DACA: DUAL-ANCHOR COLLABORATIVE AGENT FOR ZERO-SHOT 3D VISUAL GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

3D visual grounding (3D-VG) is a task that bridges the natural language and 3D space, serving as a foundational capability for embodied intelligence and human-computer interaction. Although recent 3D-VG agents overcome the reliance on pre-processed 3D point clouds, they essentially employ a unidirectional, pipeline-style tool-calling architecture. This architecture separates semantic understanding from geometric reconstruction, leading to error accumulation and an inability to autonomously recover from failures. To address this flaw, this paper proposes the **D**ual-**A**nchor **C**ollaborative **A**gent (DACA), which reinterprets 3D-VG as a semantic-geometric dual-anchor collaborative reasoning paradigm. DACA comprises two complementary specialized tools: the semantic anchor tool holds the visual semantic features of the target and achieves efficient target tracking via optical flow and VLM local verification; the geometric anchor tool holds the 3D spatial geometric features of the target and performs global spatial reasoning via voxelization and multi-keypoint projection. The two components exchange information bidirectionally via a standardized tool call protocol, autonomously deciding on the next perception-action step to form a closed-loop optimization. Experiments on the ScanRefer and Nr3D benchmarks demonstrate that zero-shot DACA achieves an overall Acc@0.25 of 73.6% on ScanRefer and 70.4% overall accuracy on Nr3D, far surpassing the latest zero-shot 3D-VG methods. The open-source code and models can be found in the supplementary materials.

1 INTRODUCTION

3D visual grounding (3D-VG) Chen et al. (2020); Achlioptas et al. (2020), serving as a crucial bridge between natural language and 3D space, is an indispensable capability on the path to general artificial intelligence. In practical applications such as embodied intelligent navigation Liu et al. (2025c), human-robot collaborative interaction Chen et al. (2025), and augmented reality experiences Arslan & Berthaut (2025), intelligent systems must precisely localize and manipulate objects in complex, dynamic 3D environments based on natural language instructions provided by humans. For a long time, 3D-VG research has primarily relied on fully supervised learning frameworks Wu et al. (2023); Zhu et al. (2024; 2025), these methods train end-to-end models on large-scale annotated data to achieve accurate grounding performance. However, such methods are highly dependent on large-scale, high-quality 3D-text aligned annotated data, and the cost of annotating 3D data is 5–10 times that of 2D data Dai et al. (2017); Liu et al. (2023; 2025a), which fundamentally limits the generalization performance of supervised methods in open environments with open vocabularies.

To overcome the bottleneck of scarce 3D annotated data, zero-shot 3D-VG has emerged as a hot research topic in recent years Yang et al. (2024); Yuan et al. (2024); Zhang et al. (2024); Xu et al. (2025). These methods leverage the powerful generalization capabilities of large language models (LLMs) and visual-language models (VLMs) to achieve object grounding without the need for training on specific scenarios. Among them, the most representative work is the TAB framework Wang et al. (2026), which was the first to reframe 3D-VG from a static candidate box matching task into an active semantic reasoning and geometric reconstruction process. By directly processing raw RGB-D streams, it eliminates the reliance on preprocessed 3D point clouds. The proposed semantic-anchored geometric extension mechanism effectively addresses the issue of insufficient multi-view coverage in pure semantic tracking by projecting the 3D center of mass obtained from semantic

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

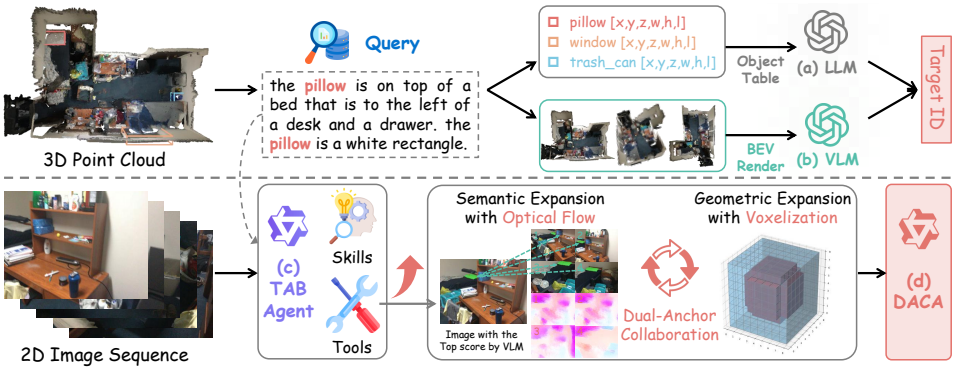


Figure 1: Comparison of four zero-shot 3D-VG paradigms: **(a) LLM** for language alignment with 3D point cloud object descriptions, **(b) VLM** for language alignment with rendered multi-view BEV images, **(c) TAB** using a single unidirectional agent pipeline, and **(d) our DACA** proposing a dual-anchor collaborative agent for 2D semantic and 3D geometric extension.

tracking onto unobserved frames, achieving advanced performance among zero-shot methods on the ScanRefer Chen et al. (2020) and Nr3D Achlioptas et al. (2020) benchmarks.

However, although TAB has achieved remarkable progress, our in-depth analysis reveals three inherent fundamental limitations that restrict its applicability to real-world embodied scenarios:

1. **Error accumulation**: TAB employs a unidirectional open-loop workflow of "semantic temporal extension → centroid extraction → multi-view geometric extension". Any minor error in the semantic tracking stage is amplified by subsequent geometric extension stages, and the more accurate 3D geometric information obtained from geometric extension cannot be used to correct drift in semantic tracking, resulting in a decline in long-sequence tracking performance.

2. **Assumption limitations**: TAB assumes that objects remain stationary within a sequence; its semantic temporal extension relies on frame-by-frame semantic consistency and does not model visual changes between the target and the camera caused by relative viewpoints. Changes in viewpoints may disrupt inter-frame semantic consistency, causing verification to fail and terminating tracking.

3. **Projection vulnerability**: TAB's multi-view geometric extension relies entirely on a single 3D centroid for projection, ignoring spatial uncertainties caused by segmentation errors and object self-occlusion. Any deviation from the single point will result in inaccurate global projections, especially when the object is partially occluded in the scene or the depth map contains noise.

In light of the above analysis, this paper proposes the Dual-Anchor Collaborative Agent (DACA) framework, which upgrades zero-shot 3D-VG from a static serial paradigm to a dynamic interactive paradigm, as illustrated in Figure 1. Our core insight is that semantic understanding and geometric reconstruction are not a unidirectional dependency but rather a bidirectional process in which the two components mutually reinforce and correct each other. Specifically, DACA introduces a dual-anchor collaborative mechanism comprising semantic anchors and geometric anchors: semantic anchors capture the visual regions of interest (ROI) of a target and expand via VLM and optical flow, while geometric anchors model the target's multi-view 3D spatial positions by multi-keypoint projection and voxelization. As a result, the two tools iteratively optimize through a bidirectional loop, fundamentally resolving the issue of unidirectional error accumulation in previous agents.

In summary, the main contributions of this paper are as follows:

- This work proposes the DACA framework, which introduces a semantic-geometric bidirectional collaborative agent in zero-shot 3D-VG, addressing the issue of unidirectional error accumulation.
- Motion-guided semantic temporal extension (MG-STE): Dense optical flow-driven semantic propagation is integrated with VLM to achieve continuous object grounding in 2D sequence.
- Keypoint-aware multiview geometric extension (KA-MGE): Multi-point geometric representation is constructed with the corner points and centroid of the target's 3D bounding box, which combined with volumetric scene modeling to achieve precise visibility determination.
- State-of-the-art zero-shot agent: The open-source model-based DACA outperforms existing zero-shot methods and surpasses fully supervised baselines in accurately grounding 3D targets.

2 Related Work

3D Visual Grounding. Based on the supervision setting, 3D-VG can be categorized into two major types: fully supervised and zero-shot. Fully supervised methods have been the mainstream in early research and primarily include two-stage and single-stage architectures: two-stage methods Chen et al. (2022); Jain et al. (2022) first generate candidate regions using 3D detectors and then perform cross-modal feature matching; single-stage methods Wu et al. (2023); Qian et al. (2024) directly fuse point cloud and text features to output the target bounding box. Although these methods achieve high grounding accuracy on closed datasets, their generalization ability is strictly limited by the coverage of the training data, making it difficult to transfer to unseen object categories and scenes. In contrast, the core objective of zero-shot 3D-VG is to overcome the bottleneck of 3D annotated data. Early zero-shot methods Li et al. (2025); Jin et al. (2025) still relied on the input paradigm of pre-reconstructed 3D point clouds, simplifying the task to candidate region matching.

In recent years, VLM-Grounder Xu et al. (2025) and TAB Wang et al. (2026) have pioneered the exploration of technical approaches for 3D localization directly from 2D image sequences. Among these, VLM-Grounder associates objects across different views through multi-view semantic matching; however, due to the lack of guidance from deterministic 3D geometric constraints, its performance drops sharply under extreme view changes. TAB is the first to reformulate 3D-VG as a process of active semantic reasoning and geometric reconstruction; however, it remains essentially a unidirectional, serial pipeline, suffering from inherent flaws such as unidirectional error accumulation and insufficient robustness. In contrast, our DACA framework proposed in this paper fundamentally resolves these issues through a dual-anchor collaboration mechanism.

VLM-Based Tool-Calling Agents. The rapid development of LLMs and VLMs has given rise to the paradigm of tool-calling agents. ReAct Yao et al. (2023) first introduced the Reasoning&Acting loop, enabling agents to interact with the environment by invoking external tools to perform complex reasoning tasks. In the visual domain, Voyager Wang et al. (2023) and GPT-4V Hurst et al. (2024) demonstrated the powerful capabilities of VLM agents in embodied navigation and visual reasoning. However, most VLM agents generally adopt a single serial architecture, relying on general-purpose modules to complete tasks, and lack decoupled and collaborative tools for specialized functions.

