Tool-Augmented VLM Agents for Zero-Shot 3D Visual Grounding

Cuong Huynh [1] **Maxim Popov** [1] **Denis Gridusov** [1] **Sergey Kolyubin** [1]

Abstract

3D Visual Grounding (3DVG) is an essential capability for embodied AI, requiring agents to localize objects in 3D scenes based on natural language descriptions. Recent zero-shot methods leverage Large Vision-Language Models (LVLMs). However, they often rely on existing sets of multi-view images and struggle with the limited semantic and spatial details provided by standard 3D segmentation tools. We present ***AgentGrounder***, a zero-shot 3D visual grounding framework that operates directly on colored point clouds without task-specific 3D training. Our approach follows a two-stage design: (1) an offline stage that applies a 3D model to build an Object Lookup Table (OLT) with instance IDs, semantic labels, 3D bounding boxes; and (2) an online tool-driven agent that decomposes each query, retrieves only relevant candidates from the OLT, performs geometric scoring, and triggers image rendering on demand when additional visual evidence (e.g., color, material, or viewpoint-sensitive cues) is required. Compared with fixed anchor-target matching pipelines, this design reduces cascading matching errors and avoids prompts filled with irrelevant objects. We evaluate on ScanRefer and Nr3D under a zero-shot setting and observe consistent improvements over SeeGround, including +2.5% Acc@0.5 on ScanRefer and +6.3% on Nr3D. Notably, gains are strongest on challenging Nr3D queries: +7.1% on *Hard*, +6.3% on *View-Independent*, and +5.6% on *View-Dependent*. These results show that combining selective retrieval, geometric reasoning, and adaptive visual inspection yields a practical and robust foundation for open-vocabulary 3D grounding.

[1]Biomechatronics and Energy-Efficient Robotics (BE2R) Lab, ITMO University, Saint Petersburg, Russia. Correspondence to: Cuong Huynh <tchuynh@itmo.ru>.

Preprint. June 6, 2026.

1. Introduction

3D Visual Grounding (3DVG) is a fundamental task in computer vision and robotics, aimed at localizing a target object within a 3D scene based on a natural language query. This capability is essential for enabling autonomous agents to interact seamlessly with their environment — for instance, following a command to "`pick up the white porcelain sink next to the counter`." Most existing 3DVG methods (Wang et al., 2024; Qian et al., 2024; Huang et al., 2024; Zheng et al., 2025) follow a supervised learning paradigm, requiring large-scale datasets with dense annotations of 3D bounding boxes paired with linguistic descriptions, which is labor-intensive and limits generalization to unseen categories. To address these limitations, zero-shot 3DVG (Yang et al., 2024; Yuan et al., 2024b; Xu et al., 2025; Li et al., 2025; Yuan et al., 2024a; Zantout et al., 2025; Fang et al., 2024) has emerged as a promising direction, leveraging the rich knowledge of Vision-Language Models (VLMs) and Large Language Models (LLMs) without task-specific 3D training. This line of work is further supported by progress in open-vocabulary 3D segmentation (Takmaz et al., 2023; Ngo et al., 2023; Yang et al., 2023; Nguyen et al., 2024; 2025) and multimodal spatial reasoning (Jatavallabhula et al., 2023; Gu et al., 2024; Guo et al., 2025; Wen et al., 2025), which project 2D foundation knowledge into 3D representations. Despite encouraging progress, existing untrained data processing pipelines still face key challenges: (1) logical reasoning is often unreliable due to early anchor-target matching errors, (2) visual analysis is not always selective, leading to unnecessary computations and token overhead, and (3) geometric relationships in 3D scenes are not always exploited in a consistently transparent and deterministic manner.

