modules and language-informed spatial attention. In *Proc. NeurIPS*.

Yuqi Zhang, Han Luo, and Yinjie Lei. 2024b. Towards clip-driven language-free 3d visual grounding via 2d-3d relational enhancement and consistency. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13063–13072.

Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 2021. 3dvg-transformer: Relation modeling for visual grounding on point clouds. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2908–2917.

Gengze Zhou, Yicong Hong, and Qi Wu. 2023. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *AAAI Conference on Artificial Intelligence*.

Chenming Zhu, Tai Wang, Wenwei Zhang, Kai Chen, and Xihui Liu. 2024a. Scanreason: Empowering 3d visual grounding with reasoning capabilities. In *European Conference on Computer Vision*.

He Zhu, Quyu Kong, Kechun Xu, Xunlong Xia, Bing Deng, Jieping Ye, Rong Xiong, and Yue Wang. 2025a. Grounding 3d object affordance with language instructions, visual observations and interactions. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17337–17346.

Yihang Zhu, Jinhao Zhang, Yuxuan Wang, Aming Wu, and Cheng Deng. 2025b. Vgmamba: Attribute-to-location clue reasoning for quantity-agnostic 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5295–5304.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 2023. 3d-vista: Pre-trained transformer for 3d vision and text alignment. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2899–2909.

Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. 2024b. Unifying 3d vision-language understanding via promptable queries. In *European Conference on Computer Vision*, pages 188–206. Springer.

PruneGround Supplementary

Contents

A Cognitive Motivation **14**

B Camera Setup Visualization **15**

C VLM Query Structure **16**

D Additional Ablation Study **20**

E Model Structure and Training Procedure **23**

F Pipeline Visualization **24**

G Failure Case Analysis **25**

A Cognitive Motivation

Our motivation is also consistent with evidence from human visual cognition. Humans rarely interpret a complex scene by exhaustively comparing all objects of the same category. Instead, visual search is guided by selective attention (Treisman and Gelade, 1980; Wolfe, 1994): early perceptual processing extracts broad cues in parallel, while more complex object identification and relational reasoning are applied only to a limited set of task-relevant regions. In situated language understanding, linguistic cues further modulate this attentional process. Eye-tracking studies show that listeners rapidly direct their gaze toward objects mentioned or implied by an unfolding utterance, indicating that language incrementally guides visual attention in the surrounding environment (Tanenhaus et al., 1995; Huettig et al., 2011).

This cognitive process is especially relevant to 3DVG. Spatial language typically describes a target object, or located object, with respect to one or more reference objects, such as "the chair closest to the door" or "the chair under chalkboard". Cognitive studies of spatial language suggest that understanding such expressions involves identifying the relevant objects, assigning a reference frame, and evaluating the spatial relation between the target and the anchor object rather than globally reasoning over every possible candidate in the scene (Logan, 1994; Carlson and Logan, 2001).

Inspired by this human-like strategy, our **PruneGround** avoids treating all object candidates as equally relevant. Given a referring expression, it first uses the mentioned anchors and spatial relations to narrow the search space to a language-conditioned local region (**LGSP**). The model then performs fine-grained grounding within this region, augmenting with candidates that are spatially and semantically plausible (**MCDR**). This design mirrors how humans use linguistic context to guide attention toward informative scene regions, reducing unnecessary comparison with irrelevant distractors while preserving the relational cues needed for accurate localization. Consequently, the localization (**LLM-Grounder**) provides a cognitively grounded alternative to exhaustive *segment-then-select* pipelines, improving both accuracy and robustness in cluttered 3D environments.

B Camera Setup Visualization

Global View Rendering. We provide a visualization of the camera setup for *Global View Rendering* - a part of LGSP in Sec. 3.2. As shown in Fig. 4, four cameras are placed at the centers of the four upper edges of the 3D bounding box, each oriented toward the center of the opposite face.

