RENDER-AND-VERIFY FOR TRAINING-FREE HYPOTHESIS VERIFICATION IN 3D VISUAL GROUNDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised 3D visual grounding (3DVG) has achieved substantial progress, yet its performance remains highly uneven across scene complexity. Modern grounders reliably localize targets when the referred object is unique, but their accuracy drops sharply when multiple same-class distractors are present. We find that many errors involving same-class distractors cannot be resolved by object appearance or grounder confidence alone. Resolving such ambiguities often requires observing the candidate together with surrounding objects, and the usefulness of these relational cues can depend on the viewpoint. Based on this observation, we propose RaV (Render-and-Verify) framework, a training-free plug-in that improves a frozen 3D grounder by verifying its candidate hypotheses with a frozen 2D vision-language model (VLM). Rather than replacing the 3D grounder, RaV preserves its geometric localization capability and utilizes the VLM only for candidate verification. Specifically, RaV first constructs diverse hypotheses from multiple transformed views. For each candidate, Visibility-constrained Canonical Framing (VCF) selects a candidate-conditioned viewpoint that maintains target visibility while preserving the surrounding context required for relational verification. Each rendering is cleaned by a frozen diffusion corrector, then verified by the VLM and re-ranked accordingly. We evaluate RaV on ScanRefer with TSP3D, PV-Ground, and EG-3DVG, and on Nr3D and Sr3D with TSP3D and PV-Ground. RaV improves the Overall accuracy of every evaluated configuration on ScanRefer and achieves state-of-the-art results on all three benchmarks under the detection protocol. The source code will be made publicly available upon acceptance.

1 INTRODUCTION

3D visual grounding (3DVG) aims to localize an object in a 3D scene according to a natural-language expression, providing an important interface between language understanding and physical environments. Since the introduction of ScanRefer (Chen et al., 2020), supervised 3DVG has progressed from proposal-based matching and relation modeling (Zhao et al., 2021; Yuan et al., 2021; Roh et al., 2022) to single-stage grounding (Luo et al., 2022; Jain et al., 2022) and stronger vision-language alignment and contextual reasoning (Wu et al., 2023; Yang et al., 2023; Wang et al., 2024; Xu et al., 2024a; Qian et al., 2024; Gu et al., 2025). More recently, several methods (Guo et al., 2025; Shang et al., 2026; Park et al., 2026) have further improved efficiency and grounding accuracy through sparse processing and geometry-aware decoding. As a result, contemporary grounders localize targets with high reliability when the referred object is unique within a scene.

However, this progress remains considerably less pronounced when multiple instances of the same semantic class are present. On ScanRefer, recent supervised grounders exceed 90% accuracy on the Unique subset (*i.e.*, only a single instance of the target class exists), whereas performance remains around 51–55% on the Multiple subset (*i.e.*, multiple instances of the target class exist). This gap reflects a central difficulty of 3D grounding. When several chairs, tables, or cabinets coexist, recognizing the semantic category alone is insufficient, and the target must instead be distinguished through its attributes and relations to surrounding objects. Existing methods have explicitly modeled object relations (Zhao et al., 2021; Yang et al., 2023), multiple attributes (Xu et al., 2024a), and viewpoint information (Shi et al., 2024). Nevertheless, a substantial performance gap remains between unique-object cases and scenes containing same-class distractors.

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.

We argue that an important source of errors involving same-class distractors lies in the visual evidence available for candidate discrimination. Same-class instances often exhibit similar appearance, while referring expressions distinguish them through contextual relations, such as the *chair beside the window* or *the pillow on the left sofa*. These relations are difficult to determine from an isolated candidate representation because the surrounding objects that define the expression must also be observed. Moreover, the usefulness of this contextual evidence can depend on viewpoint, as occlusion and projected spatial arrangement may make the same relation clear from one viewpoint but ambiguous from another (Shi et al., 2024). Consequently, a grounder may include the correct object among its candidate hypotheses but still rank another instance higher when the contextual evidence required for discrimination is not sufficiently available.

Recent studies (Yuan et al., 2024; Li et al., 2025; Xu et al., 2024b) have begun to leverage 2D VLMs for 3D visual grounding through rendered scene views, registered RGB observations, visual reasoning programs. These approaches demonstrate semantic and relational reasoning, but primarily utilize the VLM as the grounder, requiring it to identify the target from the scene. This formulation does not fully exploit the complementary strength of a supervised 3D grounder, which already provides accurate geometric localization and retains the correct target among a small set of plausible hypotheses. In contrast, we preserve the 3D grounder for hypothesis generation and utilize the VLM only to verify which candidate best satisfies the referring expression.

Motivated by this observation, we propose Render-and-Verify (RaV) framework, a training-free verification framework for 3D Visual Grounding (3DVG). RaV keeps both the pretrained 3D grounder and the 2D VLM frozen and introduces zero trainable parameters. We first construct a diverse candidate pool by combining the grounder's native predictions with hypotheses obtained under multiple yaw-transformed views. Since existing grounders are not perfectly invariant to scene orientation, these transformed predictions expose alternative candidates that may not appear in a single forward pass. For each candidate, we introduce Visibility-constrained Canonical Framing (VCF), which selects a candidate-conditioned viewpoint that maintains target visibility while preserving the surrounding scene context required for relational verification. The rendering is refined with a single-step diffusion corrector (Wu et al., 2025), to our knowledge the first artifact correction in 3DVG. Finally, the candidate is highlighted and presented to the frozen VLM for independent binary verification. The resulting affirmative probabilities are used to re-rank the candidate hypotheses.

We evaluate RaV with three representative supervised grounders, TSP3D (Guo et al., 2025), PV-Ground (Shang et al., 2026), and EG-3DVG (Park et al., 2026), on ScanRefer (Chen et al., 2020) dataset, including both single- and two-stage variants of the latter two. We further evaluate TSP3D and PV-Ground on Nr3D and Sr3D (Achlioptas et al., 2020) datasets. RaV improves Overall accuracy for all evaluated configurations on ScanRefer, with gains of up to $+4.60$ Acc@0.25, and achieves state-of-the-art performance on all three benchmarks under the detection protocol. Our contributions are summarized as follows:

- We propose RaV, a training-free Render-and-Verify framework that preserves a frozen supervised 3D grounder for candidate generation and uses a VLM for candidate verification.
- We introduce VCF and an artifact-corrected candidate rendering pipeline that exposes each hypothesis together with the surrounding spatial context required for verification.
- Extensive experiments across three grounders, multiple VLM scales and families, and three benchmarks show that external hypothesis verification consistently improves supervised 3D grounding without additional training.

2 Related Work

2.1 Supervised 3D Visual Grounding

Supervised 3D visual grounding has been extensively studied by improving how language and 3D scene representations are learned and aligned. Early approaches rely on proposal-based matching and contextual object representations (Chen et al., 2020; Yuan et al., 2021; Roh et al., 2022), while subsequent methods introduce explicit relation modeling (Zhao et al., 2021), single-stage grounding (Luo et al., 2022; Jain et al., 2022), and dense vision-language alignment (Wu et al., 2023). Recent studies further strengthen the grounding architecture by modeling contextual relations ex-

plicitly (Yang et al., 2023), semantic-geometric interactions (Wang et al., 2024), and multiple fine-grained object attributes (Xu et al., 2024a), and viewpoint information (Shi et al., 2024). More recent grounders improve feature extraction and decoding through sparse voxel processing (Guo et al., 2025), point-voxel interaction (Shang et al., 2026), and geometry-aware attention (Park et al., 2026). Multi-view information has also been incorporated within grounding architectures. MVT (Huang et al., 2022) aggregates features from rotated scene views, while ViewRefer (Guo et al., 2023) fuses multi-view cues with view-dependent language. Despite their different designs, these approaches primarily improve grounding by modifying the representation, alignment, or reasoning process within the supervised model. However, these approaches still require the grounder to both generate plausible hypotheses and resolve ambiguities among them, leaving potentially useful alternative predictions unexplored once the top-ranked output is selected.