In this paper, we propose ***AgentGrounder***, a zero-shot 3DVG framework that operates directly on colored 3D point clouds with a tool-driven agent. Our method follows a two-stage design: first, we run a 3D model to obtain instance-level segmentation and build an Object Lookup Table (OLT) containing object IDs, semantic labels, centers, and box sizes. Second, an online LVLM agent performs query decomposition, retrieves only relevant objects from the OLT, applies deterministic geometric scoring, and calls image

rendering on demand when a visual prompt (e.g., color, material, or viewpoint-sensitive cues) is required. Compared with fixed matching pipelines, this design reduces cascading matching errors and avoids prompts filled with irrelevant objects. We validated our method on ScanRefer (Chen et al., 2020) and Nr3D (Achlioptas et al., 2020), achieving consistent gains over SeeGround in our setting and showing state-of-the-art results.

2. Methodology

Overview. We studied 3D visual grounding where, given a language query Q and a scene S, the system predicts the target object and its 3D box. The task is formulated as:

$$bbox = 3DVG(S, Q) \tag{1}$$

where $bbox \in \mathbb{R}^6$ encodes the 3D box center and size, i.e. $bbox = (c, d)$, $c \in \mathbb{R}^3$ is the object center, and $d \in \mathbb{R}^3$ is the box size.

2.1. Initial 3D Instance Segmentation

As the first stage, we ran a 3D instance segmentation network on scene S to obtain per-object masks and initial semantic labels:

$$\{(mask_i, sem_i)\}_{i=1}^{N} = Seg(S). \tag{2}$$

For each predicted mask, we computed the corresponding 3D bounding box:

$$bbox_i = Bound(mask_i). \tag{3}$$

This stage was executed offline once per scene and provided the object candidates used in subsequent online grounding.

2.2. Object Lookup Table

At next stage, each scene was represented as an Object Lookup Table (OLT) constructed from the segmentation outputs:

$$\mathcal{O} = \{(id_i, \ell_i, bbox_i)\}_{i=1}^{N}, \tag{4}$$

where ℓ_i is the semantic label. During online inference, the agent queries this table using tool calls (e.g., by label) and obtains pre-calculated 3D segmentation results.

2.3. Agent Planning and Query Decomposition

Given a language query Q, the agent first generated an explicit plan $\mathcal{R}$ that dictated (i) which tools to call and (ii) how to make decisions from the returned metadata. Concretely, it decomposed Q into the target object category (head noun), descriptive attributes (e.g., color/material), spatial expressions (e.g., `left of`, `closest to`), and any anchor objects mentioned for disambiguation:

$$(\mathcal{P}, \mathcal{L}, \mathcal{R}) = PlanExtract(Q), \tag{5}$$

where $\mathcal{L}$ is the set of candidate labels and $\mathcal{R}$ contains spatial predicates (e.g., `next to`, `left of`, `below`, `closest to`). Plan $\mathcal{P}$ guides subsequent tool calls and reasoning steps, allowing the agent to adapt its strategy based on query complexity and content. To prevent unbounded or recursive planning, we limit the maximum plan length. Without explicit planning, the agent must rely on implicit reasoning within a single prompt, which can lead to less structured and more error-prone inference, especially for complex queries with multiple spatial relations or attributes.

2.4. Candidate Retrieval and Geometric Scoring

The agent retrieved candidates $\mathcal{C}$ by labels from $\mathcal{O}$:

$$\mathcal{C} = \{o_i \in \mathcal{O} \mid \ell_i \in \mathcal{L}\}. \tag{6}$$

To answer spatial queries, the agent evaluated the predicates in $\mathcal{R}$ using simple geometry derived from the OLT: (i) distances $d(i, j)$ for proximity relations, (ii) coordinate comparisons for directional relations (left/right, above/below) in the selected reference frame, and (iii) box dimensions as proxies for size constraints (e.g., area/volume). For distance-related expressions (e.g., "`closest to the TV`") it selected the candidate with the smallest or largest distance to the referenced object. For between relations (e.g., "`between two tables`"), it picked candidates that were close to both referenced objects:

$$\begin{aligned} d(i, j) &= \|c_i - c_j\|_2, \\ \hat{o} &= Resolve(\mathcal{C}, \mathcal{R}, \mathcal{O}, d(i, j)). \end{aligned} \tag{7}$$