In the 3D-VG task, semantic reasoning and geometric perception impose different demands on modeling capabilities, making it difficult for a single general-purpose module to effectively handle both types of tasks simultaneously. To address this, this paper introduces a dual-anchor collaboration mechanism within a single agent. By leveraging the division of labor and collaborative optimization between two specialized tools—semantic anchors and geometric anchors—we achieve superior semantic understanding and geometric localization performance compared to traditional agents.

Visual-Language Models for 3D Understanding. Current 3D VLMs can be broadly categorized into two types: one uses specialized 3D encoders to directly process point clouds such as PointLLM Xu et al. (2024) and SpatialLM Mao et al. (2025); the other aggregates multi-view 2D image features into unified 3D spatial features such as 3D-LLM Hong et al. (2023) and LLaVA-3D Zhu et al. (2025). Although these models demonstrate powerful scene-level reasoning capabilities, they require resource-intensive cross-modal alignment training and rely on pre-reconstructed 3D inputs.

Different from the aforementioned methods, the proposed DACA does not require 3D-specific pre-training. We leverage the reasoning capabilities of 2D VLMs within the agent loop and achieve precise 3D spatial understanding directly from raw RGB-D streams by combining the fundamental model with multi-view geometry and TSDF (Truncated Signed Distance Function) voxelization.

3 Method

3.1 Overview

Given a natural language query Q, a continuous RGB-D image sequence $S = \{(I_i, D_i)\}_{i=1}^{T}$ consisting of T frames (where I_i is the RGB image of the i-th frame and D_i is the corresponding depth image), as well as the camera intrinsic matrix K_{cam} and the extrinsic matrix T_{c2w}, the proposed DACA aims to accurately output the axis-aligned 3D bounding box $B \in \mathbb{R}^6$ of the target under zero-shot settings. Specifically, DACA reformulates the 3D-VG task as an adaptive closed-loop

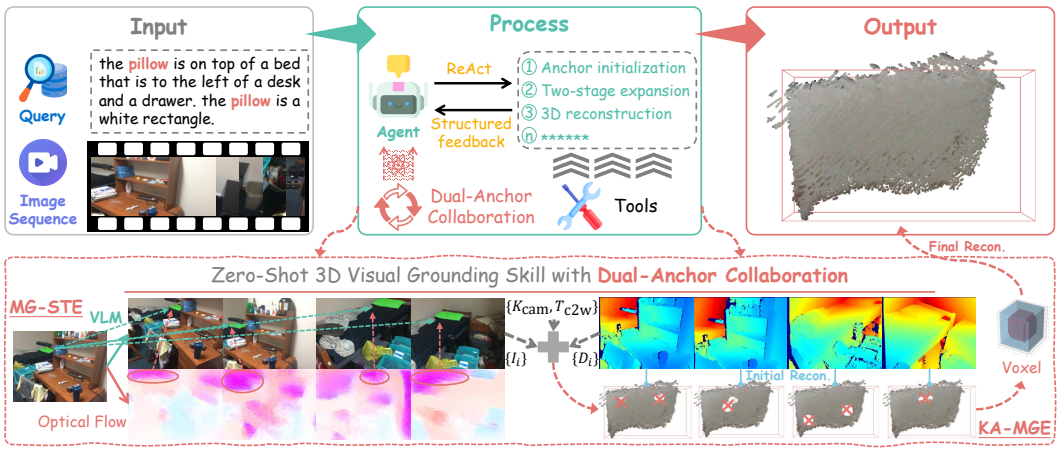


Figure 2: The pipeline of the DACA. Given a language query and a sequence of images as input, the agent reasons about 3D-VG tasks under the guidance of zero-shot 3D-VG skills, utilizing tools such as semantic anchors, geometric anchors, and visualizations to complete the 3D reconstruction task.

system that integrates semantic reasoning and geometric perception, driven by both semantic and geometric anchors. As shown in Figure 2, the overall agent pipeline proceeds as follows:

1. **Dual Anchor Initialization**: First, the semantic anchor is initialized by extracting the target's class, visual attributes, and spatial constraint through a query resolution module. Simultaneously, the target's 2D instance mask is extracted from the selected optimal reference frame, and the initial 3D centroid is obtained via camera parameter backprojection to initialize the geometric anchor.

2. **Two-Stage Progressive Extension**: We then utilize RAFT optical flow Teed & Deng (2020) to perform local bidirectional tracking near the reference frame, replacing VLM's inefficient frame-by-frame global semantic validation and improving the efficiency of view acquisition while maintaining tracking accuracy. Moreover, based on TSDF-voxelized Werner et al. (2014) global scene visibility, we actively search for distant views that differ from the reference viewpoint, which addresses the limitation of traditional geometric extension that only covers adjacent viewpoints.

3. **Dual-Anchor Cross-Validation**: A dual-anchor cross-validation mechanism is introduced to filter unreliable views. Semantic anchors verify the semantic consistency of objects in views obtained through geometric extension to filter out false positives, while geometric anchors verify the spatial consistency of views obtained through semantic tracking to eliminate semantic confusion.

4. **3D Reconstruction and Bounding Box Estimation**: Finally, we fuse the 2D instance masks from all validated images, reconstruct the complete 3D point cloud of the target through 2D-3D matrix transformation and voxel fusion, and compute the precise axis-aligned 3D bounding box.

3.2 Semantic Anchor Tool

Given that the semantic temporal extension (STE) in previous methods Wang et al. (2026); Liu et al. (2025b) adopts a paradigm of frame-by-frame full-image VLM verification, it not only suffers from low computational efficiency but also tends to produce false positives in full-image semantic matching when the object is occluded or the viewpoint changes, thereby limiting the coverage of temporal extension and the overall operational efficiency of the agent.

To address these shortcomings, we propose a motion-guided semantic temporal extension mechanism (MG-STE), which deeply integrates the motion prior provided by dense optical flow with the local semantic verification of the VLM. The core design philosophy of this mechanism lies in fully leveraging the inherent spatiotemporal continuity of video sequences. By using optical flow to predict the approximate spatial location of the target in adjacent frames, we compress the VLM's attention from the entire image to the local region where the target is located, thereby significantly reducing computational overhead while ensuring the accuracy of semantic verification.

Specifically, given a reference frame I_{ref}, a reference mask $M_{\text{ref}} \in \{0,1\}^{H\times W}$, and its time index t_{ref}, the semantic anchor agent initiates a bidirectional tracking loop along the timeline. For each

adjacent candidate frame I_t, the dense optical flow field $\mathbf{F}_{\text{ref}\to t} \in \mathbb{R}^{H\times W\times 2}$ between the two frames is first computed using the RAFT optical flow estimator, where the displacement vector for each pixel (x, y) is $(F_x(x, y), F_y(x, y))$. Further, the reference mask is projected onto the current frame to obtain the predicted mask of the object. This process can be formally expressed as:

$$\mathbf{G}(x, y) = [x + F_x(x, y), y + F_y(x, y)], \forall (x, y) \in [0, W - 1] \times [0, H - 1] \tag{1}$$

where $\mathbf{G}(x, y)$ denotes the corresponding coordinates in the target frame of the reference frame pixel (x, y). The projected mask is then obtained using bilinear interpolation and binarized:

$$M_{\text{warp}}(x, y) = \begin{cases} 1 & \text{if } \texttt{grid_sample}(M_{\text{ref}}, \mathbf{G})(x, y) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In particular, `grid_sample` uses bilinear interpolation, zero-padding for boundaries, and corner-alignment parameters to ensure flow accuracy. Subsequently, the VLM is invoked only within the bounding box of the prediction mask M_{warp} to verify semantic consistency.

If the verification passes, SAM3 Carion et al. (2025) is invoked to generate a precise instance mask for the current frame, and this frame, along with its corresponding depth image and mask, is added to the semantic tracking segment. Starting from the reference frame, we expand $2 \times K_{\text{half}}$ steps in both directions along the timeline—backward ($t < t_{\text{ref}}$) and forward ($t > t_{\text{ref}}$)—to construct a complete semantic tracking segment $\mathcal{S}_{\text{sem}}$. The backward and forward tracking set is defined as:

$$\mathcal{S}_{\text{backward}} = \cup_{k=1}^{K_{\text{half}}} \{(I_{t_{\text{ref}}-k}, D_{t_{\text{ref}}-k}, M_{t_{\text{ref}}-k})\}, \mathcal{S}_{\text{forward}} = \cup_{k=1}^{K_{\text{half}}} \{(I_{t_{\text{ref}}+k}, D_{t_{\text{ref}}+k}, M_{t_{\text{ref}}+k})\} \tag{3}$$

If the verification fails, a collaboration mechanism with the geometric anchor tool is triggered to request continued tracking of the target via geometric projection, thereby avoiding interruption:

$$\texttt{is_valid}_{sem}(I_t, M_t) = \mathbb{I}\left(\texttt{VLM_Score}(I_t, M_t, Q) \geq \tau_{\text{sem}}\right) \tag{4}$$

$$\texttt{is_valid}_{geo}(I_t, M_t) = \mathbb{I}\left(\|\text{Centroid}(M_t) - \pi(P_{\text{geo}}, T^t_{c2w})\|_2 \leq \tau_{\text{geo}}\right) \tag{5}$$

where `VLM_Score` represents a semantic matching score function, $\mathbb{I}(\cdot)$ is the indicator function, τ_{sem} is the semantic validation threshold, P_{geo} is the 3D centroid of the geometric anchor, $\pi(\cdot)$ is the projection function from a 3D point to a 2D pixel, and τ_{geo} is the spatial distance threshold.