In contrast, our proposed RaV introduces an external verification stage outside the pretrained grounder. Unlike prior multi-view methods that fuse information within the grounder, RaV uses orientation disagreement only to diversify hypotheses and resolves them with an external verifier. It treats the frozen grounder's predictions as candidate hypotheses, which are examined through viewpoint selection, artifact-corrected rendering, and VLM-based verification. Therefore, RaV complements existing supervised grounders without modifying their architecture or training objective.

2.2 VLM-BASED 3D VISUAL GROUNDING

Recent advances in LLM and VLM have enabled alternative formulations of 3D grounding that leverage pretrained foundation models for semantic and relational reasoning. ZSVG3D (Yuan et al., 2024) decomposes referring expressions into structured reasoning programs and combines language-model reasoning with geometric information. VLM-Grounder (Xu et al., 2024b) instead performs visual grounding over registered RGB observations using a VLM and subsequently lifts the predicted target into 3D. These approaches demonstrate that pretrained foundation models can support 3D grounding without conventional task-specific 3DVG training. More recent methods construct visual representations or observations that can be processed by 2D VLMs. SeeGround (Li et al., 2025) generates query-conditioned rendered views and asks a VLM to identify the referred object from visually marked candidates. SeqVLM (Lin et al., 2025) performs proposal-guided reasoning over multi-view image sequences, while PanoGrounder (Jung et al., 2026) constructs panoramic scene representations to expose broader spatial context. IVSGround (Chiang et al., 2026) further studies candidate-specific viewpoint selection and comparative VLM grounding. Despite their different representations and reasoning strategies, these approaches integrate the foundation model into the grounding process itself, where it participates in identifying or selecting the target. However, a trained 3D grounder already provides geometrically plausible hypotheses, while lacking an external mechanism to independently verify them using complementary visual reasoning.

To address this limitation, RaV reformulates VLM reasoning as external verification over the predictions of a frozen supervised 3D grounder. Rather than asking the VLM to perform 3D grounding itself, RaV constructs candidate-specific visual evidence through Visibility-constrained Canonical Framing (VCF), exposing each hypothesis together with the surrounding context required for relational verification. The frozen VLM independently verifies these hypotheses, enabling RaV to refine existing supervised grounders without modifying the underlying model.

3 METHOD

3.1 OVERALL FRAMEWORK

We introduce RaV, a training-free verification framework that improves a pretrained 3D visual grounder without modifying its parameters. As illustrated in Fig. 1, RaV first constructs diverse candidate hypotheses from a frozen grounder, renders each candidate from a candidate-conditioned viewpoint, refines the rendering using a frozen diffusion corrector, and verifies the candidate with a frozen VLM for re-ranking. All components remain frozen, introducing zero trainable parameters.

Given a 3D scene S and a referring expression q, RaV constructs N candidate hypotheses from the predictions of a frozen 3D grounder:

$$\mathcal{B} = \{(b_i, s_i)\}_{i=1}^{N}, \tag{1}$$

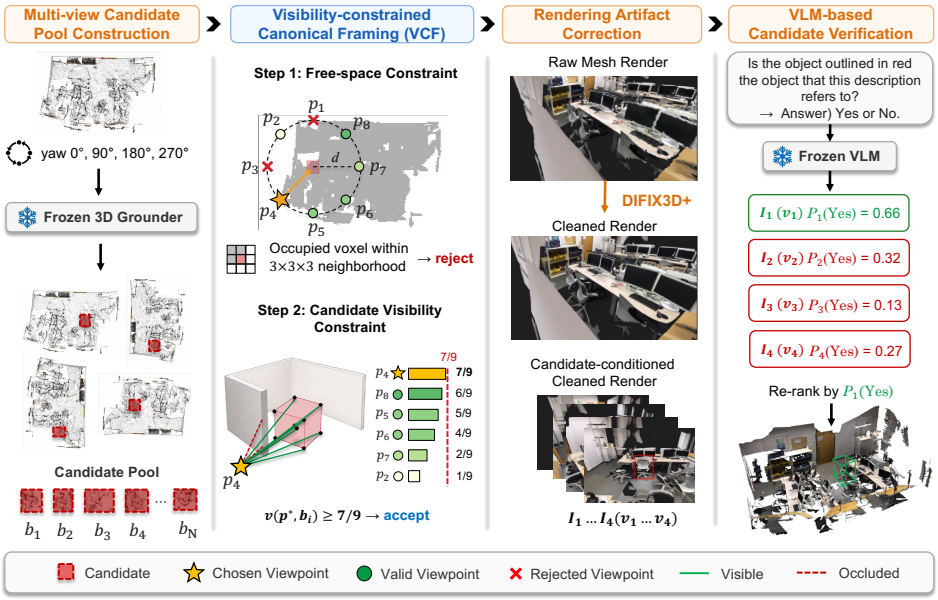


Figure 1: **Overall framework of the proposed RaV.** RaV preserves a 3D grounder for hypothesis generation and utilizes VCF-based rendering to provide contextual evidence for VLM verification.

where b_i is a 3D bounding box and $s_i \in \mathbb{R}$ is its associated grounder score. For each candidate b_i, RaV generates a candidate-conditioned image I_i and obtains a verification probability $P_i(\text{Yes})$ as:

$$P_i(\text{Yes}) = V(I_i, q), \tag{2}$$

where V denotes the frozen VLM. The final prediction is obtained by re-ranking $\mathcal{B}$ using these verification scores while retaining the original grounder prediction when the VLM provides insufficient evidence for replacement. Please refer to the following sections for more details.

3.2 Multi-view Candidate Pool Construction

Verification can only recover the correct target when it is included among the candidate hypotheses. We therefore augment the native grounder predictions with hypotheses obtained from multiple scene orientations. Given a scene point cloud S, we construct yaw-transformed scenes as:

$$S^{(r)} = T_{\theta_r}(S), \qquad \theta_r = 90°r, \quad r \in \{0, 1, 2, 3\}, \tag{3}$$

where T_{θ_r} denotes rotation around the gravity axis. Let G denote the frozen 3D visual grounder. The same grounder is applied to each transformed scene:

$$\mathcal{B}^{(r)} = G(S^{(r)}, q). \tag{4}$$

All predicted boxes are mapped back to the original coordinate system using $T_{\theta_r}^{-1}$. The candidate pool depends on the output structure of the grounder. For grounders with ranked proposals, we retain the native top-4 predictions from the original scene together with the top-1 prediction from each remaining orientation, yielding at most seven candidates. For grounders that output a single box, we retain the top-1 prediction from all four orientations. In both cases, b_1 denotes the original prediction and serves as the default hypothesis during re-ranking. For hybrid pools, spatially redundant candidates are removed using rank-preserving 3D IoU de-duplication with $\tau_{\text{dup}} = 0.5$, retaining the highest-scoring prediction among overlapping hypotheses. Single-box pools retain all four orientation-specific predictions.

The yaw transformations are used to increase candidate diversity rather than to average predictions. Thus, disagreement across orientations provides alternative hypotheses while leaving the underlying grounder unchanged. Unlike multi-view grounding architectures (Huang et al., 2022; Guo et al., 2023) that fuse rotated views inside the model, or standard test-time augmentation that averages

predictions (Shanmugam et al., 2021), RaV keeps the disagreeing hypotheses separate and resolves them by verification. Finally, if all candidate pairs satisfy $\mathrm{IoU}_{3D}(b_i, b_j) \geq 0.7$, we return the original top-ranked prediction without rendering or VLM verification, since the pool contains no sufficiently distinct alternative.