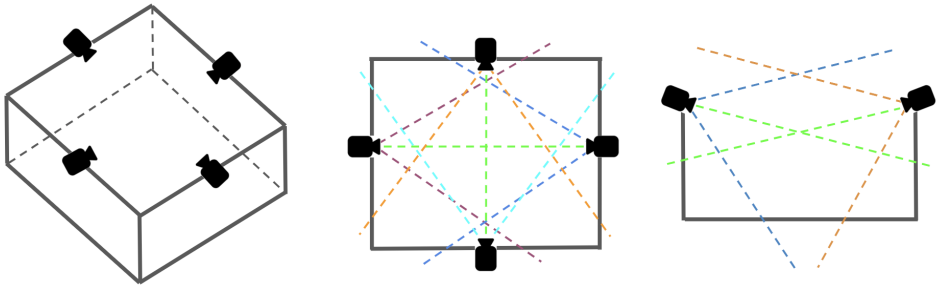


Figure 4: **Camera setup visualization:** ***Global View Rendering***. From left to right: Cameras with 3D object bounding boxes, top-view visualization, and side-view visualization.

Local View Rendering. We provide a visualization of the camera setup for *Local View Rendering* - a part of MCDR in Sec. 3.3. Given a 3D bounding box, we render side views in both the horizontal and vertical directions. As shown in the first example in Fig. 5, for each direction there are two possible camera positions that can cover the box; however, we select the one closer to the center of the original scene (the center of the black box). Note that the camera is then slightly elevated to obtain a clearer view and to prevent occlusion by surrounding furniture.

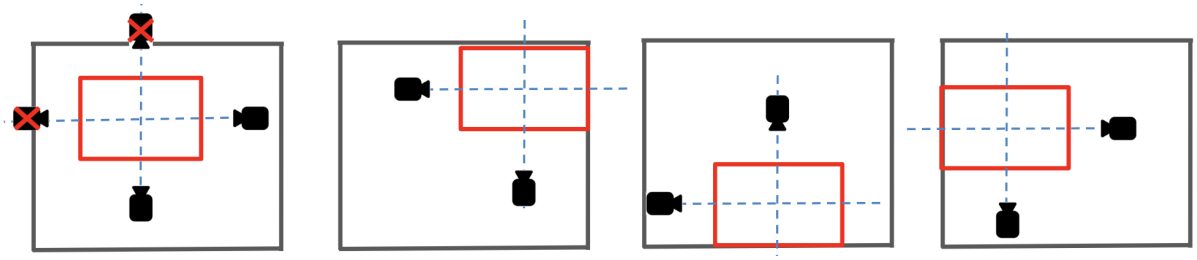


Figure 5: **Camera setup visualization:** ***Local View Rendering*** in four scenarios.

C VLM Query Structure

In this section, we describe the query structure, which consists of two tasks.

Prompt for 2D Bounding Box Estimation. The Fig. 6 shows the prompt for 2D bounding box estimation. Given the input description, the scene size (m, n, k), and the top-view image resolution (w, h), the objective is to estimate the corresponding 2D bounding box on the top-view image. This prompt is used for **Language-Guided Region Selection** in LGSP (Sec. 3.2).