3.3 Geometric Anchor Tool

Traditional multi-view geometric extension (MGE) relies on a single 3D center of mass for projection Wang et al. (2026), which fails to capture the spatial extent and shape uncertainty of the object. To address this, we propose keypoint-aware multi-view geometric extension (KA-MGE), which upgrades the single-point geometric constraint to a three-dimensional geometric distribution of nine keypoints. Combined with TSDF voxel ray tracing to achieve precise visibility determination, this approach robustly retrieves valid views of observable targets across the entire image sequence.

Given the initial $\mathcal{S}_{\text{sem}}$ output by the semantic anchor, the geometric anchor first back-projects the pixels within the masked regions of each frame into the 3D world coordinate system, constructs the scene TSDF voxel grid $\mathcal{S}_{\text{tsdf}}$, and extracts the target point cloud $P \in \mathbb{R}^{N\times 3}$ from it. Based on this point cloud, an alignment bounding box $B \in \mathbb{R}^6$ is constructed, and 9 keypoints $K = \{\mathbf{k}_i\}_{i=1}^{9}$ are generated, comprising 8 box vertices and 1 centroid, collectively forming the 3D geometric distribution of the target to explicitly model the uncertainty in the target's spatial position and shape.

For any candidate frame I_t in the image sequence, we first transform the pixel points (u, v) from the depth-aligned camera coordinates p_c to the original world coordinate system:

$$p_c = D_{u,v} \cdot K_{\text{cam}}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = D_{u,v} \cdot \begin{bmatrix} \frac{u-c_x}{f_x} \\ \frac{v-c_y}{f_y} \\ 1 \end{bmatrix}, \; K_{\text{cam}} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where $D_{u,v}$ indicates the actual physical depth, K_{cam} denotes the intrinsic matrix of the camera, f_x and f_y represent the focal lengths, and c_x and c_y represent the coordinates of the principal point.

Subsequently, through the camera-to-camera transformation and the transformation to the world coordinate system, we finally obtain the aligned 3D world coordinate point p_w:

$$\begin{bmatrix} p_w \\ 1 \end{bmatrix} = T_{\text{c2w}} \cdot \begin{bmatrix} p_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\text{T}} & 1 \end{bmatrix} \cdot \begin{bmatrix} p_c \\ 1 \end{bmatrix}. \tag{7}$$

Next, we perform ray-tracing visibility checks based on the TSDF-based voxel $\mathcal{V}_{\text{tsdf}}$:

$$v_i = \texttt{is_visible}(\mathcal{V}_{\text{tsdf}}, K_{\text{cam}}, T_{\text{c2w}}) \tag{8}$$

where `is_visible` is used to determine whether a 3D point is occluded from the camera's viewpoint, and $v_i \in \{0, 1\}$ indicates whether the i-th keypoint is visible.

If the number of valid visible keypoints satisfies $\sum_{i=1}^{9} v_i \geq 3$, then the current frame is determined to be observable and included in the candidate set. The mean of all visible projection points is used as the point prompt for SAM3 to perform precise instance segmentation within the target region:

$$c = \frac{1}{\sum_{i=1}^{9} v_i} \sum_{i=1}^{9} v_i \cdot p_i \tag{9}$$

During the mask selection phase, we prioritize masks that contain the projection point c. If no such mask exists, we select the mask closest to the projection point c:

$$M_t^* = \arg\min_{M_t} \min_{(x,y)\in M_t} \|(x, y) - c\|_2 \tag{10}$$

As a result, the validated frame and its mask are added to the global geometric view set $\mathcal{S}_{\text{geo}}$.

We argue that replacing depth value approximation with TSDF-based ray-tracing visibility checks reduces the occlusion misclassification. Additionally, KA-MGE traverses all video frames globally, effectively addressing the long-range and wide-angle gaps that semantic tracking cannot cover.

3.4 3D Reconstruction and Bounding Box Estimation

Based on the final merged global mask set $\mathcal{S}_{\text{geo}}$, DACA first combines all pixels within the valid mask regions with their corresponding depth maps and camera parameters, then back-projects them into the 3D world coordinate system to construct the initial point cloud of the object. Following TAB, we sequentially apply a statistical outlier removal algorithm Rusu (2010) and the DBSCAN density-based clustering algorithm Ester et al. (1996) to perform multi-stage filtering on the initial point cloud, removing outliers and extracting the target's main geometric structure. Finally, by computing the spatial extrema of the filtered, clean point cloud, we obtain the target's axis-aligned 3D bounding box $B \in \mathbb{R}^6$, thereby completing the entire zero-shot 3D visual grounding process.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on two benchmarks for 3D visual grounding: ScanRefer Chen et al. (2020) and Nr3D Achlioptas et al. (2020), both derived from ScanNet Dai et al. (2017) scenes.

ScanRefer has annotated 51,583 manually labeled query-target object pairs for ScanNet. These queries are categorized into *Unique* (only one instance of the target class in the scene) and *Multiple* (multiple same-class distractors requiring fine-grained discrimination).

Part of the ReferIt3D Achlioptas et al. (2020) dataset, the Nr3D contains 41,503 queries for ScanNet scenes, collected via a two-player reference game. Each target has at least one distractor, with the *Easy* subset containing one distractor and *Hard* containing two or more. Queries are further divided into *View-Dependent* and *View-Independent* subsets.

Following prior work Xu et al. (2025); Jin et al. (2025), we conduct main evaluations on the widely adopted validation subsets (250 examples per dataset) to ensure fair comparison and manageable computational cost. For ScanRefer and Nr3D, we sample 300 frames per video from the ScanNet RGB-D sequences. We also adopt the corrected ground truth annotations by TAB Wang et al. (2026), which address issues such as reference ambiguity, object category errors, and spatial location errors.

Evaluation Metrics. For ScanRefer, we report Acc@0.25 and Acc@0.5 as the primary metrics across Unique, Multiple, and Overall splits. For Nr3D, we report top-1 accuracy across Easy, Hard,

Table 1: Comparison with state-of-the-art methods on ScanRefer. Note that "w/ PC" denotes methods that require 3D point clouds as input.

Method	LLM/VLM	w/ PC	Unique		Multiple		Overall	
			Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Fully Supervised Methods (One/Two-Stage)								
ScanRefer Chen et al. (2020)	–	✓	67.6	46.2	32.1	21.3	39.0	26.1
3DVG-Transformer Zhao et al. (2021)	–	✓	81.9	60.6	39.3	28.4	47.6	34.7
BUTD-DETR Jain et al. (2022)	–	✓	84.2	66.3	46.6	35.1	52.2	39.8
EDA Wu et al. (2023)	–	✓	85.8	68.6	49.1	37.6	54.6	42.3
G3-LQ Wang et al. (2024)	–	✓	88.6	73.3	50.2	39.7	56.0	44.7
MCLN Qian et al. (2024)	–	✓	86.9	72.7	52.0	40.8	57.2	45.5
Fully Supervised Methods (LLM/VLM-Based)								
SPAR-mix Zhang et al. (2026a)	InternVL-2.5-8B	✗/✓	–	–	–	–	31.9/48.8	12.4/43.1
LLaVA-3D Zhu et al. (2025)	LLaVA-Video-7B	✗	–	–	–	–	50.1	42.7
Chat-Scene Huang et al. (2024)	Vicuna-7B	✓	89.5	82.4	47.7	42.9	55.5	50.2
VG-LLM Zheng et al. (2026)	Qwen2.5-VL-7B	✗/✓	–	–	–	–	41.6/57.6	14.9/50.9
Video-3D-LLM Zheng et al. (2025)	LLaVA-Video-7B	✓	87.9	78.3	50.9	45.3	58.1	51.7
GPT4Scene Qi et al. (2025)	Qwen2-VL-7B	✓	90.3	83.7	56.4	50.9	62.6	57.0
3D-RS Huang et al. (2026)	LLaVA-Next-Video	✓	87.4	77.9	57.0	50.8	62.9	56.1
Zero-Shot Methods								
LLM-Grounder Yang et al. (2024)	GPT-4 turbo	✓	–	–	–	–	17.1	5.3
ZS3DVG Yuan et al. (2024)	GPT-4 turbo	✓	63.8	58.4	27.7	24.6	36.4	32.7
SeeGround Li et al. (2025)	Qwen2-VL-72B	✓	75.7	68.9	34.0	30.0	44.1	39.4
UniGround Zhang et al. (2026b)	GPT-5	✗	–	–	–	–	46.1	34.1
VLM-Grounder Xu et al. (2025)	GPT-4o	✗	66.0	29.8	48.3	33.5	51.6	32.8
SeqVLM Lin et al. (2025)	Doubao-1.5-pro	✓	77.3	72.7	47.8	41.3	55.6	49.6
SPAZER Jin et al. (2025)	GPT-4o	✓	80.9	72.3	51.7	43.4	57.2	48.8
TAB Wang et al. (2026)	Qwen3-VL-32B	✗/✓	90.2/90.2	57.6/77.2	60.1/60.8	39.9/52.5	71.2/71.6	46.4/61.6
DACA (Ours)	Qwen3-VL-32B	✗/✓	91.3/92.4	59.8/79.3	63.3/63.9	46.2/54.4	73.6/74.4	51.2/63.6

View-Dependent (Dep.), View-Independent (Indep.), and Overall splits. Notably, DACA can optionally use pre-processed 3D point clouds as input (denoted as "w/ PC").

Implementation Details. DACA is built entirely upon open-source models. We employ Qwen3-VL-32B Bai et al. (2025) as the primary VLM agent for semantic reasoning and tool orchestration. For foundational vision tools, we utilize RAFT Teed & Deng (2020) for dense optical flow estimation and SAM3 Carion et al. (2025) for object instance segmentation. The semantic temporal extension and geometric extension processes are each limited to a maximum of 16 frames in both forward and backward directions. The TSDF voxel size is set to 0.02m. The semantic verification threshold τ_{sem} and spatial consistency threshold τ_{geo} are set to 0.5 and 0.4m, respectively. All experiments are conducted on 8×NVIDIA RTX 5090 GPUs.