3.3 Visibility-constrained Canonical Framing

Reliable candidate verification requires both target visibility and sufficient surrounding context to interpret the referring expression. A fixed scene-level viewpoint may occlude the candidate or obscure spatial relations needed to distinguish it from same-class distractors. Therefore, we introduce Visibility-constrained Canonical Framing (VCF), which selects a candidate-conditioned viewpoint while accounting for camera feasibility and target visibility.

Given a candidate box $b_i = (\mathbf{c}_i, \mathbf{e}_i)$ with center $\mathbf{c}_i = (c_{i,x}, c_{i,y}, c_{i,z})$ and extent $\mathbf{e}_i$, VCF samples $A = 8$ camera poses on a horizontal orbit around the candidate as:

$$\theta_a = \frac{2\pi a}{A}, \qquad \mathbf{p}_a = (c_{i,x} + d\cos\theta_a,\ c_{i,y} + d\sin\theta_a,\ h_{\mathrm{cam}}), \quad a \in \{0, \ldots, A-1\}, \tag{5}$$

where we set $d = 2.5$ m and $h_{\mathrm{cam}} = 1.7$ m. Each camera is oriented toward the candidate center, with the gravity axis defining the vertical direction. The fixed radius avoids conditioning the framing scale on potentially inaccurate candidate extents.

Note that rather than rendering all sampled poses, VCF evaluates viewpoint validity directly in 3D. We construct a 5 cm scene occupancy grid by marking voxels containing at least one reconstructed mesh vertex, and use it to evaluate camera feasibility and candidate visibility before rendering.

Free-space Constraint. Let $\mathrm{vox}(\mathbf{p}_a)$ denote the voxel containing camera position $\mathbf{p}_a$, and $\mathcal{N}_{27}(\mathrm{vox}(\mathbf{p}_a))$ its $3 \times 3 \times 3$ neighborhood. A pose is considered free-space valid when

$$O(\mathbf{v}) = 0, \qquad \forall \mathbf{v} \in \mathcal{N}_{27}\big(\mathrm{vox}(\mathbf{p}_a)\big), \tag{6}$$

where $O(\mathbf{v}) \in \{0, 1\}$ denotes voxel occupancy. This removes camera positions that lie inside or immediately adjacent to reconstructed geometry.

Candidate Visibility Constraint. For each free-space-valid pose, we evaluate visibility using nine reference points consisting of the box center and its eight corners as:

$$\mathcal{P}_i = \{\mathbf{c}_i\} \cup \left\{ \mathbf{c}_i + \frac{1}{2}\mathbf{s} \odot \mathbf{e}_i \;\middle|\; \mathbf{s} \in \{-1, +1\}^3 \right\}, \tag{7}$$

where $\odot$ denotes element-wise multiplication. A ray is cast from the camera to each reference point through the occupancy grid. Voxels inside the candidate box, dilated by one voxel, are excluded from the occlusion test to avoid treating the candidate itself as an occluder.

Let $r_{a,j} \in \{0, 1\}$ indicate whether the ray from $\mathbf{p}_a$ to the j-th reference point in $\mathcal{P}_i$ reaches the candidate without intersecting other occupied voxels. The visibility score is defined as follows:

$$v(\mathbf{p}_a, b_i) = \frac{1}{9} \sum_{j=1}^{9} r_{a,j}. \tag{8}$$

Among the free-space-valid viewpoints, VCF selects the camera pose with the highest visibility:

$$\mathbf{p}^* = \underset{\mathbf{p}_a,\, a \in \mathcal{A}_{\mathrm{free}}}{\arg\max}\ v(\mathbf{p}_a, b_i). \tag{9}$$

where $\mathcal{A}_{\mathrm{free}}$ denotes the set of viewpoint indices satisfying the free-space constraint. The selected pose is accepted when $v(\mathbf{p}^*, b_i) \geq \frac{7}{9}$, and the search terminates early if a fully visible pose with $v(\mathbf{p}_a, b_i) = 1$ is found.

If no free-space-valid pose satisfies the visibility threshold, VCF selects the valid pose with the highest visibility. If none of the eight sampled viewpoints satisfies the free-space constraint, we apply a final fallback over additional camera distances using candidate visibility and projected image coverage without enforcing the free-space constraint. Thus, VCF returns a rendering viewpoint for every candidate. The selected viewpoint $\mathbf{p}^*$ is then used to render the reconstructed ScanNet mesh once at 1024×576 resolution with a 70° vertical field of view. We use flat shading and uniform ambient illumination with back-face culling disabled to preserve indoor surfaces and surrounding context. The candidate annotation is added after artifact correction, as described in Sec. 3.4.

3.4 RENDERING ARTIFACT CORRECTION

Although mesh rendering provides a direct interface between 3D hypotheses and a 2D VLM, reconstructed meshes often contain holes, incomplete surfaces, and texture artifacts. These distortions introduce a domain gap relative to VLM pretraining images and can obscure object boundaries, which prior VLM-based grounders leave uncorrected (Jung et al., 2026).

To reduce this degradation, we employ Difix3D+ (Wu et al., 2025), a pretrained diffusion-based image restoration model, as a frozen rendering corrector. Given the raw rendering I_i^{raw}, the corrected image is obtained as follows:

$$I_i^{\mathrm{cor}} = D_{\mathrm{fix}}\left(I_i^{\mathrm{raw}}\right), \tag{10}$$

where D_{fix} denotes the Difix3D+ model. We use a single diffusion step with the restoration prompt "*remove degradation*". No parameter of the correction model is optimized for 3D visual grounding.

Importantly, artifact correction is applied before candidate annotation. Applying the diffusion model after annotation could modify or remove the visual marker used by the VLM. We first restore the rendered scene and subsequently project the candidate's 3D bounding box into the image plane. Let $\{\mathbf{x}_{i,j}\}_{j=1}^{8}$ denote the eight corners of b_i. Given the projection function $\pi_{\mathbf{p}^*}(\cdot)$ of the selected camera, the projected candidate region is defined as:

$$\mathcal{R}_i = \mathrm{BBox}_{2D}\left(\{\pi_{\mathbf{p}^*}(\mathbf{x}_{i,j})\}_{j=1}^{8}\right), \tag{11}$$

where points outside the visible camera frustum are discarded. We draw a red rectangular marker around $\mathcal{R}_i$, producing the final verification image I_i. The same marker representation is used for all candidates and grounders.

3.5 VLM-BASED CANDIDATE VERIFICATION

Given the candidate-conditioned image I_i and referring expression q, RaV evaluates each candidate independently rather than jointly comparing multiple candidates. Each candidate is queried with the same binary prompt, "*Is the object outlined in red the object that this description refers to?*". This formulation allows the VLM to assess the candidate using both its appearance and surrounding spatial context. The complete prompt and additional details are provided in Appendix A.1.8.

We obtain the verification score directly from the first-token logits of the frozen VLM rather than generating an autoregressive response. Let ℓ_i^{yes} and ℓ_i^{no} denote the logits corresponding to the affirmative and negative tokens. The verification probability for candidate i is computed as:

$$P_i(\mathrm{Yes}) = \frac{\exp(\ell_i^{\mathrm{yes}})}{\exp(\ell_i^{\mathrm{yes}}) + \exp(\ell_i^{\mathrm{no}})}. \tag{12}$$

This formulation provides a deterministic scalar verification score for every candidate and avoids ambiguity arising from free-form VLM responses.