PruneGround Prompt for2D Bounding Box Estimation

```
messages = [
    {
        "role": "user",
        "content": [

            {"type": "image", "image": "topview.png"},

            {"type": "image", "image": "view_1_rgb.png"},
            {"type": "image", "image": "view_1_depth.png"},

            {"type": "image", "image": "view_2_rgb.png"},
            {"type": "image", "image": "view_2_depth.png"},

            {"type": "image", "image": "view_3_rgb.png"},
            {"type": "image", "image": "view_3_depth.png"},

            {"type": "image", "image": "view_4_rgb.png"},
            {"type": "image", "image": "view_4_depth.png"},

            {
                "type": "text",
                "text": f"""
Task:
Given the description: "{description}", Determine the relevant region in the topview image as
    a 2D bounding box.

Image order:
The images appear in the following order:
1. topview.png
2. view_1_rgb.png
3. view_1_depth.png
4. view_2_rgb.png
5. view_2_depth.png
6. view_3_rgb.png
7. view_3_depth.png
8. view_4_rgb.png
9. view_4_depth.png

Scene coordinate system:
- X axis: horizontal width = {m} meters
- Y axis: vertical length = {n} meters
- Z axis: upward height = {k} meters

Topview image:
- topview.png is a topview rendering of the scene
- width = {w} pixels corresponding to the X dimension
- height = {h} pixels corresponding to the Y dimension

Image coordinate system:
- (0, 0) is the top-left corner of the topview image
- x increases to the right
- y increases downward

Oblique views:
- Each oblique view consists of two consecutive images:
first the RGB image, then the corresponding depth image.
```

```
- The cameras are placed at the midpoint of the four top
edges of the axis-aligned 3D bounding box.

View mapping:
- view_1 (images 2-3: view_1_rgb.png, view_1_depth.png)
  camera looks from bottom to top in the topview
- view_2 (images 4-5: view_2_rgb.png, view_2_depth.png)
  camera looks from right to left in the topview
- view_3 (images 6-7: view_3_rgb.png, view_3_depth.png)
  camera looks from top to bottom in the topview
- view_4 (images 8-9: view_4_rgb.png, view_4_depth.png)
  camera looks from left to right in the topview

Reasoning procedure:
Step 1: Identify possible target objects and anchors mentioned in the description.
Step 2: Use the topview and oblique RGB-depth images to determine plausible spatial regions
    containing target candidates.
Step 3: Determine the minimal bounding region that covers those candidates.

Bounding box requirements:
- cover all candidate regions that could plausibly contain the target object
- exclude regions that are unlikely to contain the target
- include anchors mentioned in the description only when they are spatially close to the
    target candidates
- if the anchors are far from the target candidates, they should not be included
- the bounding box should be as tight as possible while still covering all plausible target
    candidates
- the bounding box must not exceed the scene boundary and should avoid covering large empty
    areas

If you are uncertain, prefer a slightly larger box that still excludes clearly irrelevant
    regions.

Output format:
- Return ONLY four values: x_min, x_max, y_min, y_max
- All coordinates must be in pixel coordinates of the topview image.
- Output ONLY the four numbers separated by commas.
- Do not output any explanation or additional text.
"""
            },
        ],
    }
]
```

Figure 6: **Prompt for 2D bounding box estimation: Top-view visual grounding.**

Prompt for MCDR. The VLM query for **Description Simplification and Augmentation** - a part of MCDR (Sec. 3.3) is described in Fig. 7. Given rendered local multi-view images and the original description, we prompt the VLM to augment the relational information by using the local multi-view images, thereby incorporating not only reformulated content from the original description but also additional target–anchor pairs.

PruneGround Prompt for MCDR

```
messages = [
    {
        "role": "user",
        "content": [

            {"type": "image", "image": "side_1.png"},
            {"type": "image", "image": "side_2.png"},

            {
                "type": "text",
                "text": f"""
Task:

You are given two side-view images of a pruned 3D scene.

Images:
1. side_1.png
2. side_2.png

Description: "{description}"
Your goal is to analyze the description and the two side views to extract the target object
    and anchors, and reformulate the description into structured single target-single anchor
    utterances.

Reasoning procedure:
Step 1 - Analysis
Analyze the description and side views to identify:
- the target object
- the list of anchors explicitly mentioned
in the description
Step 2 - Visual Anchor Detection
From the two side-view images, identify additional anchors that are visible in the scene but
    NOT mentioned in the description.
Step 3 - Reformulate Description
Reformulate the description into a list of single target - single anchor utterances.
Each utterance should describe the spatial relation between the target and one anchor.

Output Structure:
- target name
- anchors name:
[list of anchors mentioned in the description]
- anchors not in description but visible:
[list of anchors visible in side views but not mentioned in the description]
- reformulated_description:
   + target properties
   (short and concise description of the target's properties)
   + augmented target properties
   (additional target properties that can be inferred
   from the side views but are not explicitly described)
   + single target single anchor utterances
   (list of utterances where each describes the relation between the target and one anchor)
   + augmented single target - single anchor utterances
   (list of utterances using anchors that are not mentioned in the description but detected
       in the side views)

Constraints:
- target properties and augmented target properties must be short and concise.
- the TOTAL number of single target - single anchor utterances must be at most 5.
```

```
- each utterance must follow this format:
target (+properties) + spatial relation + anchor (+properties)
- Use only clear spatial relations such as:
left of, right of, in front of, behind, next to, under, on, ...
- Do not produce explanations.
- Return only the structured output.

Example target properties:
- the chair is wooden and yellow
Example utterance:
- the chair is behind the black coffee table
"""
            },
        ],
    }
]
```

Figure 7: **Prompt for MCDR.**

D Additional Ablation Study

Inference analysis. In this study, we report the runtime and resource consumption of PruneGround during inference. We evaluate our pipeline on GPU RTX A100, reported by Tab. 7.