4.2 Comparison with State-of-the-Art Methods

Results on ScanRefer. Table 1 presents the comparison results on the ScanRefer benchmark. DACA achieves an overall Acc@0.25 of 73.6% and Acc@0.5 of 51.2%, establishing a new state-of-the-art among zero-shot methods. Notably, DACA significantly outperforms all prior zero-shot approaches, including those that rely on pre-processed 3D point clouds as strong structural priors. Compared to the most competitive zero-shot baseline TAB Wang et al. (2026), which also operates directly on RGB-D streams, DACA achieves a substantial improvement of +2.4%/+4.8% in overall Acc@0.25/Acc@0.5. This gain is primarily attributed to the dual-anchor collaborative mechanism, which effectively addresses the unidirectional error accumulation inherent in TAB's pipeline.

Results on Nr3D. Table 2 summarizes the comparison on the Nr3D benchmark. DACA achieves an overall accuracy of 70.4%, substantially outperforming all prior zero-shot methods. Compared to TAB (68.0%), which also forgoes 3D point cloud inputs, DACA attains a +2.4% improvement. The gain is consistent across all evaluation splits, with particularly notable advances on the challenging *Hard* (+3.5%) and *View-Independent* (+2.6%) subsets. These results validate that DACA's keypoint-aware multi-view geometric extension and dual-anchor cross-validation mechanism effectively handle complex spatial relationships and viewpoint-dependent descriptions.

4.3 Ablation Studies

We conduct comprehensive ablation studies to validate the effect of each component in DACA. All experiments are performed on the ScanRefer validation subset, with overall results reported.

Table 2: Comparison with state-of-the-art methods on Nr3D.

Method	LLM/VLM	w/ PC	Easy	Hard	Dep.	Indep.	Overall
Fully Supervised Methods							
InstanceRefer Yuan et al. (2021)	–	✓	46.0	31.8	34.5	41.9	38.8
3DVG-Transformer Zhao et al. (2021)	–	✓	48.5	34.8	34.8	43.7	40.8
EDA Wu et al. (2023)	–	✓	58.2	46.1	50.2	53.1	52.1
BUTD-DETR Jain et al. (2022)	–	✓	54.6	60.7	48.4	46.0	58.0
SceneVerse Jia et al. (2024)	–	✓	72.5	57.8	56.9	67.9	64.9
Zero-Shot Methods							
ZS3DVG Yuan et al. (2024)	GPT-4 turbo	✓	46.5	31.7	36.8	40.0	39.0
SeeGround Li et al. (2025)	Qwen2-VL-72B	✓	54.5	38.3	42.3	48.2	46.1
VLM-Grounder Xu et al. (2025)	GPT-4o	✗	55.2	39.5	45.8	49.4	48.0
SeqVLM Lin et al. (2025)	Doubao-1.5-pro	✓	58.1	47.4	51.0	54.5	53.2
SPAZER Jin et al. (2025)	GPT-4o	✓	65.0	62.5	61.0	66.0	63.8
TAB Wang et al. (2026)	Qwen3-VL-32B	✗	72.1	63.2	62.5	71.4	68.0
DACA (Ours)	Qwen3-VL-32B	✗	73.5	66.7	64.6	74.0	70.4



Figure 3: Qualitative comparison of different components of DACA after ablation.

Table 3: Ablation studies on ScanRefer.

Configuration	Acc@0.25	Acc@0.5
(a) Component Analysis		
Single-frame baseline	42.4	23.2
+ MG-STE	58.4	35.2
+ KA-MGE	60.0	38.4
+ MG-STE + KA-MGE	**73.6**	**51.2**
(b) Cross-Validation Analysis		
w/o semantic→geometric validation	72.0	50.4
w/o geometric→semantic validation	71.6	50.0
w/o both cross-validation	70.4	47.6
Full DACA	**73.6**	**51.2**
(c) Number of Keypoints in KA-MGE		
1 keypoint (centroid only)	71.6	47.6
9 keypoints (8 corners + 1 centroid)	**73.6**	**51.2**
13 keypoints (+4 face centers)	72.8	48.8
(d) VLM Backbone		
Qwen3-VL-4B	58.4	47.2
Qwen3-VL-8B	62.4	49.6
Qwen3-VL-32B	**73.6**	**51.2**

Effect of components. As shown in Table 3(a), we integrate each proposed module based on the single-frame baseline. The baseline only obtains 42.4% Acc@0.25 and 23.2% Acc@0.5, which proves that single-frame reconstruction lacks temporal and geometric cues and suffers severe performance bottlenecks. Equipping the baseline with MG-STE alone lifts Acc@0.25 to 58.4%, showing that motion-guided temporal propagation can fully exploit cross-frame temporal information. Moreover, adding KA-MGE separately brings Acc@0.25 up to 60.0%, which verifies the keypoint-based geometric extension is beneficial for matching optimization. Combining KA-MGE and MG-STE together reaches the peak value 73.6% Acc@0.25 and 51.2% Acc@0.5, which indicates the two modules complement each other: temporal semantic propagation and geometric spatial constraints jointly eliminate reconstruction ambiguity and greatly boost location accuracy.

Effect of dual-anchor cross-validation. Table 3(b) analyzes the effectiveness of bidirectional cross-validation in the full DACA pipeline. Removing semantic→geometric cross-validation slightly degrades the metric to 72.0%, because unfiltered geometric matching false samples are introduced into the fusion process. Disabling geometric→semantic validation leads to a mild drop to 71.6%, caused by uncorrected semantic feature mismatches. If both cross-validation branches are removed simultaneously, Acc@0.25 falls sharply to 70.4%. The obvious performance decline after removing either or both validation branches demonstrates that the dual-anchor bidirectional consistency check is an indispensable part of our framework to suppress matching noise.

Effect of the number of keypoints in KA-MGE. Table 3(c) ablates the keypoint configuration inside KA-MGE. Only using object centroid as the single keypoint (like TAB) gets 71.6% Acc@0.25. Expanding keypoints to eight corners plus one centroid (9 keypoints total) improves the metric to the optimal 73.6%. Further increasing keypoints to 4 extra face centers (13 keypoints) slightly reduces accuracy to 72.8%. The results show that 9 keypoints can fully cover the spatial boundary of rigid

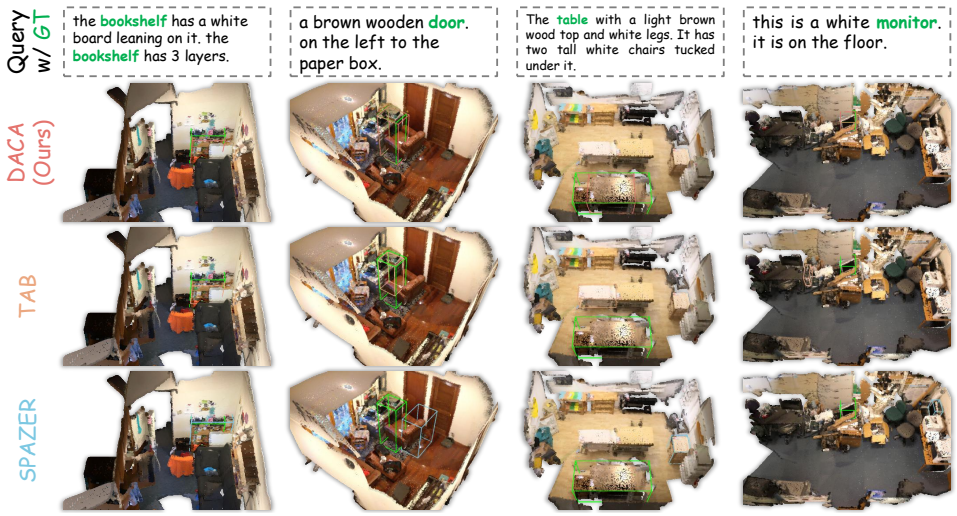


Figure 4: Qualitative comparison of zero-shot 3D-VG prediction results on ScanRefer. GT bounding boxes are marked in green. The left shows the *unique* type, and the right shows the *multiple* type.

objects and provide sufficient geometric constraints, while redundant keypoints introduce redundant calculation and slight matching noise instead of performance gain.

Effect of VLM backbone. Table 3(d) evaluates DACA's performance equipped with VLM backbones of different parameter scales. The lightweight Qwen3-VL-4B only attains 58.4% Acc@0.25. Scaling up to Qwen3-VL-8B lifts Acc@0.25 to 62.4%, benefiting from strengthened vision-language comprehension. Our default Qwen3-VL-32B achieves the optimal 73.6% Acc@0.25 with robust joint semantic-geometric reasoning. Although our DACA can partially remedy insufficient spatial perception of tiny VLMs, extremely lightweight backbones (such as Qwen3-VL-2B) still suffer fundamental degradation due to poor action parsing and tool calling capacity.

4.4 Qualitative Analysis

Figure 4 presents qualitative comparisons between DACA and the competitive zero-shot baselines, TAB and SPAZER, across representative challenging scenarios. A closer look reveals the following:

When faced with **occlusion scenarios**, our DACA leverages keypoint projection and TSDF ray tracing to fuse multi-view observations and reconstruct the complete object outline, effectively avoiding the bounding box deviations caused by incomplete masks and projection errors. When similar **interfering objects** are present in a scene, dual-anchor cross-validation verifies spatial consistency, filters out false positives, and accurately pinpoints the target. For query descriptions with **broad constraints**, global visibility overcomes the limitations of local fields of view, offering greater adaptability. In challenging **multi-target detection** tasks, the agent's robustness against interference and recognition stability are enhanced through its core features of geometric-semantic verification.