Directly selecting the candidate with the highest verification probability can degrade the original prediction when the VLM provides weak evidence. Thus, we adopt a re-ranking strategy that treats the original top-1 grounder prediction as the default hypothesis. Let b_1 denote the original prediction and $m = \arg\max_{i \neq 1} P_i(\mathrm{Yes})$ the strongest alternative. The final prediction is determined as:

$$\hat{b} = \begin{cases} b_m, & P_m(\mathrm{Yes}) - P_1(\mathrm{Yes}) > \tau, \\ b_1, & \text{otherwise}, \end{cases} \tag{13}$$

where τ denotes the verification margin. The margin requires an alternative hypothesis to receive sufficiently stronger VLM evidence before replacing the original grounder prediction. For ranked-proposal grounders (*i.e.*, PV-Ground and EG-3DVG), we apply the margin rule above using the original top-1 prediction as the default. For TSP3D, which provides one prediction per orientation, we directly use $\hat{b} = b_{\arg\max_i P_i(\mathrm{Yes})}$. The dataset-specific margin is shared across ranked-proposal grounders and verifier scales, with sensitivity reported in Appendix A.3.2.

The complete RaV procedure preserves the localization capability of a supervised 3D grounder while introducing additional visual evidence only at the candidate-selection stage. The grounder, artifact corrector, and VLM remain frozen throughout the entire pipeline, and no gradient-based optimization is required.

Table 1: **Performance comparison on the ScanRefer dataset.** "Ours" denotes our framework plugged into existing 3D visual grounding baselines.

Method	**Unique** (∼19%)		**Multiple** (∼81%)		**Overall**	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
Two-Stage Model						
ScanRefer (3D+2D)	76.33	53.51	32.73	21.11	41.19	27.40
3DVG-Transformer	81.93	60.64	39.30	28.42	47.57	34.67
BUTD-DETR	82.88	64.98	44.73	33.97	50.42	38.60
EDA	85.76	68.57	49.13	37.64	54.59	42.26
M3DRef-CLIP	85.30	77.20	43.80	36.80	51.90	44.70
CORE-3DVG	84.00	67.09	51.82	39.76	56.77	43.83
MCLN	86.89	72.73	51.96	40.76	57.17	45.53
VGMamba	**91.90**	79.60	54.80	<u>47.60</u>	60.00	<u>53.90</u>
PV-Ground	87.67	72.94	54.99	43.11	59.87	47.56
EG-3DVG	91.10	**85.00**	52.90	46.70	58.50	52.40
PanoGrounder	84.30	–	55.30	–	61.00	–
Ours (PV-Ground)	89.36	73.85	**57.44**	44.70	**62.20**	49.05
Ours (EG-3DVG)	<u>91.61</u>	<u>84.71</u>	<u>56.66</u>	**49.71**	<u>61.87</u>	**54.93**
Single-Stage Model						
3D-SPS	81.63	64.77	39.48	29.61	47.65	36.43
BUTD-DETR	81.47	61.24	44.20	32.81	49.76	37.05
EDA	86.40	69.42	48.11	36.82	53.83	41.70
MCLN	84.43	68.36	49.72	38.41	54.30	42.64
TSP3D	87.25	71.41	51.04	42.37	56.45	46.71
PV-Ground	86.11	72.45	54.61	43.44	59.31	47.77
EG-3DVG	<u>90.63</u>	<u>81.89</u>	51.18	44.44	57.07	50.03
Ours (TSP3D)	88.37	72.59	56.03	<u>46.17</u>	60.86	<u>50.12</u>
Ours (PV-Ground)	87.10	73.15	**56.78**	44.63	<u>61.31</u>	48.89
Ours (EG-3DVG)	**91.68**	**82.80**	<u>56.41</u>	**48.70**	**61.67**	**53.79**

Bold: The best, <u>Underline</u>: The second-best (within each group)

4 EXPERIMENT

4.1 IMPLEMENTATION DETAILS

We plug RaV into TSP3D (Guo et al., 2025), PV-Ground (Shang et al., 2026), and EG-3DVG (Park et al., 2026), using single-stage and two-stage variants of the latter two under the detection protocol. For candidate verification, we use Qwen2.5-VL-7B and Qwen2.5-VL-72B (Bai et al., 2025) without additional training. The re-ranking margin is fixed per dataset, with $\tau = 0.1, 0.2, 0.4$ for ScanRefer, Nr3D, and Sr3D, respectively. Additional implementation details are provided in Appendix A.4.

4.2 PERFORMANCE COMPARISON

Comparison on ScanRefer. We compare RaV with representative 3D visual grounding methods. For two-stage models, we include ScanRefer (Chen et al., 2020), 3DVG-Transformer (Zhao et al., 2021), BUTD-DETR (Jain et al., 2022), EDA (Wu et al., 2023), M3DRef-CLIP (Zhang et al., 2023), CORE-3DVG (Yang et al., 2023), MCLN (Qian et al., 2024), VGMamba (Zhu et al., 2025b), PV-Ground, EG-3DVG, and Panogrounder (Jung et al., 2026). For single-stage models, we include 3D-SPS (Luo et al., 2022), BUTD-DETR, EDA, MCLN, TSP3D, PV-Ground, and EG-3DVG.

Table 1 compares RaV with existing 3D visual grounding methods on ScanRefer. RaV improves the Overall accuracy of all evaluated base grounders and achieves new state-of-the-art performance in both single- and two-stage settings. In the two-stage setting, RaV reaches 62.20 Overall Acc@0.25 with PV-Ground and 54.93 Acc@0.5 with EG-3DVG, surpassing the previous best results by $+2.20$ and $+1.03$ points, respectively. In the single-stage setting, RaV with EG-3DVG improves its base model by $+4.60/+3.76$ in Overall Acc@0.25/Acc@0.5. The gains are particularly pronounced on the Multiple subset. For example, RaV improves TSP3D from 51.04 to 56.03 and EG-3DVG from 51.18 to 56.41 in Multiple Acc@0.25, while also improving Unique by $+1.12$ and $+1.05$, respectively. As shown in Fig. 2, EG-3DVG confuses same-class instances when the target is distinguished primarily by directional or positional relations. RaV resolves these failures by exposing the relevant spatial context through VCF-selected views and verifying the resulting candidates with the VLM.

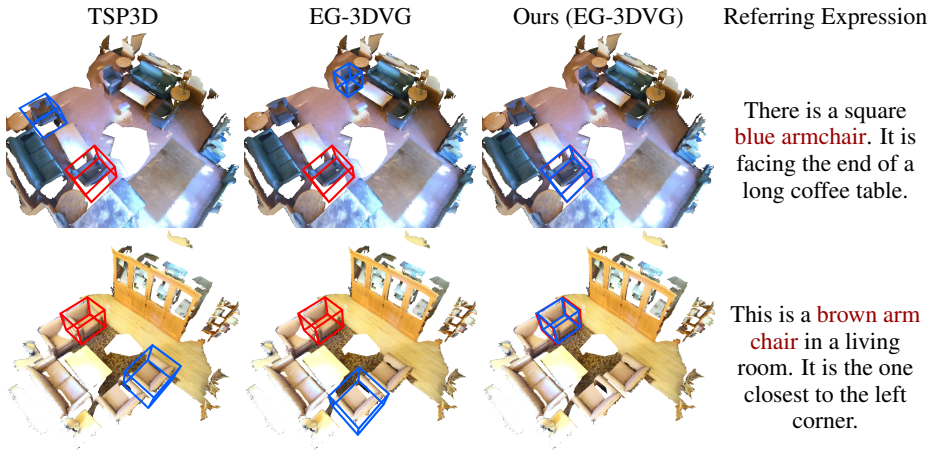


Figure 2: **Qualitative comparison of 3D visual grounding results on the ScanRefer dataset.** The red box denotes the ground-truth object and the blue box denotes the model prediction.