Module	Runtime	GPU usage
LGSP	1.4	10
MCDR	1.3	8
LLM-Grounder	1.4	5
PruneGround	4.1 (total)	10 (max)

Table 7: **Inference analysis** based on running time (in seconds) and GPU usage (in GBs) of our framework.The frozen VLM used in LGSP and MCDR is Qwen2.5-VL-3B-Instruct. Its inference time comparison against other VLMs is shown in Tab. 6.

Performance with other grounders. Since our pruning mechanism is designed as a *plug-and-play* module, the pruned regions can be directly used as input to other grounders. To verify that reducing the search space improves grounding performance, we retrain ScanRefer (Chen et al., 2020) and ReferIt3D (Achlioptas et al., 2020), which are foundational works that introduced both the methods and datasets for the 3DVG task. Both methods follow the *segment-then-select* paradigm. As shown in Tab. 8, integrating our language-based pruning consistently improves their performance, demonstrating its effectiveness for 3DVG. Additionally, we apply our pruning module to 3DVG-Transformer (Zhao et al., 2021), a representative *one-stage* approach, and observe consistent improvements.

Grounder	ScanRefer Acc@0.25	ScanRefer Acc@0.5	Nr3D Overall	Sr3D Overall
ScanRefer (*two-stage*)	37.30	24.32	-	-
ScanRefer + LGSP	**44.53** +7.23	**28.46** +4.14	-	-
ReferIt3D (*two-stage*)	-	-	35.6	40.8
ReferIt3D + LGSP	-	-	**43.8** +3.2	**44.6** +3.8
3DVG-Transformer (*one-stage*)	45.90	34.47	40.8	51.4
3DVG-Transformer + LGSP	**50.25** +4.35	**37.52** +3.05	**45.3** +4.5	**55.6** +4.2

Table 8: **Performance of LGSP and MCDR when plug-and-played with different grounding methods.**

Ablation study of proposed modules on Nr3D and Sr3D. We further provide an ablation study of each proposed module on the Nr3D and Sr3D datasets in Tab. 9, complementing the study on ScanRefer dataset in Tab. 3. Consistent improvements are observed across both datasets, confirming the effectiveness of the proposed modules.

Method	**Nr3D**					**Sr3D**				
	Easy	Hard	View-D	View-I	Overall	Easy	Hard	View-D	View-I	Overall
LLM-Grounder	79.8	61.7	66.2	72.1	69.7	77.0	69.7	56.1	75.6	74.8
+ LGSP	82.1 +2.3	64.6 +2.9	68.4 +2.2	74.3 +2.2	72.1 +2.4	79.2 +2.2	72.9 +3.2	58.6 +2.5	78.1 +2.5	77.3 +2.5
+ MCDR	83.5 +2.2	66.8 +2.2	69.8 +1.4	75.3 +1.0	73.6 +1.5	80.8 +1.6	75.1 +2.2	59.6 +1.0	80.2 +2.1	79.1 +1.8
+ MCDR (S + A)	**84.6** +1.1	**68.8** +2.0	**71.2** +1.4	**76.6** +1.3	**75.1** +1.5	**83.7** +2.9	**76.5** +1.4	**61.7** +2.1	**82.4** +2.2	**81.5** +2.4

Table 9: **Impact of each proposal module on Nr3D and Sr3D datasets**. Baseline is LLM-Grounder trained on *original* descriptions and *unpruned* scenes, then incrementally add LGSP for scene pruning and MCDR for description reformulation (via simplification (S) or simplification + augmentation (S + A)).