2.5. View-Dependent Disambiguation by Rendering

If the query contained viewpoint-sensitive language (e.g., `when facing`, `on the right`), the agent called a rendering tool over a subset of candidate IDs and used visual evidence to resolve ambiguity:

$$I = Render(S, \mathcal{I}_{cand}), \qquad \hat{o} = Resolve(I, Q). \tag{8}$$

The camera pose for rendering is computed such that the camera looks at the mean 3D center of the relevant objects from OLT. Object relevancy is defined inside the agent without hard-coded assumptions. For non-view-dependent queries with clear geometric ranking, this step was skipped.

2.6. Robust Handling of Label Mismatch

When the query Q referred to an object, the agent first used tool calls to retrieve candidates from the Object Lookup Table (OLT) by semantic label. If the label lookup returned no results (i.e., the mentioned object/label was not present in the OLT; e.g., `board`, `fridge`), the agent mapped that term to the *closest* available object/label in the OLT based on context (e.g., `tv`, `kitchen cabinet`). After this

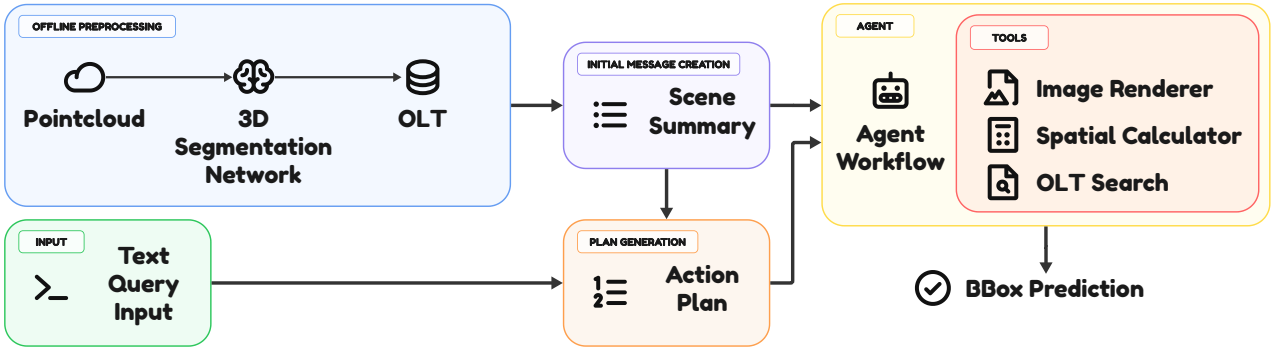

Figure 1. **Overview of the *AgentGrounder* agent pipeline**. In the offline phase, a 3D segmentation network built an Object Lookup Table. For each scene, the system then generated a textual description that listed all detected objects. In the online phase, the agent received a query and first formulated an explicit plan. Guided by this plan, it retrieved candidate objects based on semantic labels from the scene metadata and computed geometric relationships (such as nearest/farthest, left/right, below) using 3D object centers and bounding box dimensions. For view-dependent queries, the agent invoked a rendering tool to examine selected object IDs and resolved ambiguities. Finally, it output a structured response that included the predicted object ID along with a textual justification.

mapping, the agent continued the same geometric reasoning pipeline on the new candidate set. This simple fallback increased query coverage without any retraining.

2.7. Final Prediction

The final output was a structured triplet containing object ID, bounding box, and explanation:

$$\hat{bbox} = BBox(\hat{o}), \qquad \text{answer} = (\hat{o}, \hat{bbox}, \text{rationale}). \tag{9}$$

This formulation captures the agent behavior during inference: metadata retrieval, geometric reasoning, selective image calls for view dependence, and structured final reporting.