Table 2: **Generalization across base grounders and Qwen2.5-VL scales on the ScanRefer dataset.** We report Overall Acc@0.25 / Acc@0.5 with performance gain.

Grounder	Pipeline	Base	Ours	
			7B	72B
TSP3D	Single-stage	56.45 / 46.71	59.86 / 49.42 (+3.41 / +2.71)	60.86 / 50.12 (+4.41 / +3.41)
PV-Ground	Single-stage	59.31 / 47.77	60.94 / 48.64 (+1.63 / +0.87)	61.31 / 48.89 (+2.00 / +1.12)
	Two-stage	59.87 / 47.56	61.65 / 48.47 (+1.78 / +0.91)	62.20 / 49.05 (+2.33 / +1.49)
EG-3DVG	Single-stage	57.07 / 50.03	60.38 / 52.88 (+3.31 / +2.85)	61.67 / 53.79 (+4.60 / +3.76)
	Two-stage	58.50 / 52.40	61.24 / 54.33 (+2.74 / +1.93)	61.87 / 54.93 (+3.37 / +2.53)

Table 3: **Performance comparison on Nr3D and Sr3D under the detection protocol with Acc@0.25.** We exclude the "two-stage with ground-truth proposals" setting, which replaces detected proposals with ground-truth boxes and assumes a perfect proposal stage, making it not directly comparable to the detection setting.

Method	Venue	**Nr3D**	**Sr3D**
Two-Stage Model (detected boxes)			
InstanceRefer (Yuan et al., 2021)	ICCV'21	29.9	31.5
LanguageRefer (Roh et al., 2022)	CoRL'22	28.6	39.5
BUTD-DETR (Jain et al., 2022)	ECCV'22	43.3	52.1
EDA (Wu et al., 2023)	CVPR'23	40.7	49.9
MCLN (Qian et al., 2024)	ECCV'24	46.1	53.9
Single-Stage Model			
3D-SPS (Luo et al., 2022)	CVPR'22	39.2	47.1
BUTD-DETR (Jain et al., 2022)	ECCV'22	38.7	50.1
EDA (Wu et al., 2023)	CVPR'23	40.0	49.7
MCLN (Qian et al., 2024)	ECCV'24	45.7	53.4
TSP3D (Guo et al., 2025)	CVPR'25	48.7	57.1
PV-Ground (Shang et al., 2026)	CVPR'26	51.3	56.2†
Ours (TSP3D)	–	53.1	**57.8**
Ours (PV-Ground)	–	**53.3**	56.4

†: Value from the official pretrained model

Generalization across Base Grounders. Table 2 evaluates RaV across TSP3D, PV-Ground, and EG-3DVG with two Qwen2.5-VL scales. Both 7B and 72B consistently improve all five grounder configurations, with Acc@0.25 gains of $+1.63$ to $+3.41$ for 7B and $+2.00$ to $+4.60$ for 72B. These results show that RaV generalizes across grounders and single-/two-stage pipelines. Larger gains with the Qwen2.5-VL-72B suggest stronger verification, while consistent improvements across different VLM models demonstrate robustness across verifier families in Appendix A.1.4.

Table 4: **Cumulative ablation of the proposed pipeline on the ScanRefer validation set.** We start from the single-stage TSP3D baseline and progressively add each component.

VLM	VCF	Difix3D+	Overall		Multiple		Δ
			Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	
–			56.45	46.71	51.04	42.37	–
7B			56.92	46.83	51.61	42.52	(+0.47 / +0.12)
7B		✓	57.38	47.20	52.05	42.89	(+0.93 / +0.49)
7B	✓		59.29	48.67	54.26	44.51	(+2.84 / +1.96)
7B	✓	✓	<u>59.86</u>	<u>49.42</u>	<u>54.95</u>	<u>45.30</u>	(+3.41 / +2.71)
72B	✓	✓	**60.86**	**50.12**	**56.03**	**46.17**	(+4.41 / +3.41)

Δ: gain in Overall Acc@0.25 / Acc@0.5, **Bold**: The best, <u>Underline</u>: The second-best

Table 5: **Comparison of candidate selection strategies on the ScanRefer Multiple subset.** All methods re-rank the same 4-yaw candidate pool from a frozen TSP3D grounder. Recovered (%) denotes the net Acc@0.25 gain over the top-1 prediction as a fraction of the oracle headroom.

Evidence	Selection method	Recovered (%)
Prediction-level	Grounder confidence	2.0
	Learned cross-view selector	1.3
Feature-level	DINOv2 fusion (4-view) (Oquab et al., 2023)	1.5
	SigLIP learned (Zhai et al., 2023)	1.9
	SigLIP zero-shot (Zhai et al., 2023)	2.1
Context-level	**RaV (Ours), 72B verifier**	**31.9**

Results on Nr3D and Sr3D. Table 3 reports results on Nr3D and Sr3D (Achlioptas et al., 2020) under the detection protocol. On Nr3D, RaV improves TSP3D from 48.7 to 53.1 and PV-Ground from 51.3 to 53.3, establishing the best reported Acc@0.25. On Sr3D, RaV with TSP3D improves from 57.1 to 57.8 and achieves the best performance, while PV-Ground shows a modest improvement from 56.2 to 56.4. Overall, RaV remains effective across multiple 3D grounding benchmarks, with larger gains when the candidate pool provides greater ambiguity and re-ranking headroom.

4.3 Ablation Study

Contribution of Each Component. Table 4 evaluates each component using single-stage TSP3D on ScanRefer. The 7B VLM alone provides only $+0.47/+0.12$ in Overall Acc@0.25/Acc@0.5, while adding Difix3D+ increases the gain to $+0.93/+0.49$. VCF yields a larger improvement of $+2.84/+1.96$, and combining VCF with Difix3D+ reaches $59.86/49.42$, or $+3.41/+2.71$ over the base grounder. The same trend appears on the Multiple subset, where Acc@0.25 increases from 51.04 to 54.95. These results show that viewpoint selection contributes most strongly, while using the 72B VLM further improves performance to $60.86/50.12$. Further analysis of VCF, including ablations of its free-space and candidate-visibility constraints, is provided in Appendix A.1.2. Moreover, qualitative visualizations of the complete RaV pipeline are provided in Appendix A.2.

Necessity of Context-aware Verifier. Table 5 compares candidate selectors under the same 4-yaw pool on the ScanRefer Multiple subset. Prediction-level selectors use grounder confidence or a learned cross-view transformer, while feature-level baselines use DINOv2 or SigLIP candidate features. These methods recover at most 2.0% and 2.1% of the oracle headroom, respectively, whereas RaV recovers 31.9%. This result shows that candidate-local evidence is insufficient for resolving ambiguous hypotheses, and that verification benefits from surrounding spatial context. Further analyses and baseline details are provided in Appendices A.1 and A.4.

5 Conclusion

We presented RaV, a training-free Render-and-Verify framework that improves frozen 3D visual grounders through VLM-based hypothesis verification. Experiments on ScanRefer, Nr3D, and Sr3D demonstrate strong improvements and state-of-the-art results across multiple settings, particularly in scenes with same-class distractors. Future work will focus on improving hypothesis coverage and spatially informative verification while preserving the training-free plug-in design.