Anchor verification analysis. To assess whether MCDR introduces meaningful spatial information rather than spurious content, we analyze the anchors before and after reformulation. For each augmented anchor (i.e., anchors not present in the original description but added by MCDR), we verify its existence against ScanNet object annotations within the pruned region. An augmented anchor is marked as *verified*

Dataset	# Anch. (Orig.)	# Anch. MCDR	Verif. (%)	Halluc. (%)
ScanRefer	2.14	3.22	93.4	6.6
Nr3D	1.07	2.12	91.7	8.3
Sr3D	0.72	1.75	96.1	5.9
Average	1.31	2.36	93.7	6.3

Table 10: **Anchor verification** on 500 sampled descriptions per dataset. **Verif.**: augmented anchors whose class exists in the pruned region. **Halluc.**: augmented anchors with no corresponding object instance in the pruned region.

Method	Faith. ↑	Compl. ↑	Sp. Prec. ↑	Concise. ↑
Original	4.21	2.84	3.15	**4.52**
Text-only Reformulation	3.58	3.62	3.23	3.31
MCDR (Ours)	**4.47**	**4.39**	**4.28**	3.67
Krippendorff's α	0.72	0.69	0.74	0.61

Table 11: **Human evaluation of description quality** (mean rating on a 5-point scale across 150 samples, 3 annotators). Bold indicates the best per column. Krippendorff's $\alpha \geq 0.67$ indicates substantial agreement across all dimensions. *Text-only Reformulation* follows a similar text only restructuring philosophy to ViewSRD (Huang et al., 2025), but uses our prompting protocol, the comparison thus isolates the contribution of visual conditioning in MCDR.

if an instance of its class label actually exists in the pruned scene, and *hallucinated* otherwise. We perform this analysis on 500 randomly sampled descriptions from each dataset.

As shown in Tab. 10, MCDR increases the average number of anchors per description from 2.14 (original) to 3.22 (reformulated) on ScanRefer, with similar trends on Nr3D and Sr3D. Among the augmented anchors, average 93.7% are verified as existing in the pruned scene. The hallucination rate remains below 6.3%, demonstrating that visual conditioning effectively constrains the VLM to ground augmentations in actual scene content rather than fabricating plausible-sounding but non-existent references. This verification analysis confirms that the additional anchors introduced by MCDR are predominantly grounded in the scene and contribute genuine spatial information, supporting the downstream grounding improvements.

Human evaluation of Reformulated Descriptions. To assess the linguistic quality of MCDR beyond downstream grounding accuracy, we conduct a human study comparing three description variants: (i) the *Original* free-form utterance, (ii) a *Text-only Reformulation* produced by prompting the same VLM with the original description but no visual input, and (iii) MCDR (ours), which conditions the VLM on multi-view renderings of the pruned scene.

We randomly sample 150 descriptions from ScanRefer (50), Nr3D (50), and Sr3D (50). Three annotators with backgrounds in computer vision and natural language processing independently rate each description on a 5-point scale (1 = poor, 5 = excellent) along four dimensions: (i) **Faithfulness** – whether the reformulation preserves the original intent without distorting the target referent; (ii) **Completeness** – whether useful spatial information is added relative to the original; (iii) **Spatial Precision** – whether the spatial relations are accurate and specific (e.g., "left of" vs. vague "near"); (iv) **Conciseness** – whether the description is compact and free of redundancy. Annotators are provided with the corresponding 3D scene visualization and the ground-truth target bounding box. Inter-annotator agreement is measured via Krippendorff's α.

As reported in Tab. 11, MCDR achieves the highest scores on **Faithfulness**, **Completeness**, and **Spatial Precision**, outperforming both the Original utterance and the Text-only Reformulation. Notably, while Text-only Reformulation also enriches the original description with additional content, it underperforms MCDR on Faithfulness and Spatial Precision – a gap attributable to its lack of visual grounding, which causes the VLM to fabricate plausible-sounding but inaccurate spatial relations. In contrast, MCDR leverages multi-view scene context to anchor augmentations in actual geometry, yielding spatially precise

and faithful descriptions. The Conciseness score of **MCDR** is slightly lower than the Original, since reformulation inevitably expands the utterance, however, this trade-off is justified by the substantial gains in Completeness and Spatial Precision, which translate into improved downstream grounding. These results indicate that the improvements brought by **MCDR** are not merely an artifact of training-time supervision, but stem from genuine enhancements in the linguistic quality of the referring expressions themselves.