3. Experiments

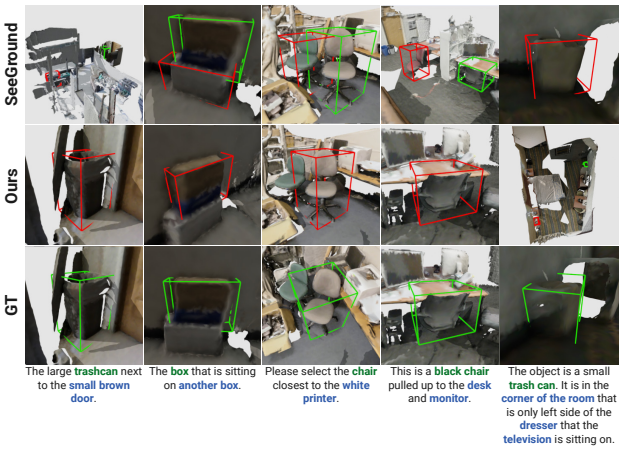

Figure 2. Results comparison between SeeGround (Li et al., 2025) and ***AgentGrounder*** (Ours).

3.1. Implementation Details

Our framework followed a two-stage pipeline. In the offline preprocessing stage, we employed Mask3D (Schult et al., 2023) for 3D instance segmentation to construct an OLT for each scene. Although Mask3D (Schult et al., 2023) was trained on ScanNet (Dai et al., 2017), it was not trained for the 3D visual grounding task, which allows us to treat our method as operating in a zero-shot setting.

For the online reasoning stage, we used Qwen3-VL-32B-Instruct (Yang et al., 2025) as the core Vision-Language Model (VLM). The model was deployed locally using the Ollama framework (Marcondes et al., 2025) and interfaced through the LangChain API (Chase, 2022).

3.2. Experimental Settings

We evaluated our proposed 3DVG approach on two commonly-used benchmark datasets: ScanRefer (Chen et al., 2020) and Nr3D (Achlioptas et al., 2020). ScanRefer provides natural language descriptions for objects in ScanNet scenes, requiring the model to distinguish target objects based on spatial context and geometric features. Queries are categorized as *Unique* or *Multiple* based on the presence of same-class distractors. Nr3D, part of the ReferIt3D benchmark, contains descriptions collected via a reference game, classified into *Easy* or *Hard* levels and further divided by viewpoint dependency (*View-Dependent* or *View-Independent*). To ensure efficient evaluation while maintaining representativeness, we evaluated on the full validation set of Nr3D and drew a 1,000-query subsample from ScanRefer using stratified random sampling to preserve the *Unique* / *Multiple* split. We followed the evaluation protocol established in ZSVG3D (Yuan et al., 2024b), using accuracy with IoU thresholds as the primary metric. The results of

Table 1. Evaluation results of 3DVG methods on *ScanRefer* (Chen et al., 2020) validation set. Results are reported for *"Unique"* (scenes with a single target object) and *"Multiple"* (scenes with distractors of the same class) subsets, along with overall performance.