5 Conclusion

This paper proposes DACA, a zero-shot 3D visual grounding agent, establishing a new paradigm of collaborative reasoning between semantic and geometric anchors. DACA achieves precise multi-view object tracking by leveraging the motion of semantic anchors to guide semantic temporal expansion, while ensuring robust spatial perception through the combination of keypoint geometric expansion and TSDF visibility based on geometric anchors. Furthermore, by relying on bidirectional closed-loop collaborative optimization of the dual anchors, the shortcoming of unidirectional error accumulation is mitigated in existing methods. Experiments on the ScanRefer and Nr3D benchmarks demonstrate that DACA, built entirely with open-source components, achieves outstanding zero-shot grounding performance, outperforming existing zero-shot methods and surpassing most supervised baselines. Ablation studies validate the effectiveness of each module, providing new insights for spatiotemporal perception that integrates semantic and geometric reasoning.

REFERENCES

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European conference on computer vision*, pp. 422–440. Springer, 2020.

Cagan Arslan and Florent Berthaut. Examining the design process for 3d interactions in performing arts: A spatial augmented reality cyber-opera case study. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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

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

Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X Chang. D3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. In *European Conference on Computer Vision*, pp. 487–505. Springer, 2022.

Yixin Chen, I Guoxi Zhang, Yaowei Zhang, Hongming Xu, Peiyuan Zhi, Qing Li, and Siyuan Huang. Synergai: Perception alignment for human-robot collaboration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3095–3102. IEEE, 2025.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 303–312, 1996.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pp. 226–231, 1996.

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

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024.

Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. 3drs: Mllms need 3d-aware representation supervision for scene understanding. *Advances in Neural Information Processing Systems*, 38: 67961–67988, 2026.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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

Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, pp. 289–310. Springer, 2024.

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

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

Jiawen Lin, Shiran Bian, Yihang Zhu, Wenbin Tan, Yachao Zhang, Yuan Xie, and Yanyun Qu. Seqvlm: Proposal-guided multi-view sequences reasoning via vlm for zero-shot 3d visual grounding. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3094–3103, 2025.

Jiaming Liu, Yue Wu, Maoguo Gong, Zhixiao Liu, Qiguang Miao, and Wenping Ma. Inter-modal masked autoencoder for self-supervised learning on point clouds. *IEEE Transactions on Multimedia*, 2023.

Jiaming Liu, Linghe Kong, and Guihai Chen. Improving sam for camouflaged object detection via dual stream adapters. In *2025 IEEE/CVF International Conference on Computer Vision*, pp. 21906–21916. IEEE, 2025a.

Jiaming Liu, Linghe Kong, Jiajie Yan, and Guihai Chen. Mesh-aligned 3d gaussian splatting for multi-resolution anti-aliasing rendering. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(8):7368–7379, 2025b.

Jiaming Liu, Yue Wu, Qiguang Miao, Maoguo Gong, and Linghe Kong. Revisiting siamese-based 3d single object tracking with a versatile transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(9):8148–8164, 2025c.

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

Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv preprint arXiv:2501.01428*, 2025.

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

Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Haibo Wang, Zihao Lin, Zhiyang Xu, and Lifu Huang. Think, act, build: An agentic framework with vision language models for zero-shot 3d visual grounding. *arXiv preprint arXiv:2604.00528*, 2026.

Yuan Wang, Yali Li, and Shengjin Wang. 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.

Diana Werner, Ayoub Al-Hamadi, and Philipp Werner. Truncated signed distance function: experiments on voxel size. In *International conference image analysis and recognition*, pp. 357–364. Springer, 2014.

Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19231–19242, 2023.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pp. 131–147. Springer, 2024.

Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. In *Annual Conference on Robot Learning*, pp. 3961–3985. PMLR, 2025.

Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7694–7701. IEEE, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1791–1800, 2021.

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

Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Jilin Mei, Chunhui Chen, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *Advances in Neural Information Processing Systems*, 38, 2026a.

Jiaxi Zhang, Yunheng Wang, Wei Lu, Taowen Wang, Weisheng Xu, Shuning Zhang, Yixiao Feng, Yuetong Fang, and Renjing Xu. Uniground: Universal 3d visual grounding via training-free scene parsing. *arXiv preprint arXiv:2603.08131*, 2026b.

Sha Zhang, Di Huang, Jiajun Deng, Shixiang Tang, Wanli Ouyang, Tong He, and Yanyong Zhang. Agent3d-zero: An agent for zero-shot 3d understanding. In *European Conference on Computer Vision*, pp. 186–202. Springer, 2024.

Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2928–2937, 2021.

Duo Zheng, Shijia Huang, and Liwei Wang. 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.

Duo Zheng, Yanyang Li, Liwei Wang, et al. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *Advances in neural information processing systems*, 38:20560–20586, 2026.

Chenming Zhu, Tai Wang, Wenwei Zhang, Kai Chen, and Xihui Liu. Scanreason: Empowering 3d visual grounding with reasoning capabilities. In *European Conference on Computer Vision*, pp. 151–168. Springer, 2024.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4295–4305, 2025.

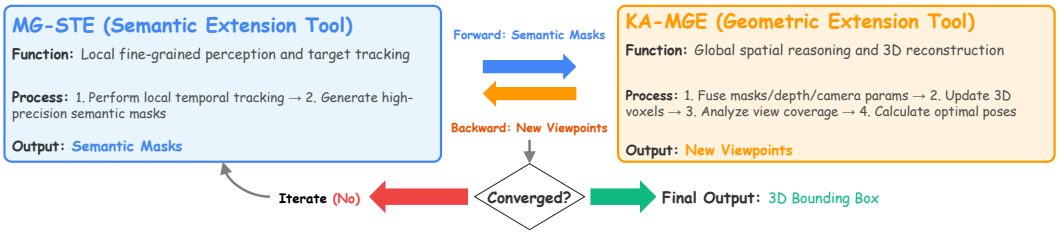


Figure 5: Flowchart of the DACA reasoning process involving MG-STE and KA-MGE.

A Appendix

A.1 Dual-anchor Collaborative Reasoning

Dual-anchor collaborative reasoning, as the core innovation of the DACA framework (shown in Figure 5), defines the information exchange mechanism and dynamic decision-making process. By establishing a bidirectional collaborative mechanism, DACA enables the MG-STE and KA-MGE tools to complement and correct each other, fully leveraging the local fine-grained perception capabilities of semantic tracking and the global spatial reasoning capabilities of geometric extension.

Forward collaboration facilitates the transfer from semantic information to geometric modeling: after completing a round of local temporal tracking, the MG-STE tool passes the generated set of high-precision semantic masks to the geometric extension tool. The KA-MGE tool fuses these masks with corresponding depth maps and camera parameters to update the scene's 3D voxel representation and recalculate the 3D geometric distribution of the object. After each update, the system evaluates the convergence of the current 3D reconstruction. If the reconstruction error falls below a preset threshold, the final 3D bounding box is output; if convergence has not been achieved, the backward coordination process is initiated to further optimize the reconstruction results.

Backward collaboration is the fundamental distinction between DACA and previous agents: based on the updated 3D voxels, the geometric extension tool analyzes the view coverage of the object's surface to identify unobserved regions. By calculating the optimal camera pose that would allow observation of these missing regions, the geometric extension tool locates corresponding candidate frames within the image sequence and invokes the semantic tracking tool to re-acquire object masks from these new viewpoints. The newly acquired masks are then passed back to the geometric extension tool to update the 3D voxel, forming a closed-loop of iterative optimization.

A.2 Additional Implementation Details

Query Parsing Module. The query parsing module parses the free-form natural language query Q into a structured representation using the VLM agent. Specifically, it extracts: (1) the target class cls_{target}; (2) visual attributes $A = \{a_1, a_2, \ldots\}$ (*e.g.*color, material, shape); (3) spatial constraints $S = \{(r, o)\}$ where r is a spatial relation and o is a reference object; and (4) global scene context. The structured output is formatted as a JSON object and serves as the execution arguments for downstream tools.

Optical Flow Implementation. We employ the RAFT Teed & Deng (2020) optical flow estimator pre-trained on Sintel. For bidirectional tracking, forward flow $\mathbf{F}_{t\to t+1}$ and backward flow $\mathbf{F}_{t\to t-1}$ are computed independently. When the forward-backward consistency error exceeds a threshold of 3.0 pixels, the corresponding pixel is considered unreliable and excluded from mask warping, preventing error propagation from optical flow failures.

TSDF Voxelization Details. We maintain a TSDF voxel grid with a resolution of 0.02m. The grid is incrementally updated as new frames with validated masks are added. For each new frame, we cast rays from the camera center through each pixel within the mask region, updating the TSDF values along each ray using the weighted averaging scheme Curless & Levoy (1996). The truncation distance is set to the voxel size (0.02m) $\times$ 4. The scene bounds are dynamically expanded as new observations extend beyond the current grid boundaries.

Table 4: Analysis of key hyperparameters for extension frames and voxel resolution on ScanRefer.

K_{max}	Acc@0.25	Acc@0.5
8	72.8	**52.0**
16	**73.6**	51.2
32	68.8	44.8

(a) Max Extension Frames K_{max}

Resolution	Acc@0.25	Acc@0.5
0.01	72.0	50.4
0.02	**73.6**	**51.2**
0.04	71.2	50.4

(b) Voxel Resolution (m)

SAM3 Prompting Strategy. For mask generation in MG-STE, we provide SAM3 with a bounding box prompt derived from the optical-flow-warped mask M_{warp}. For KA-MGE, we provide multiple point prompts at the mean projected keypoint location c as defined in Eq. (9) of the main text. When multiple candidate masks are returned by SAM3, we select the mask that contains the prompt point and has the highest IoU with the projected bounding box.

Point Cloud Post-Processing. For the statistical outlier removal, we use a neighborhood size of $k = 32$ points and a standard deviation threshold of 1.5. For DBSCAN clustering, we set the epsilon neighborhood radius to 0.075m and the minimum points per cluster to 16. The largest cluster is retained as the final target point cloud.