AI USE STATEMENT

Generative AI tools were used after the initial manuscript draft was completed to improve the clarity and presentation of the writing and to assist with the retrieval and discovery of relevant prior work. All AI-assisted edits and retrieved references were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide the methodological details required to reproduce RaV in Sec. 3, with additional implementation settings in Appendix A.4. Additional experiments and ablations, including analyses of VCF, verifier choice, candidate verification, and prompt design, are provided in Appendix A.1, while computational cost and robustness analyses are reported in Appendix A.3. All experiments use publicly available datasets and pretrained models, and the source code will be released upon acceptance.

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 *Proc. of European Conf. on Computer Vision (ECCV)*, pp. 422–440. Springer, 2020.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Proc. of European Conf. on Computer Vision (ECCV)*, pp. 202–221. Springer, 2020.

Tsung-Chih Chiang, Hsuan-Kung Yang, Jou-Min Liu, Ting-Ru Liu, Chun-Wei Huang, Quan Kong, and Chun-Yi Lee. Where to look matters: Learning influential views for vlm-based 3d visual grounding. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2026.

Hongjie Gu, Jinlong Fan, Liang Zheng, Jing Zhang, and Yuxiang Yang. Ddpa-3dvg: vision-language dual-decoupling and progressive alignment for 3d visual grounding. In *Proc. of Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 1044–1052, 2025.

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

Zoey Guo, Yiwen Tang, Ray Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pp. 15326–15337. IEEE, 2023.

Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3d visual grounding. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 15503–15512. IEEE, 2022.

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

Seongmin Jung, Seongho Choi, Gunwoo Jeon, Minsu Cho, and Jongwoo Lim. Panogrounder: Bridging 2d and 3d with panoramic scene representations for vlm-based 3d visual grounding. In *Proc. of European Conf. on Computer Vision (ECCV)*, pp. 192–209. Springer, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 3707–3717. IEEE, 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 *Proc. of ACM Int'l Conf. on Multimedia (ACM MM)*, pp. 3094–3103, 2025.

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

GwangWook Park, Hyo-Jun Lee, Jong-Hyeon Baek, Hanul Kim, and Yeong Jun Koh. Eg-3dvg: Expression and geometry aware grounding decoder for 3d visual grounding. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634, 2026.

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 *Proc. of European Conf. on Computer Vision (ECCV)*, pp. 381–398. Springer, 2024.

Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. Languagerefer: Spatial-language model for 3d visual grounding. In *Proc. of Conference on Robot Learning (CoRL)*, pp. 1046–1056. PMLR, 2022.

Junpeng Shang, Feifei Shao, Jun Xiao, Lin Li, Hongwei Wang, and Dongfang Ma. Pv-ground: Text-guided point-voxel interaction for 3d visual grounding. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 38657–38667, 2026.

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pp. 1194–1203. IEEE, 2021.

Xiangxi Shi, Zhonghua Wu, and Stefan Lee. Aware visual grounding in 3d scenes. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 14056–14065. IEEE, 2024.

Yuan Wang, Yali Li, and Shengjin Wang. Gˆ 3-lq: Marrying hyperbolic alignment with explicit semantic-geometric modeling for 3d visual grounding. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 13917–13926. IEEE, 2024.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 26024–26035. IEEE, 2025.

Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 19231–19242. IEEE, 2023.

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

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 *Proc. of Conference on Robot Learning (CoRL)*, 2024b.

Li Yang, Ziqi Zhang, Zhongang Qi, Yan Xu, Wei Liu, Ying Shan, Bing Li, Weiping Yang, Peng Li, Yan Wang, et al. Exploiting contextual objects and relations for 3d visual grounding. *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 36:49542–49554, 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 *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pp. 1771–1780. IEEE, 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 *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pp. 20623–20633. IEEE, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pp. 15179–15179. IEEE, 2023.

Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pp. 2908–2917. IEEE, 2021.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025a.

Yihang Zhu, Jinhao Zhang, Yuxuan Wang, Aming Wu, and Cheng Deng. Vgmamba: Attribute-to-location clue reasoning for quantity-agnostic 3d visual grounding. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pp. 5295–5304. IEEE, 2025b.

A Appendix

We provide additional analyses and implementation details supporting the main paper. We first provide further experiments on Nr3D, including generalization across base grounders and verifier scales, followed by detailed ablations of VCF and different verification VLMs on ScanRefer. In addition, we analyze how effectively RaV recovers the oracle headroom available in the candidate pool and investigate the effect of different VLM verification formats and prompt wordings. These results further clarify the contributions of candidate-conditioned viewpoint selection and independent VLM-based hypothesis verification. We also describe the remaining rendering, verification, and reproducibility implementation details omitted from the main text.

A.1 Additional Experimental Analysis

A.1.1 Generalization and Breakdown on Nr3D

Table A1: **Generalization across base grounders on Nr3D under the detection protocol.** We report Acc@0.25 / Acc@0.5. Base results (†) are reproduced with the official checkpoints. The published Acc@0.25 values (48.7 and 51.3 in Tab. 3) are matched within 0.2, and Acc@0.5 is not reported by the original papers.

Base Grounder	Base	Ours	
		7B	72B
TSP3D	48.6 / 37.1†	52.7 / 39.9 (+4.1 / +2.8)	53.1 / 40.2 (+4.5 / +3.1)
PV-Ground	51.1 / 38.1†	52.4 / 38.7 (+1.3 / +0.6)	53.3 / 39.1 (+2.2 / +1.0)

Generalization on Nr3D. Table A1 further evaluates whether the effectiveness of RaV transfers beyond ScanRefer by applying it to TSP3D and PV-Ground on Nr3D. RaV improves both grounders with either VLM scale, indicating that the verification framework is not tied to a particular grounding architecture. With Qwen2.5-VL-7B (Bai et al., 2025), RaV improves TSP3D by $+4.1/+2.8$ and

PV-Ground by $+1.3/+0.6$ in Acc@0.25/Acc@0.5. Increasing the verifier scale to 72B further raises the gains to $+4.5/+3.1$ and $+2.2/+1.0$, respectively. The larger improvement on TSP3D also suggests that the benefit depends on the ambiguity and remaining headroom of the underlying candidate pool, rather than yielding a fixed gain across grounders. Overall, these results demonstrate that VLM-based hypothesis verification generalizes across both datasets and base grounders.

Table A2: **Breakdown on Nr3D under the detection protocol with Acc@0.25.** We report the standard Easy/Hard and View-dependent/View-independent subsets. † denotes baselines reproduced with official checkpoints under our evaluation pipeline. For reference, the published PV-Ground breakdown is also included.

Method	**Easy**	**Hard**	**View-dep.**	**View-indep.**	**Overall**
PV-Ground (Shang et al., 2026)	55.20	47.50	48.30	52.90	51.30
TSP3D† (Guo et al., 2025)	53.38	43.95	46.69	49.61	48.60
Ours (TSP3D, 7B)	<u>56.65</u>	<u>48.75</u>	**51.04**	<u>53.49</u>	<u>52.70</u>
PV-Ground† (Shang et al., 2026)	55.12	47.17	48.09	52.65	51.10
Ours (PV-Ground, 72B)	**57.36**	**49.23**	<u>50.32</u>	**54.76**	**53.30**

Bold: The best, <u>Underline</u>: The second-best

Nr3D Subset Analysis. Table A2 reports results on the standard Nr3D Easy/Hard and View-dependent/View-independent subsets. Our reproduced PV-Ground baseline closely matches its published breakdown, validating the evaluation pipeline. RaV improves both TSP3D and PV-Ground across all four subsets. Notably, the gains remain strong on view-dependent queries ($+4.35$ and $+2.23$), indicating that visibility-based candidate framing does not degrade performance on expressions involving viewpoint-dependent relations. The larger gain on Hard queries for TSP3D ($+4.80$) further supports the benefit of verification in scenes with stronger candidate ambiguity.