Method	Venue	Zero-Shot	Agent	Unique		Multiple		Overall	
				Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
G^3-LQ (Wang et al., 2024)	CVPR'24	✗	-	88.6	73.3	50.2	39.7	56.0	44.7
MCLN (Qian et al., 2024)	ECCV'24	✗	-	86.9	72.7	52.0	40.8	57.2	45.7
ConcreteNet (Unal et al., 2024)	ECCV'24	✗	-	86.4	82.1	42.4	38.4	50.6	46.5
Chat-Scene (Huang et al., 2024)	NIPS'24	✗	Vicuna-7B	89.6	82.5	47.8	42.9	55.5	50.2
Video-3D LLM (Zheng et al., 2025)	CVPR'25	✗	LLaVA-Video 7B	88.0	78.3	50.9	45.3	58.1	51.7
GPT4Scene (Qi et al., 2025)	ICLR'26	✗	Qwen2-VL-7B	90.3	83.7	56.4	50.9	62.6	57.0
LLM-G (Yang et al., 2024)	ICRA'24	✓	GPT-3.5	-	-	-	-	14.3	4.7
LLM-G (Yang et al., 2024)	ICRA'24	✓	GPT-4 turbo	-	-	-	-	17.1	5.3
ZSVG3D (Yuan et al., 2024b)	CVPR'24	✓	GPT-4 turbo	63.8	58.4	27.7	24.6	36.4	32.7
SeeGround (Li et al., 2025)	CVPR'25	✓	Qwen2-VL-72B	<u>75.7</u>	<u>68.9</u>	34.0	<u>30.0</u>	44.1	39.4
CSVG (Yuan et al., 2024a)	BMVC'25	✓	Mistral-Large-240B	68.8	61.1	**38.4**	27.3	**49.6**	<u>39.8</u>
AgentGrounder$_{GT_bboxes}$ (Ours)	-	✓	Qwen3-VL-32B	95.1	95.1	53.4	53.4	63.5	63.5
AgentGrounder (Ours)	-	✓	Qwen3-VL-32B	**80.3**	**73.7**	<u>36.6</u>	**31.7**	<u>47.2</u>	**41.9**

Table 2. Performance results on *Nr3D* (Achlioptas et al., 2020) validation set. Queries are labeled as *"Easy"* (only one distractor) or *"Hard"* (with multiple distractors), and as *"View-Dependent"* or *"View-Independent"* based on viewpoint requirements for grounding.

Method	Easy	Hard	Dep.	Indep.	Overall
Supervision: Fully Supervised					
MiKASA (Chang et al., 2024)	69.7	59.4	65.4	64.0	64.4
ViL3DRel (Chen et al., 2022)	70.2	57.4	62.0	64.5	64.4
SceneVerse (Jia et al., 2024)	72.5	57.8	56.9	67.9	64.9
Supervision: Zero-Shot					
ZSVG3D (Yuan et al., 2024b)	46.5	31.7	36.8	40.0	39.0
VLM-Grounder (Xu et al., 2025)	<u>55.2</u>	<u>39.5</u>	<u>45.8</u>	<u>49.4</u>	<u>48.0</u>
SeeGround (Li et al., 2025)	54.5	38.3	42.3	48.2	46.1
AgentGrounder (Ours)	**59.6**	**45.4**	**47.9**	**54.5**	**52.4**

Table 3. Ablation study of agent tools on a held-out Nr3D subset ($N = 120$), measured by $Acc@0.5$.

#	Retrieval	Distance	Planning	Rendering	Easy	Hard	Dep	Indep	Overall
1	✓	✗	✗	✗	51.7	29.0	35.7	42.3	40.0
2	✓	✓	✗	✗	50.0	**40.3**	35.7	<u>50.0</u>	45.0
3	✓	✓	✓	✗	58.6	35.5	38.1	**51.3**	46.7
4	✓	✓	✗	✓	<u>62.1</u>	33.9	**50.0**	46.2	<u>47.5</u>
5	✓	✓	✓	✓	**65.5**	<u>35.5</u>	<u>47.6</u>	**51.3**	**50.0**

the methods were directly taken from original published papers. Our experiments were performed on a workstation equipped with two NVIDIA RTX 4090 GPUs, each providing 24 GB of VRAM. It took approximately one day to evaluate the entire Nr3D dataset, comprising 6,307 queries, with an average inference time of 13.5 seconds per query.