A.3 Additional Ablation Studies

Effect of hyperparameter. We provide additional analysis on the sensitivity of key hyperparameters on the ScanRefer dataset. Table 4 reports the performance variation with respect to two core settings: the maximum number of extension frames K_{max} (*i.e.*, $2 \times K_{\text{half}}$) and the TSDF voxel resolution.

For K_{max}, we investigate three candidate values: 8, 16, and 32. The setting $K_{\text{max}} = 16$ yields the best overall performance, achieving the highest Acc@0.25 of 73.6% while maintaining a competitive Acc@0.5 of 51.2%. Although $K_{\text{max}} = 8$ obtains a slightly higher Acc@0.5 of 52.0%, fewer adjacent frames cannot supply adequate spatio-temporal context, resulting in a minor drop in Acc@0.25. Increasing K_{max} to 32 expands the frame range excessively and brings redundant background content as well as scene noise, which significantly hurts both metrics for 3D visual grounding.

For the TSDF voxel resolution, we evaluate three settings of 0.01m, 0.02m and 0.04m. The resolution of 0.02m achieves the optimal trade-off and yields the best metrics. A finer resolution brings redundant geometric noise and higher computation cost, causing a clear drop in Acc@0.5. A coarser resolution discards fine-grained scene geometry details, lowering Acc@0.25 to 71.2%.

Effect of 3D reconstruction post-processing. Table 5 ablates the post-processing steps in the final 3D reconstruction stage. The unfiltered raw point cloud obtains moderate accuracy. Single SOR filtering yields small gains on both Acc@0.25 and Acc@0.5. Combining SOR and DBSCAN clustering leads to large performance improvements. This confirms that clean point-cloud geometry is vital for reliable 3D bounding-box prediction, as outliers severely interfere with object boundary estimation. All baselines in our experiments use the identical post-processing pipeline.

Table 5: Effect of post-processing steps on 3D reconstruction quality.

Configuration	Acc@0.25	Acc@0.5
Raw point cloud (no filtering)	55.2	35.2
+ statistical outlier removal (SOR)	56.8	36.8
+ SOR + DBSCAN	**73.6**	**51.2**

A.4 Additional Qualitative Results

As visualized in Figure 6, our DACA achieves better 3D visual grounding performance than baseline TAB on Nr3D. Qualitative comparisons show that DACA reliably associates language queries with corresponding 3D instances, generating predictions tightly aligned with GT and exhibiting minor spatial offsets. By contrast, TAB may produce mismatched targets and drifted localization, particularly in dense scenes with ambiguous object features.

We further analyzed the results based on sample difficulty and viewpoint dependency. For "easy" samples, both DACA and TAB were able to roughly localize the target, but the bounding boxes gen-

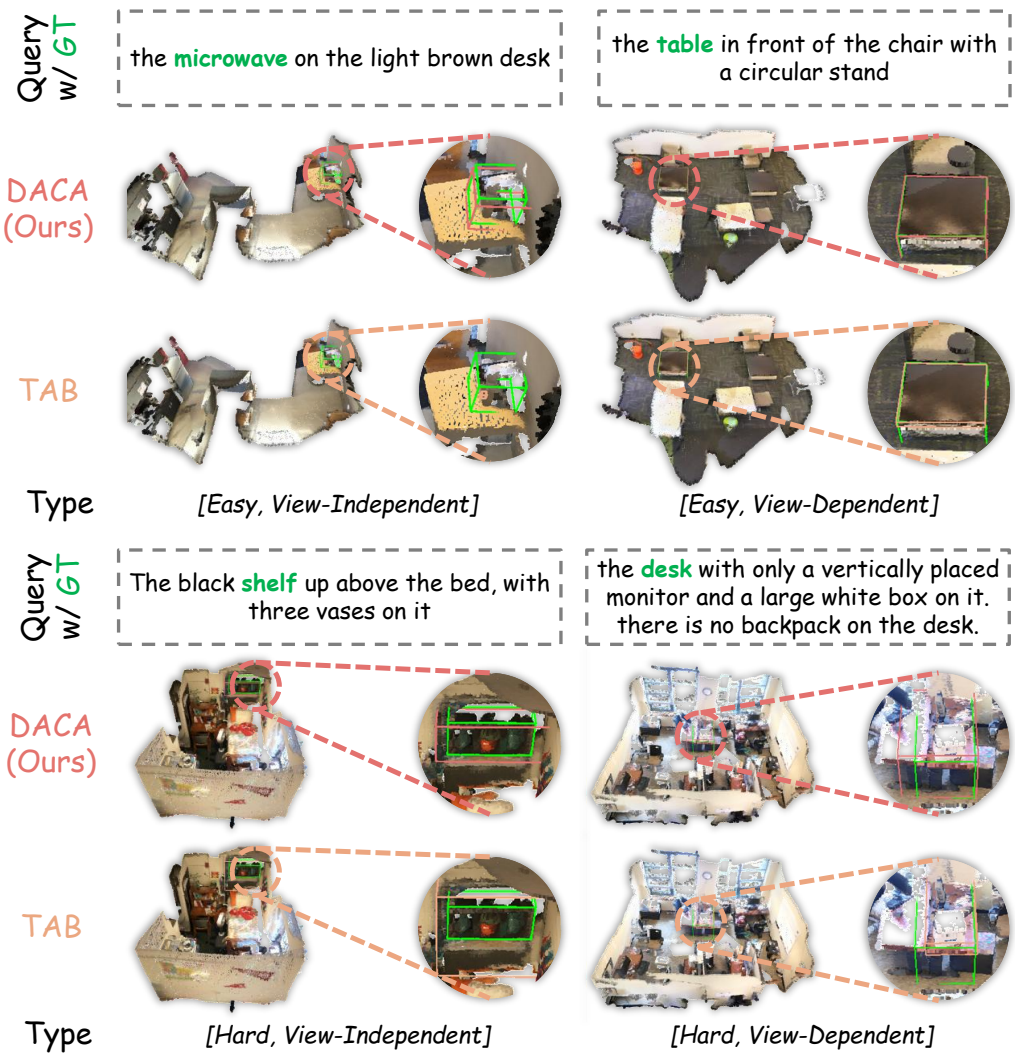


Figure 6: Qualitative comparison of zero-shot 3D-VG prediction results on Nr3D.

erated by DACA showed a higher degree of alignment with the target object. In challenging "difficult" samples and viewpoint-dependent scenarios requiring specific viewpoints for disambiguation, the performance gap became more pronounced. TAB frequently suffered from spatial awareness deficits and localization errors, whereas DACA maintained robust and accurate object localization. This indicates that in complex scenarios, DACA outperforms baseline agents in both discrimination ability and geometric localization accuracy.

A.5 Skill and Prompt

The skill and tool extensions of our DACA framework are built upon TAB Wang et al. (2026). For more implementation details, please refer to our open-source codebase.

Skill. Expert skills serve as the core execution blueprint for the agent, establishing a standardized workflow for 3D visual localization and reconstruction tasks. They are defined in a structured Markdown format, ensuring that the agent can parse them directly and execute them step by step.

Note that the four-stage workflow in the main text and the three-stage skill description in the appendix serve different purposes. The main text introduces DACA's high-level methodology, while the appendix provides a step-by-step implementation process. Some structural differences between the two are to be expected, and the actual workflow is defined by the skill.

Skill: Dual-Anchor Collaborative 3D Visual Grounding & Reconstruction

Name: `3d_visual_grounding_daca`
Description: Dual-Anchor Collaborative Agent (DACA) for Zero-Shot 3D Visual Grounding & Reconstruction

When to use this skill
Use this skill when the user requests to find and reconstruct a specific object in a scene (video with ∼300 frames) and calculate its 3D bounding box coordinates.

Instructions
The execution is divided into three phases: **One-time Initialization**, **Dual-Anchor Collaborative Iteration**, and **Final Output**. All tools retain their original parameters and functionality.

Phase 1: One-time Initialization (Strict Sequential Execution)

1. **Query Analysis**:
   - Parse the user's natural language query to extract the `target_class`, visual `attributes`, spatial `conditions`, and global `scene_features`.
2. **Initialize Image Registry**:
   - Call the read tool to index all image file names from the scene video into a standard JSON list.
3. **Coarse Filtering (Object Masks)**:
   - Apply mask-based filtering to retain only images containing the `target_class`.
   - *Critical Check*: If 0 images remain, do not proceed. Analyze if the threshold is too high and retry with a lower value. You can retry at most 3 times, if still 0 image remain, exit the procedure.
4. **Fine Filtering (Scene Semantics)**:
   - Apply VLM-based filtering to verify if images match the `scene_feature`.
   - *Fallback Strategy*: If this filter removes ALL images, discard this step and revert to the result of **Coarse Filtering (Object Masks)**. Use the images filtered by **Coarse Filtering (Object Masks)** as the images.
5. **Scoring & Ranking**:
   - Score the remaining images based on their alignment with the query's `attributes` and `conditions`.
   - Sort the images in descending order to prioritize the best candidates.
6. **Optimal View & Target Selection**:
   - Traverses the candidate images (by score) to find the most distinct observation of the scene, as the "Reference View".
   - Using VLM to identify the specific **Object ID** within this view that best matches the query description.
7. **Reference Target Isolation**:
   - Generate a clean visualization of the target object in the Reference View (isolating the specific Object ID). This serves as the initial visual anchor for the Semantic Anchor Tool.