A.1.2 Analysis of Visibility-Constrained Canonical Framing

Table A3: **Component-wise ablation of VCF on the ScanRefer validation set.** Starting from the full VCF, we remove one criterion at a time: free-space (C1) and visibility (C2). "Fixed view" replaces VCF with a single fixed viewpoint, and is identical to the VLM+Difix configuration without VCF in Tab. 4. We report Overall Acc@0.25 / Acc@0.5.

Variant	C1	C2	**Overall**
Base			56.45 / 46.71
Fixed view			57.38 / 47.20
w/o C1		✓	<u>59.02 / 48.60</u>
w/o C2	✓		57.47 / 47.57
Full VCF	✓	✓	**59.86 / 49.42**

Bold: The best, <u>Underline</u>: The second-best

Table A3 analyzes the two geometric criteria used to select candidate-conditioned viewpoints in VCF. Using a single fixed viewpoint yields only $57.38/47.20$, showing that simply rendering the scene from an additional view is insufficient for reliable verification. Removing the free-space constraint (C1) reduces performance from $59.86/49.42$ to $59.02/48.60$, indicating that avoiding viewpoints located inside or near reconstructed geometry improves the reliability of the rendered evidence. The effect is substantially larger when removing the visibility constraint (C2), which decreases performance to $57.47/47.57$. This confirms that explicitly selecting viewpoints from which the candidate remains sufficiently visible is the primary factor behind VCF, while the free-space constraint provides an additional improvement by preventing geometrically implausible camera placements. Together, the two criteria provide the most reliable contextual view for subsequent VLM verification.

Table A4: **Sensitivity of VCF to the number of sampled azimuths A, the orbit radius d, and the camera height** h (single-stage TSP3D (Guo et al., 2025), ScanRefer, 7B verifier; Overall Acc@0.25 / Acc@0.5 within our evaluation run). The default configuration (A=8, d=2.5 m, h=1.7 m) was fixed a priori and shared across all experiments.

A	d (m)	h (m)	**Overall**
4	2.5	1.7	59.08 / 48.68
12	2.5	1.7	59.70 / 49.04
8	1.5	1.7	**60.20 / 49.52**
8	3.5	1.7	58.28 / 48.11
8	2.5	1.5	59.84 / 49.23
8	2.5	1.9	59.65 / 49.39
8	2.5	1.7 (default)	59.86 / 49.42

A.1.3 Sensitivity of VCF Hyperparameters

Table A4 varies the three hyperparameters of the VCF camera orbit while keeping the rest of the pipeline fixed. RaV remains effective across all configurations, with every setting improving the baseline by at least $+1.8$ points at Acc@0.25. Performance varies only moderately with the number of sampled azimuths, where $A = 4$, 8, and 12 achieve 59.08, 59.86, and 59.70 Acc@0.25, respectively. The camera height is similarly insensitive, with $h = 1.5$, 1.7, and 1.9 m differing by at most 0.21 points. The orbit radius remains stable within $d \in [1.5, 2.5]$ m, where the two settings differ by only 0.34 points at Acc@0.25, while performance decreases to 58.28 when the camera is moved farther to $d = 3.5$ m. These results indicate that VCF does not depend on a narrowly tuned camera configuration. The default setting ($A = 8$, $d = 2.5$ m, $h = 1.7$ m) was fixed a priori and shared across all experiments rather than selected from this sweep. When no pose reaches the threshold, VCF falls back to the most visible free-space pose, so the selected viewpoint remains the visibility argmax.

A.1.4 Generalization across Verification VLMs

Table A5: **Ablation on the choice of the verification VLM with the single-stage TSP3D (Guo et al., 2025) base grounder.** We report Overall Acc@0.25 / Acc@0.5. The value in parentheses is the gain over the base grounder.

Verification VLM	**Overall**
– (TSP3D baseline)	56.45 / 46.71
Qwen2.5-VL-7B	59.86 / 49.42 (+3.41 / +2.71)
InternVL3-8B	60.08 / 49.52 (+3.63 / +2.81)
LLaVA-OneVision-7B	59.95 / 49.56 (+3.50 / +2.85)
Qwen2.5-VL-72B	**60.86** / 50.12 (+4.41 / +3.41)
InternVL3-78B	60.79 / **50.37** (+4.34 / +3.66)

Table A5 examines whether the performance of RaV depends on a particular VLM family. We replace Qwen2.5-VL (Bai et al., 2025) with InternVL3 (Zhu et al., 2025a) and LLaVA-OneVision (Li et al., 2024) while keeping the candidate pool, rendering procedure, verification format, and reranking strategy unchanged. All frozen VLMs substantially improve the TSP3D baseline: at the 7–8B scale, Acc@0.25 gains fall within $+3.41$ to $+3.63$ despite differences in architecture and pretraining, and the non-Qwen families remain comparable to Qwen2.5-VL-7B, indicating that our main results are not an optimistic choice of verifier. Verifier scaling also reproduces across families: Qwen2.5-VL improves from $+3.41$ to $+4.41$ when moving from 7B to 72B, and InternVL3 improves from $+3.63$ to $+4.34$ when moving from 8B to 78B, with InternVL3-78B achieving the best Acc@0.5. The small variation across VLM families indicates that the improvement is not specific to Qwen2.5-VL. Instead, RaV primarily benefits from exposing each verifier to a clearly marked candidate together with the surrounding spatial context needed to interpret the referring expression.

A.1.5 Oracle Headroom and Prediction Flips

Table A6: **Recovery rate relative to the oracle headroom on ScanRefer at Acc@0.25.** The headroom is the gap between the grounder's top-1 accuracy and an oracle that always selects the best box available in the candidate pool. We report the gain achieved by RaV, with the recovered fraction of that headroom in parentheses.

Base Grounder	Headroom	7B	72B
TSP3D (Guo et al., 2025)	13.57	+3.41 (**25%**)	+4.41 (**32%**)
PV-Ground (Shang et al., 2026) (single-stage)	15.74	+1.63 (**10%**)	+2.00 (**13%**)
PV-Ground (two-stage)	15.31	+1.78 (**12%**)	+2.33 (**15%**)
EG-3DVG (Park et al., 2026) (single-stage)	20.93	+3.31 (**16%**)	+4.60 (**22%**)
EG-3DVG (two-stage)	16.78	+2.74 (**16%**)	+3.37 (**20%**)

Table A7: **Prediction flips induced by re-ranking on ScanRefer** (query counts, within our evaluation run). "Fixed" denotes wrong→correct and "broken" correct→wrong at the given IoU threshold.

Configuration	IoU	Fixed	Broken	Kept correct	Kept wrong	Net
TSP3D + 7B	0.25	732	422	4,956	3,392	+310
TSP3D + 72B	0.25	782	377	5,001	3,342	+405
TSP3D + 72B	0.5	740	430	4,022	4,310	+310
EG-3DVG + 72B	0.25	863	425	5,001	3,219	+438
EG-3DVG + 72B	0.5	767	410	4,347	3,984	+357

Table A6 quantifies how effectively RaV exploits the alternative hypotheses made available by the candidate pool. For each grounder, the oracle headroom measures the maximum possible improvement if an oracle always selected the correct candidate whenever it appeared in the pool. We then report the fraction of this headroom recovered by RaV. Across the five evaluated configurations, the 7B verifier recovers 10–25% of the available headroom, while the 72B verifier increases this range to 13–32%. For example, TSP3D provides 13.57 points of oracle headroom, of which RaV recovers 3.41 points with the 7B verifier and 4.41 points with the 72B verifier. These results distinguish candidate generation from candidate verification. Multi-yaw inference creates additional opportunities by increasing the chance that the correct object appears among the hypotheses, but these opportunities alone do not improve the final prediction unless the correct candidate can be identified. RaV consistently converts a meaningful portion of this latent headroom into actual grounding accuracy, and the higher recovery rates obtained with the larger VLM further suggest that stronger verification can exploit the same candidate pool more effectively.