E Model Structure and Training Procedure

Qwen2.5-VL-3B-Instruct. We use this frozen VLM for inference inside LGSP and MCDR. The language backbone consists of 36 transformer layers (Vaswani et al., 2017) with a hidden size of 2048, SiLU activations (Elfwing et al., 2018), RMS normalization (Zhang and Sennrich, 2019), 16 attention heads, and grouped-query attention with 2 key–value heads. The model uses a vocabulary of 151,936 tokens, tied input–output embeddings, bfloat16 precision, rotary positional embeddings with $\theta_{\mathrm{RoPE}} = 1{,}000{,}000$, and mRoPE scaling for multimodal position encoding, supporting a maximum context length of 128,000 tokens. Although the configuration specifies a sliding-window size of 32,768 tokens, sliding-window attention was disabled during inference. The visual encoder contains 32 layers (Dosovitskiy et al., 2021) with hidden size 1280, 16 attention heads, 14×14 spatial patches, spatial merge size 2, and selected full-attention blocks at layers 7, 15, 23, and 31; its output is projected to the 2048-dimensional language-model hidden space. We used the model strictly for inference, with caching enabled and without any parameter updates or additional training.

SpatialLM. As LLM-Grounder, we use **SpatialLM1.1-Qwen-0.5B**, a spatial-language causal decoder model. SpatialLM is designed with an Encoder–MLP–LLM architecture. We use Qwen2.5-0.5B (Yang et al., 2024) as the base LLM, Sonata (Wu et al., 2025) as the point-cloud encoder, and a two-layer MLP as the projector. The language backbone follows a Qwen-style transformer architecture (Vaswani et al., 2017) with 24 hidden layers, hidden size 896, intermediate size 4864, SiLU activations (Elfwing et al., 2018), RMS normalization with $\epsilon = 10^{-6}$ (Zhang and Sennrich, 2019), and 14 attention heads with grouped-query attention using 2 key-value heads. The model uses a vocabulary of 151,936 tokens with tied input-output embeddings, rotary positional embeddings with `rope_theta`$= 1{,}000{,}000$, a maximum context length of 32,768 tokens, and no sliding-window attention. Spatial inputs are represented using dedicated point start, end, and point tokens, and encoded with a Sonata point backbone. The point encoder consumes 6-channel point features and uses five stages with channel widths $[48, 96, 192, 384, 512]$, depths $[3, 3, 3, 12, 3]$, multi-head attention widths $[3, 6, 12, 24, 32]$, patch size 1024, stride-2 downsampling between stages, masked-token support, and spatial ordering based on z-order, transposed z-order, Hilbert, and transposed Hilbert curves. Point features are mapped into the language-model space through an MLP projector.

Sonata. As a frozen encoder of LLM-Grounder, we used the **encoder-only** configuration. The encoder consists of five hierarchical stages with channel dimensions $[48, 96, 192, 384, 512]$, depths $[3, 3, 3, 12, 3]$, with a patch size of 1024 at each stage and stride-2 down-sampling between stages. The model uses pre-normalization, an MLP expansion ratio of 4, learned query/key/value biases, mask tokens, stochastic depth with a drop-path rate of 0.3, and no attention or projection dropout. Flash attention (Dao et al., 2022) is enabled for efficient inference, while relative positional encoding, attention upcasting, and softmax upcasting are disabled. Sonata serializes points using multiple space-filling curve orders, including Z-order, transformed Z-order, Hilbert order, and transformed Hilbert order, with order shuffling enabled.