3.3. Comparative Study

Table 1 compares our proposed ***AgentGrounder*** with other 3DVG methods on the ScanRefer validation set. Our method achieved an overall Acc@0.5 of 41.9, outperforming SeeGround by 2.5% in this metric. Breaking down by subset: in the *Unique* category, we achieved Acc@0.5 of 73.7 compared to SeeGround's 68.9 (4.8% gain), demonstrating stronger spatial precision under stricter IoU thresholds. In the *Multiple* subset with same-class distractors, ***AgentGrounder*** obtained 31.7 Acc@0.5 versus SeeGround's 30.0, showing consistent improvements even in cluttered scenes requiring fine-grained reasoning. We also outperform CSVG on Acc@0.5 across both the *Unique* and *Multiple* subsets and on Acc@0.25 across *Unique*. The *Overall* Acc@0.25 metric, however, appears lower for our method because CSVG's evaluation subsample does not preserve the *Unique*/*Multiple* distribution of the full validation set, which inflates their aggregate score. When ablating the use of GT bounding boxes (Table 1), we reach 63.5 Acc@0.5, a 21.5% improvement over our full method with Mask3D proposals, indicating that while the agent can reason over noisy proposals, there remains substantial headroom in instance segmentation to boost downstream grounding.

Nr3D. Table 2 presents results on the Nr3D dataset, where ***AgentGrounder*** achieved an overall accuracy of 52.4, a 6.3% improvement over SeeGround (46.1). Gains are consistent across difficulty and viewpoint splits: on *Hard* queries, we reach 45.4 versus 38.3 (+7.1%), indicating stronger robustness to multiple distractors, while on *Easy* queries we improve to 59.6 over 54.5 (+5.1%), suggesting better precision even with fewer confounders. For viewpoint splits, ***AgentGrounder*** achieves 54.5 on *View-Independent* compared to 48.2 (+6.3%) and 47.9 on *View-Dependent* versus 42.3 (+5.6%), showing improvements in both geometry-driven and appearance-sensitive cases. Several examples of failure cases from SeeGround (Li et al., 2025) are also shown in Figure 2. All qualitative failure analyses are conducted on Nr3D.

3.4. Ablation Study

We studied how each tool in our agent contributes to 3DVG performance on a held-out Nr3D test subset ($N = 120$), stratified across *Easy/Hard* and *View-Dependent/View-Independent* queries. We evaluate overall localization accuracy at IoU threshold 0.5 (Acc@0.5).

Table 3 incrementally enables four tools: (i) *Label-based OLT Retrieval*, (ii) *Geometric Distance Reasoning*, (iii) *Explicit Planning*, and (iv) *On-demand Visual Rendering*. Starting from retrieval-only (Overall 40.0; Easy 51.7, Hard 29.0; Dep 35.7, Indep 42.3), adding geometric distance reasoning yields a large improvement to Overall 45.0, mainly driven by gains on Hard (40.3) and Indep (50.0), indicating that deterministic spatial scoring is critical for resolving distractors. Introducing explicit planning provides a smaller but consistent gain to Overall 46.7 (Easy 58.6; Dep 38.1), suggesting that structured decomposition helps the agent select and order tool calls.

Notably, enabling on-demand rendering yields the strongest improvement on *view-dependent* queries: keeping retrieval+distance fixed (no planning), turning on rendering boosts Dep from 35.7 to 50.0 (+14.3), showing that visual evidence is particularly helpful when language depends on viewpoint-sensitive attributes. In contrast, Indep slightly drops from 50.0 to 46.2, so the overall gain is more modest ($45.0 \rightarrow 47.5$). Using the full toolset achieves the best Overall accuracy (50.0) while balancing Dep (47.6) and Indep (51.3), demonstrating that geometric reasoning, planning, and selective rendering are complementary.

3.5. Discussion

The performance superiority of ***AgentGrounder*** can be attributed to two key factors working in concert.

(1) Tool-driven reasoning with deterministic geometry: Unlike previous fixed anchor-target decomposition and learned reranking, our agent adaptively selected tools per query and ranked candidates via explicit geometric relations from OLT metadata (centers, box sizes). This yielded stable spatial understanding on ScanRefer *Unique* and Nr3D *View-Independent*, where viewpoint-agnostic cues dominate.