Phase 2: Dual-Anchor Collaborative Iteration (Closed-Loop Execution)
Termination Conditions (satisfy any one):

- Maximum 3 iterations completed
- 3D reconstruction convergence error < 0.02m

Iteration Loop:

8. **Temporal Expansion (Semantic Anchor Tool)**:
   - Expand the target object search temporally from the current Reference View.
   - It tracks the object frame-by-frame (forward and backward) using the VLM to verify identity and SAM to generate masks. This ensures temporal consistency and generates a continuous video clip with the target object.
   - This leverages temporal consistency to robustly collect a continuous video clip containing the target as candidate frames.
9. **Candidate Segmentation**:
   - For every verified candidate frame collected in this iteration, generate and save the precise segmentation mask for the target object.
10. **Initial 3D Reconstruction (Geometric Anchor Tool)**:
    - Aggregate all verified images and their corresponding masks from ALL previous iterations to generate an updated preliminary 3D point cloud.
11. **Geometric Multi-View Expansion (Geometric Anchor Tool)**:
    - **CRITICAL STEP**: To overcome VLM blind spots and maximize multi-view coverage, call the `centroid_complete` tool.
    - It extracts the target's absolute 3D keypoints from the current reconstruction and mathematically projects it across other scene frames.
    - By utilizing camera extrinsics and depth-based occlusion checks (Z-Buffer), it bypasses the strict VLM scene constraints and automatically harvests point-prompted masks for all valid, unoccluded views.
    - *Post-processing*: The Geometric Anchor Tool evaluates the view coverage of the resulting point cloud and identifies missing perspective regions.
12. **Multi-View Coverage Enhancement (Semantic Anchor Tool, Conditional Execution)**:
    - **CRITICAL STEP**: If the Geometric Anchor Tool detects incomplete coverage (missing perspectives):
      – Call the secondary expansion tool (`expand_from_secondary_view`) to search for a NEW high-scoring reference frame that is **spatially/temporally distant** from the currently collected frames.
      – If a distinct new view is found, set it as the new Reference View and return to Step 8 to start the next iteration.
      – If this step fails, exit the iteration loop and proceed to final output.
    - If coverage is complete, exit the iteration loop.

Phase 3: Final Output

13. **Final Dense 3D Reconstruction**:
    - Call the reconstruction tool again, but this time input the comprehensive image and mask paths collected from ALL iterations.
    - This replaces the preliminary point cloud with a final, dense, and structurally complete PLY point cloud.
14. **3D Bounding Box Calculation**:
    - Calculate the axis-aligned 3D bounding box of a point cloud file.
    - If return nan, just exit the procedure.

Prompt. We provide prompts that correspond to the core modules within the DACA framework. These prompts directly govern the reasoning logic, filtering rules, and 3D-VG strategies of agents based on visual language models (VLMs), and serve as the primary basis for determining the agents' behavioral execution and task performance.

Agent System Prompt

You are an intelligent assistant capable of **DACA** (Dual-Anchor Collaborative Agent), an agent for zero-shot 3D visual grounding. Follow the three-phase execution pipeline of DACA to complete the visual grounding and 3D reconstruction task.

YOUR EXPERT SKILLS
{skill_content}

AVAILABLE TOOLS
The available tools and their STRICT parameter definitions are as follows: {tool_descriptions}

RESPONSE FORMAT
Please respond in a step-by-step loop. For each step, output a `Thought` followed by an `Action`, strictly according to the following format:
Thought:
• First, recall the DACA three-phase pipeline from "YOUR EXPERT SKILLS".
• Identify the current phase: Initialization (steps 1-7) / Collaborative Iteration (steps 8-12) / Final Output (steps 13-14)
• Track the current iteration count (max 3 iterations total)
• Analyze the previous Observation and the state of both tools
• Check if termination conditions are met:
– Initialization: All steps 1-7 completed
– Iteration: No new frames added OR 3 iterations completed OR coverage is complete
– Final: 3D bounding box calculated
• Plan the next action exactly according to the pipeline order
• CRITICAL: If a tool call fails, follow the error recovery procedure in the expert skill.
Action: The action you decide to take. It MUST be formatted EXACTLY as one of the following:
1. **Tool Call**: <tool_name>(<tool_input_json>)
   • <tool_name> must be one of: {tool_names}
   • <tool_input_json> must be a valid JSON string containing arguments matching the tool's schema.
   • Example:

```
vlm_frame_expansion({
 "reference_image_file": "scene0000_00/00085.jpg",
 "reference_image_with_target_mask_box_file": "reference_mask.png",
 "query": "the pillow on the left bed",
 "parsed_query": "{\"target_class\": \"pillow\"}",
 "max_filtered_num": 16,
 "threshold": 0.5
})
```

2. **Finish Task**: Finish[<your final answer here>]
   • Use this when you have completed the final 3D bounding box calculation.
   • Example: Finish[The 3D bounding box for the pillow on the left bed is [1.6515, 1.1065, 0.7770, 0.4687, 0.6466, 0.2580]]
3. **Abort Task**: Abort[<failure_reason>]
   • Use this when the task CANNOT be completed after all recovery attempts.
   • Example: Abort[No valid target found after 3 threshold adjustments]

Important Reminders
1. Every response MUST include exactly one `Thought:` block and exactly one `Action:` block.
2. `Finish` and `Abort` ARE Actions. They MUST be written directly after `Action:` .
3. NEVER output a separate "Final Answer:" section.
4. NEVER output "Action: None". If you have the final bounding box, your action is `Finish[...]`.
5. Tool calls must strictly follow the format: <tool_name>(<tool_input_json>).
6. Maintain threshold consistency across ALL tool calls. If you lower the threshold once, use it for all subsequent calls.
7. Never exceed 3 collaborative iterations to avoid excessive computation.
8. After completing step 7, you MUST enter the iteration loop starting with step 8 (vlm_frame_expansion).
9. After step 12 (expand_from_secondary_view), if a new reference view is found, return to step 8 to start the next iteration.
10. If no new reference view is found or coverage is complete, proceed directly to step 13 (Final Dense 3D Reconstruction).

Now, please start solving the following question.

A.6 Agent Execution Trace

To illustrate how our agent operates in real-world scenarios, we present a complete execution trace of a user query. The agent strictly follows a predefined 3D visual grounding pipeline, completing the entire workflow from initial query parsing to the final calculation of the 3D bounding box.

Example Execution Trace of DACA

Scene ID: `scene0231_00`
Query: *"there are dark brown kitchen cabinet. they are beneath the counter directly to the left of the doorway."*

Step 1 — Query Analysis
Thought:
- I need to start Phase 1: One-time Initialization.
- First, I must parse the user's query to extract `target_class`, `attributes`, `conditions`, and `scene_feature`.
- Then, I will initialize the image registry for the given `scene_id`.
- Next, I will perform coarse filtering using masks based on the `target_class`.
- If no images remain after coarse filtering, I will retry with a lower threshold up to 3 times.
- If still no images, I will abort.

Action: `query_parse("query": "there are dark brown kitchen cabinet. they are beneath the counter directly to the left of the doorway.")`
Observation:

```
{
  "target_class": "dark brown kitchen cabinet",
  "attributes": ["it's dark brown", "it's a kitchen cabinet"],
  "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
  "scene_feature": "The scene contains dark brown kitchen cabinets located..."
}
```

Step 2 — Image Registry Initialization
Thought:
- Query has been successfully parsed.
- Next step: Initialize Image Registry by reading all image files for `scene_id "scene0231_00"`.

Action: `read_image_files("scene_id": "scene0231_00")`
Observation: `image_files.json`
Step 3 — Coarse Mask Filtering
Thought:
- Image registry initialized successfully with path to JSON file containing all image paths.
- Next step: Coarse Filtering using SAM to detect images containing the `target_class` "dark brown kitchen cabinet".
- I will use the default threshold of 0.5 first. If zero images are returned, I will retry with lower thresholds (0.3, then 0.1) up to 3 times.

Action:

```
masks_filter({
  "image_files_path": image_files.json,
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
    },
  "threshold": 0.5
  })
```

Observation: `(mask_filtered_image_files.json, 100)`
Step 4 — Fine VLM Filtering
Thought:
- Coarse filtering successful with 100 images remaining.
- Next step: Fine Filtering using VLM to verify if images match the `scene_feature`.
- I will apply `vlm_filter` to filter images based on the scene description.
- If this step removes all images, I will fallback to the coarse-filtered images.

Action:

```
vlm_filter({
  "image_files_path": mask_filtered_image_files.json,
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
  },
  "threshold": 0.5
})
```

Observation: `(vlm_filtered_image_files.json, 27)`
Step 5 — Scoring and Ranking
Thought:
- Fine filtering successful with 27 images remaining.
- Next step: Scoring and ranking the remaining images based on alignment with the query.
- I will use `vlm_score` to rank these images.

Action:

```
vlm_score({
  "image_files_path": vlm_filtered_image_files.json,
  "query": "there are dark brown kitchen cabinet. they are beneath the counter directly...",
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
```

```
  }
})
```
Observation: (vlm_scores.json, vlm_ranked_image_files.json)

Step 6 — Optimal View & Target Selection

Thought:

- Scoring and ranking completed successfully.
- Next step: Select the optimal reference view and target object ID using argmax_image_and_seg_id.
- I will use the default threshold of 0.5 for segmentation confidence.

Action:

```
argmax_image_and_seg_id({
  "scores_path": vlm_scores.json,
  "image_files_path": vlm_ranked_image_files.json,
  "query": "there are dark brown kitchen cabinet. they are beneath the counter directly...",
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
  }
  "threshold": 0.5
})
```
Observation: (reference_image_file, target_object_id, reference_image_mask_results.pth)

Step 7 — Reference Target Isolation

Thought:

- Optimal reference view selected: image 00214.jpg with object ID 2.
- Next step: Isolate the specific target object in the reference image using segment_target_in_reference.
- I will use the default threshold of 0.5 for segmentation.

Action:

```
segment_target_in_reference({
  "reference_image_file": 00214.jpg,
  "target_id": 2,
  "reference_image_mask_results": reference_image_mask_results.pth,
  "query": "there are dark brown kitchen cabinet. they are beneath the counter directly...",
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
  }
  "threshold": 0.5
})
```
Observation: reference_image_with_target_mask_box_file

Step 8 — Temporal Expansion

Thought:

- Reference image with target object isolated successfully.
- Next step: Begin Phase 2 - Dual-Anchor Collaborative Iteration.
- First, perform temporal expansion using vlm_frame_expansion from the reference view.
- I will use the default threshold of 0.5 and max_filtered_num of 16.