Training Protocol. Since ScanRefer, Nr3D, and Sr3D are all built upon the ScanNet dataset, we adopt a two-stage training strategy.
In Stage 1, the model is first pre-trained on the SpatialLM synthetic dataset, then further trained for 3 epochs on the same dataset with class labels mapped to the ScanNet taxonomy, and finally fine-tuned on ScanNet for 30 epochs. All steps in this stage are conducted on the object detection task by (Mao et al., 2025).
In Stage 2, the model is fine-tuned on the 3DVG task for 10 epochs with a batch size of 4 on each dataset (ScanRefer, Nr3D, Sr3D) separately. We use the AdamW optimizer with a learning rate of 1e-4, a cosine scheduler, and a warm-up ratio of 0.03; all other hyperparameters remain unchanged. All components (encoder, projector, and LLM) are jointly trainable.

F Pipeline Visualization

In this section, we provide visualizations of each step in our PruneGround pipeline in Fig.8: (i) LGSP, (ii) MCDR, (iii) LLM-Grounder.



Figure 8: **Pipeline visualization.**
Top row: input description, input scene, and 4 × oblique views (RGB and depth images) rendered inside LGSP.
Middle row: top view with the 2D bounding box predicted by the frozen VLM, the pruned scene, and two side views.
Bottom row: reformulated description generated by MCDR and the final localization result by LLM-Grounder (red: prediction; green: ground truth).

G Failure Case Analysis

In this section, we discuss the remaining limitations of our proposed framework.

Rendering failures leads to pruning failures. First, rendering oblique views of complex indoor scenes remains challenging, particularly under low lighting, with incomplete point clouds, or in multi-room layouts. In such environments, objects are frequently occluded by walls or surrounding furniture, resulting in low-quality renderings. As illustrated in Fig. 9(a), these degraded images can adversely affect the performance of the VLM in identifying language-relevant regions, then lead to final incorrect localization.

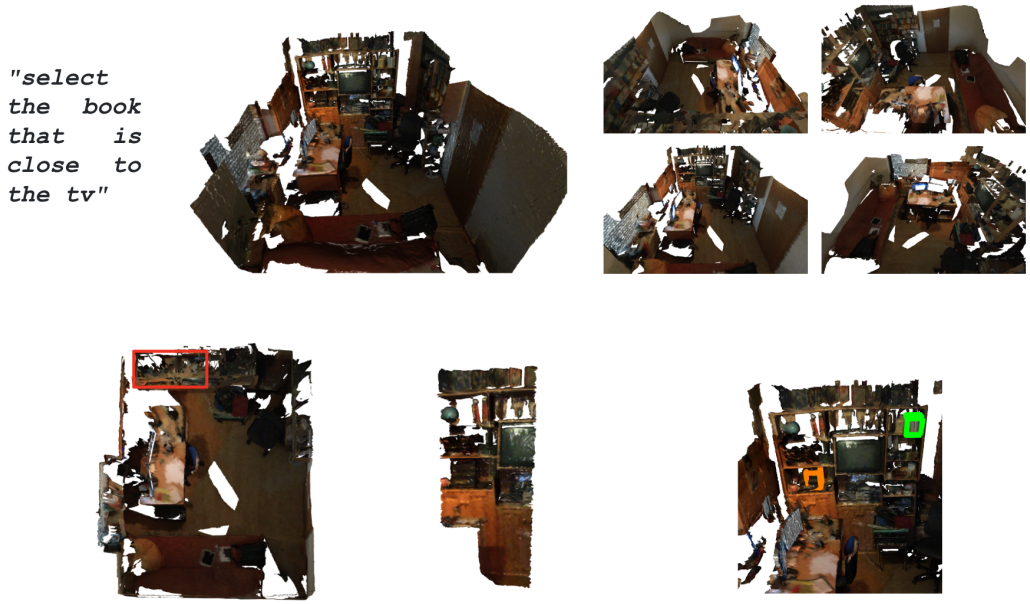


Figure 9: **Failure case analysis**.
Top row: input description, input scene, and 4 $\times$ oblique views (RGB) rendered inside **LGSP**.
Bottom row: top view with the 2D bounding box predicted inside **MCDR**, the pruned scene by **LGSP**, and 3D bounding box predicted by **LLM-Grounder** after mapping to original scene.
(red: prediction; green: ground truth).