(2) Selective rendering and efficient context use: Our agent rendered images only when visual attributes were needed or geometry was ambiguous, and otherwise kept prompts compact by retrieving only query-relevant objects. In contrast, SeeGround (Li et al., 2025) enriched and prompted over a broader set of objects, increasing token overhead and the chance of matching-caused errors. This targeted visual prompts and reduced context load support improved performance on *Hard* and *Multiple* subsets.

A potential confound is that ***AgentGrounder*** uses Qwen3-VL-32B while SeeGround (Li et al., 2025) reports results with Qwen2-VL-72B. We note, however, that ***AgentGrounder*** also surpasses zero-shot pipelines built on much larger proprietary backbones (GPT-4 turbo in LLM-Grounder (Yang et al., 2024) and ZSVG3D (Yuan et al., 2024b), Mistral-Large-240B in CSVG (Yuan et al., 2024a)) while using a base model roughly 2 times smaller than SeeGround's and an order of magnitude smaller than these proprietary alternatives. This indicates that the improvements come predominantly from the agent design rather than from base-model capacity. A fully controlled, VLM-matched comparison is left to future work.

Despite these advances, our method is limited by runtime latency (rendering and LVLM inference) and by failures in the initial 3D instance segmentation, which can propagate to downstream reasoning. Future work might explore human-in-the-loop clarification and adaptive tools (e.g., fallback detectors or re-segmentation heuristics) to improve robustness for real-world robotic deployment.

4. Conclusion

In this paper, we introduced ***AgentGrounder***, a zero-shot framework for 3D visual grounding on colored point clouds. Our method couples a 3D segmentation network with a tool-driven VLM agent. We demonstrated that tools enabling geometric reasoning, scene image generation, and querying an object lookup table can enhance the performance of 3D visual grounding approaches. As evidence, ***AgentGrounder*** achieved state-of-the-art zero-shot results on ScanRefer and Nr3D, with clear gains on view-independent queries and cluttered scenes.

References

Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., and Guibas, L. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European conference on computer vision*, pp. 422–440. Springer, 2020.

Chang, C.-P., Wang, S., Pagani, A., and 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.

Chase, H. Langchain. https://github.com/langchain-ai/langchain, 10 2022. Version accessed via GitHub.

Chen, D. Z., Chang, A. X., and Nießner, M. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pp. 202–221. Springer, 2020.

Chen, S., Guhur, P.-L., Tapaswi, M., Schmid, C., and Laptev, I. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in neural information processing systems*, 35:20522–20535, 2022.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and 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.

Fang, J., Tan, X., Lin, S., Vasiljevic, I., Guizilini, V., Mei, H., Ambrus, R., Shakhnarovich, G., and Walter, M. R. Transcrib3d: 3d referring expression resolution through large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9737–9744. IEEE, 2024.

Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K. M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

Guo, D., Xiang, Y., Zhao, S., Zhu, X., Tomizuka, M., Ding, M., and Zhan, W. Phygrasp: generalizing robotic grasping with physics-informed large multimodal models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 14915–14922. IEEE, 2025.

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.

Jatavallabhula, K. M., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Maalouf, A., Li, S., Iyer, G., Saryazdi, S., Keetha, N., et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.

Jia, B., Chen, Y., Yu, H., Wang, Y., Niu, X., Liu, T., Li, Q., and Huang, S. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, pp. 289–310. Springer, 2024.

Li, R., Li, S., Kong, L., Yang, X., and Liang, J. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3707–3717, 2025.

Marcondes, F. S., Gala, A., Magalhães, R., Perez de Britto, F., Durães, D., and Novais, P. Using ollama. In *Natural Language Analytics with Generative Large-Language Models: A Practical Approach with Ollama and Open-Source LLMs*, pp. 23–35. Springer, 2025.

Ngo, T. D., Hua, B.-S., and Nguyen, K. Isbnet: a 3d point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13550–13559, 2023.