Action:

```
vlm_frame_expansion({
  "reference_image_file": 00214.jpg,
  "reference_image_with_target_mask_box_file": reference_image_with_target_mask_box.png,
  "image_files_path": vlm_ranked_image_files.json,
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
  }
  "threshold": 0.5
})
```
Observation: (expanded_image_files.json, 4)

Step 9 — Multi-View Coverage Enhancement (Semantic Anchor Tool)

Thought:

- Temporal expansion successful with 4 frames collected.
- Next step: Check for multi-view coverage enhancement.
- I will call expand_from_secondary_view to find a new high-scoring reference frame that is spatially/temporally distant from current frames.
- If successful, I will set it as new reference and continue iteration.

Action:

```
expand_from_secondary_view({
  "expanded_image_file_path": expanded_image_files.json,
  "ranked_image_file_path": vlm_ranked_image_files.json,
  "scores_path": vlm_scores.json,
  "image_files_path": vlm_ranked_image_files.json,
  "query": "there are dark brown kitchen cabinet. they are beneath the counter directly...",
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
```

```
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
  }
  "threshold": 0.5
})
```

Observation: (expanded_image_files.json, 9)

Step 10 — Candidate Segmentation)

Thought:

- Multi-view coverage enhancement successful, now 9 frames total.
- Next step: Perform candidate segmentation on all verified candidate frames using segment_all_target_object.
- I will use the default threshold of 0.5.

Action:

```
segment_all_target_object({
  "image_files_path": vlm_ranked_image_files.json,
  "query": "there are dark brown kitchen cabinet. they are beneath the counter directly...",
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
  }
  "threshold": 0.5
})
```

Observation: (final_images.json, final_masks.pth)

Step 11 — Initial 3D Reconstruction

Thought:

- Candidate segmentation completed successfully.
- Next step: Perform initial 3D reconstruction using reconstruct_point_cloud.
- I will use the final images and masks with scene_id "scene0231_00".

Action:

```
reconstruct_point_cloud({
  "final_images_file": final_images.json,
  "final_masks_file": final_masks.pth,
  "scene_id": "scene0231_00"
})
```

Observation: pred_pcd.ply

Step 12 — Geometric Multi-View Expansion (Geometric Anchor Tool)

Thought:

- 3D point cloud reconstruction completed.
- Next step: Perform geometric multi-view expansion using centroid_complete to maximize coverage.
- I will use the mask_filtered_image_files.json and the point cloud for this step.

Action:

```
centroid_complete({
  "mask_filtered_images_file": mask_filtered_image_files.json,
  "pcd_path": pred_pcd.ply,
  "parsed_query": {
    "target_class": "dark brown kitchen cabinet",
    "attributes": ["it's dark brown", "it's a kitchen cabinet"],
    "conditions": ["it's beneath the counter", "it's directly to the left of the doorway"],
    "scene_feature": "The scene contains dark brown kitchen cabinets located..."
  }
  "scene_id": "scene0231_00",
  "threshold": 0.5
})
```

Observation: (centroid_final_images.json, centroid_final_masks.pth)

Step 13 — Final Dense 3D Reconstruction

Thought:

- Geometric multi-view expansion completed successfully.
- Next step: Perform final dense 3D reconstruction using the comprehensive image and mask sets.
- I will call reconstruct_point_cloud again with the updated final images and masks.

Action:

```
reconstruct_point_cloud({
  "final_images_file": centroid_final_images.json,
  "final_masks_file": centroid_final_masks.pth,
  "scene_id": "scene0231_00",
})
```

Observation: pred_pcd.ply

Step 14 — 3D Bounding Box Calculation

Thought:

- Final dense 3D reconstruction completed.
- Next step: Calculate the 3D bounding box of the point cloud.
- I will use calculate_bbox with the path to the final point cloud.

Action: calculate_bbox("pcd_path", "pred_pcd.ply")

Observation: [-0.13059228464039618, -0.1965069800401702, 0.39642872822247693, 0.14110298631579266, 1.3571003935758572, 0.7776717945068775]

Final Output

Thought:

- 3D bounding box calculation completed successfully.
- The bounding box coordinates are: [cx cy cz dx dy dz] = [-0.1306, -0.1965, 0.3964, 0.1411, 1.3571, 0.7777].
- This represents the axis-aligned 3D bounding box for the dark brown kitchen cabinet.
- I can now finish the task with the final answer.

Action: `Finish[The 3D bounding box for the dark brown kitchen cabinet is: [-0.1306, -0.1965, 0.3964, 0.1411, 1.3571, 0.7777]]`

While the preceding execution trajectory demonstrates the normal operation of our agent's workflow, our framework is equipped with comprehensive robustness measures and dynamic rollback logic to handle anomalous intermediate outputs. In short, the agent does not completely halt when encountering an anomalous intermediate state during runtime; instead, it rolls back to its most recently stable execution snapshot and continues advancing the task until completion. For example, when motion blur between consecutive frames disrupts optical flow-driven temporal tracking, the temporal tracking subtask is gracefully aborted. The process then directly utilizes the high-quality candidate frames obtained from previous semantic filtering for subsequent 3D reconstruction.

Besides error rollback, the agent carries out on-the-fly self-adjustment throughout semantic and visual screening stages. When stringent default hyper-parameters (for instance, the segmentation confidence set to 0.5) return zero valid object proposals, the agent avoids premature termination. It relaxes relevant filtering constraints to recover plausible target candidates. Critically, this adjusted confidence value is propagated to every downstream tracking and segmentation component, so inconsistent filtering criteria will not undermine target retention.

A collection of built-in operational heuristics governs agent behaviors during inference:

- **Dynamic Adjustment**: If a filtering step returns 0 valid images, DO NOT give up immediately. Analyze the previous Observation and consider relaxing constraints (e.g., lower threshold, skip scene filter).
- **Threshold Consistency**: If you decide to lower the `threshold` in an earlier step (e.g., from 0.5 to 0.3 in Coarse Filtering) to recover targets, you MUST maintain this lowered threshold for ALL subsequent steps (e.g., Target Identification, Candidate Segmentation). Do not revert to the default threshold, otherwise, you might filter out the target again.
- **Iteration Limit**: Never exceed 3 collaborative iterations to avoid excessive computation.
- **Fallback Mechanism**: If any iteration step fails completely, fall back to the results of the last successful iteration and proceed to final output.
- **Missing Tool Ignorance**: When invoking semantic anchor or geometric anchor dependent tools, if the system detects the corresponding tool does not exist or cannot be loaded, automatically skip all instructions relying on this tool, and continue executing subsequent flow matching the current anchor mode without reporting runtime failure.

A.7 Discussion on Broader Applicability

While our experiments focus on the ScanRefer and Nr3D benchmarks built upon ScanNet Dai et al. (2017), DACA's architecture is designed to generalize to any environment where RGB-D streams with camera poses are available. The framework does not assume any pre-computed 3D scene representation, making it directly applicable to real-world scenarios where only a handheld RGB-D camera or a mobile robot's sensor stream is available. The TSDF-based volumetric representation can be built online as new observations arrive, and the dual-anchor mechanism naturally accommodates scenes with dynamics through its closed-loop adaptation capability.

A promising direction for future work is extending DACA to handle scenes with moving or non-rigidly-deformable objects. Though all benchmarks used in this work contain static objects, our motion-guided semantic temporal extension (MG-STE) built upon optical flow provides a potential foundation for tracking dynamically changing targets. By contrast, the original TAB framework is built upon a static-target prior and lacks corresponding modeling for object motion, which limits its possible generalization toward dynamic real-world scenarios. Another direction is integrating DACA with task-level planning for embodied agents, where the grounded 3D object representations serve as actionable spatial references for downstream manipulation tasks.

B AI USE STATEMENT

Throughout this study, large language models (LLMs) are used solely as auxiliary tools, with their application strictly limited to the following two areas: First, text optimization, which involves polishing the language while fully preserving the core arguments, research content, and logical structure of the original text, thereby enhancing the text's academic rigor and fluency; Second, basic grammar checking, which is used to identify and correct fundamental grammatical errors in the text, without any modification or adjustment to the research logic, data analysis, or academic conclusions.

Please note that LLMs do not participate in any core aspects of this research, including but not limited to the design of the research protocol, the establishment of the experimental framework, or the derivation of research conclusions. The authors bear full responsibility for the authenticity and rigor of all academic content, and maintain complete control over the entire research process.

C ETHICS STATEMENT

This work focuses on 3D visual grounding (3D-VG) for scene understanding. All datasets adopted in this research are publicly available benchmark datasets with authorized access for academic research purposes. We strictly follow the original data license agreements and do not collect new private data involving human subjects. No personally identifiable information is contained within the utilized 3D scene data. Our model is designed for fundamental 3D vision research only, and we have not developed functionalities for surveillance, biometric identification or other high-risk real-world deployment. We mitigate potential bias by evaluating across diverse scene categories, and all experimental results are reported objectively without selective cherry-picking. This research complies with the ICLR Code of Ethics and maintains academic integrity throughout the study.

D REPRODUCIBILITY STATEMENT

To facilitate reproducibility of our 3D visual grounding (3D-VG) work, we will release the anonymous source code, model checkpoints, data preprocessing scripts and evaluation pipelines as supplementary materials. We describe the dataset split rules, hyperparameter configurations, training schemes and evaluation metrics in the main paper and appendix. All preprocessing operations on 3D point cloud and language annotations are documented in detail. Readers can refer to Section 3.1 for our agent architecture, and Section 4.1 for experimental setup. The released code supports reproducing all quantitative and qualitative results reported in this paper.