Nguyen, P., Ngo, T. D., Kalogerakis, E., Gan, C., Tran, A., Pham, C., and Nguyen, K. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4018–4028, 2024.

Nguyen, P., Luu, M., Tran, A., Pham, C., and Nguyen, K. Any3dis: Class-agnostic 3d instance segmentation by 2d mask tracking. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3636–3645, 2025.

Qi, Z., Zhang, Z., Fang, Y., Wang, J., and Zhao, H. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv preprint arXiv:2501.01428*, 2025.

Qian, Z., Ma, Y., Lin, Z., Ji, J., Zheng, X., Sun, X., and Ji, R. Multi-branch collaborative learning network for 3d visual grounding. In *European Conference on Computer Vision*, pp. 381–398. Springer, 2024.

Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., and Leibe, B. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8216–8223. IEEE, 2023.

Takmaz, A., Fedele, E., Sumner, R. W., Pollefeys, M., Tombari, F., and Engelmann, F. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023.

Unal, O., Sakaridis, C., Saha, S., and Van Gool, L. Four ways to improve verbo-visual fusion for dense 3d visual grounding. In *European Conference on Computer Vision*, pp. 196–213. Springer, 2024.

Wang, Y., Li, Y., and Wang, S. Gˆ 3-lq: Marrying hyperbolic alignment with explicit semantic-geometric modeling for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13917–13926, 2024.

Wen, C., Huang, Y., Huang, H., Huang, Y., Yuan, S., Hao, Y., Lin, H., Liu, Y.-S., and Fang, Y. Zero-shot object navigation with vision-language models reasoning. In *International Conference on Pattern Recognition*, pp. 389–404. Springer, 2025.

Xu, R., Huang, Z., Wang, T., Chen, Y., Pang, J., and Lin, D. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. In *Conference on Robot Learning*, pp. 3961–3985. PMLR, 2025.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang, J., Chen, X., Qian, S., Madaan, N., Iyengar, M., Fouhey, D. F., and Chai, J. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7694–7701. IEEE, 2024.

Yang, Y., Wu, X., He, T., Zhao, H., and Liu, X. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023.

Yuan, Q., Li, K., and Zhang, J. Solving zero-shot 3d visual grounding as constraint satisfaction problems. *arXiv preprint arXiv:2411.14594*, 2024a.

Yuan, Z., Ren, J., Feng, C.-M., Zhao, H., Cui, S., and 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, 2024b.

Zantout, N., Zhang, H., Kachana, P., Qiu, J., Chen, G., Zhang, J., and Wang, W. Sort3d: Spatial object-centric reasoning toolbox for zero-shot 3d grounding using large language models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2201–2208. IEEE, 2025.

Zheng, D., Huang, S., and Wang, L. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8995–9006, 2025.

A. LLM/Agent Usage Disclosure

During the preparation of this manuscript, we employed several large language models (LLMs) and agent-based tools, including web-based interfaces, search agents, and integrated development environment (IDE) assistants, to support selected stages of the research and writing process.

These tools were used for the following purposes:

- Assisting in the literature review process, including identifying relevant prior work;
- Supporting brainstorming and refinement of high-level ideas, as well as structuring the paper and its sections;
- Improving clarity, readability, and linguistic quality of draft text;
- Suggesting code snippets and assisting with debugging for non-core components (e.g., data processing, visualization, and evaluation scripts);
- Aiding in the summarization of experimental observations and providing consistency checks for interpretations (without generating novel experimental results);
- Providing critical feedback on draft versions of the manuscript.

We did not use LLMs/agents to generate or modify evaluation datasets, produce ground-truth annotations, fabricate results, or create citations without manual verification. We did not provide any private or proprietary data to the tool. Inputs were limited to high-level descriptions and/or our own manuscript drafts.

All technical claims, citations, and experimental results were verified by the authors. All reported numbers come from experiments executed by the authors' code and were checked for consistency.