Table A7 decomposes every re-ranking decision into its effect on the final prediction. Across configurations, RaV fixes roughly two predictions for every one it breaks (*e.g.*, 782 versus 377 for TSP3D with the 72B verifier at Acc@0.25; 742 versus 349 on the Multiple subset). The decomposition also explains why the recovered fraction of the oracle headroom in Tab. A6 is bounded: for TSP3D at Acc@0.25, only $1{,}289$ of the $4{,}124$ initially wrong queries are addressable at all (the pool contains a correct box), and the verifier recovers 782 of them (61%); the dominant residual failure is therefore pool coverage rather than verification, while 7.0% of initially correct predictions ($377/5{,}378$) are flipped and partially offset the gross recovery.

A.1.6 Verification Format

We further analyze how the frozen VLM is queried while keeping all other components fixed. All experiments use the same frozen TSP3D grounder (Guo et al., 2025), 4-yaw candidate pool, VCF viewpoints, Difix3D+-corrected renderings (Wu et al., 2025), red candidate markers, Qwen2.5-VL-7B verifier (Bai et al., 2025), and re-ranking rule. Results are computed on the full ScanRefer validation set of $9{,}508$ queries, among which $6{,}301$ reach the verifier after the all-agree early exit. All runs passed our automated audit with zero missing renderings or dropped samples.

Our default verifier evaluates each candidate independently with a binary Yes/No question. To examine whether this formulation itself is important, we compare it with three alternative formats using identical candidate renderings. *Ranking* jointly presents all candidates and predicts their ordering.

Direct selection jointly presents all candidates and selects a single candidate. *Pairwise comparison* evaluates every candidate pair and aggregates the resulting preferences. In contrast, our formulation independently verifies each candidate using its first-token Yes/No logits.

Table A8: **Ablation on the verification format.** All variants use the same TSP3D candidate pool and Qwen2.5-VL-7B verifier on ScanRefer. Δ is measured over the TSP3D baseline ($56.45/46.71$).

Verification format	VLM calls	**Overall**	**Multiple**	Δ Overall
Ranking prompt	6,301	57.44 / 47.06	52.20 / 42.71	+0.99 / +0.35
Direct selection	6,301	57.84 / 47.72	52.63 / 43.31	+1.39 / +1.01
Pairwise comparison	37,806	58.23 / 47.91	53.04 / 43.47	+1.78 / +1.20
Per-candidate Yes/No (ours)	25,204	**59.86 / 49.42**	**54.95 / 45.30**	**+3.41 / +2.71**

The per-candidate binary formulation performs substantially better than joint selection or ranking. Notably, pairwise comparison requires $1.5\times$ more VLM calls than our method, yet remains 1.63 points lower in Acc@0.25. This indicates that the improvement does not arise from additional inference cost, but from independently evaluating each marked candidate without requiring the VLM to compare several hypotheses within a single context.

A.1.7 Prompt Wording

We next test whether the improvement depends on a carefully tuned prompt. The verification format is fixed to per-candidate Yes/No, while only the wording is varied. P1 is our default prompt, which introduces the rendered scene and explicitly refers to the red candidate marker. P2 removes the scene preamble, P3 places the verification question before the expression, P4 explicitly frames the task as verifying a 3D grounding prediction, and P5 refers only to the centered object without mentioning the red marker.

Table A9: **Ablation on verification prompt wording.** The per-candidate Yes/No format is fixed for all variants. Δ is measured over the TSP3D (Guo et al., 2025) baseline ($56.45/46.71$).

Prompt	Wording	**Overall**	**Multiple**	Δ Overall
P1	scene preamble + marker reference (ours)	**59.86 / 49.42**	**54.95 / 45.30**	**+3.41 / +2.71**
P3	instruction first, expression second	59.82 / 49.19	54.86 / 45.09	+3.37 / +2.48
P2	minimal, preamble removed	59.47 / 48.97	54.51 / 44.75	+3.02 / +2.26
P5	marker present but not referred to	59.37 / 48.62	54.42 / 44.49	+2.92 / +1.91
P4	explicit task framing	59.31 / 48.93	54.34 / 44.75	+2.86 / +2.22

Performance is relatively insensitive to wording. Across the five variants, Acc@0.25 varies by only 0.55 points, compared with 2.42 points across verification formats. Even the weakest prompt (P4, $+2.86$) substantially outperforms the strongest alternative format (pairwise comparison, $+1.78$). This suggests that RaV's gain primarily comes from decomposing candidate selection into independent binary verification rather than prompt engineering. Explicitly referring to the candidate marker also helps: removing this reference in P5 reduces performance by $0.49/0.80$ at Acc@0.25/Acc@0.5 relative to P1.

A.1.8 Prompt Details

All verification prompts operate on a single rendered view in which one candidate is outlined with a red rectangle. The referring expression is substituted verbatim into the placeholder $\{\mathrm{u}\}$, with no paraphrasing, no chain-of-thought instruction, and no in-context examples. In every format the decision is read from the first-token logits (a two-way softmax over the "Yes"/"No" token ids for binary prompts, and over the admissible digit tokens for the joint formats), so no free-form text is generated and no parsing is involved.

Main prompt (P1).

```
This is a rendered view of a room.  One object is outlined
with a RED box.  Description:  "{u}".  Is the object
```

```
outlined in red the object that this description refers
to?  Answer Yes or No.
```

Alternative verification formats. *Direct selection* shows all candidate views at once: "You are given {n} rendered views of the same room. In each view a different candidate object is outlined with a RED box. Description: "{u}". Which view shows the object that this description refers to? Answer with the view number only ({opts})." *Ranking* uses the same joint presentation but asks to rank the views from most to least likely. *Pairwise comparison* presents two views and asks "Which view shows the object that this description refers to? Answer with 1 or 2 only."

Wording variants. P2 (minimal): "Description: "{u}". Does the RED box mark this object? Answer Yes or No." P3 (instruction first): "Does the object inside the RED box match the following description? "{u}" Answer Yes or No." P4 (task framing): "You are verifying a 3D visual grounding prediction. The RED box marks the predicted object. Referring expression: "{u}". Is the prediction correct? Answer Yes or No." P5 (marker not referred to): "This is a rendered view of a room, centered on one particular object. Description: "{u}". Is the object in the CENTER of this image the object that this description refers to? Answer Yes or No." P5 is a control: the rendered image still carries the red marker, but the prompt never mentions it, isolating how much of the gain is attributable to the marker being referred to rather than merely being present.

A.2 Qualitative Analysis of the RaV Pipeline

Figures A1 and A2 visualize representative examples of the complete RaV pipeline with EG-3DVG (Park et al., 2026). For each scene, panel (a) shows the hybrid candidate construction used for ranked-proposal grounders: the native top-4 predictions from $S^{(0)}$ are combined with the top-1 prediction from each of $S^{(1)}, S^{(2)}, S^{(3)}$, yielding up to seven raw hypotheses before 3D IoU de-duplication with $\tau_{\text{dup}} = 0.5$. Panel (b) shows the resulting candidate pool $\mathcal{B}$ together with the VCF search for each candidate, where free-space-invalid viewpoints are rejected and the highest-visibility feasible view is selected. Panel (c) visualizes the raw and Difix3D+-corrected renderings, followed by the candidate-conditioned verification images and their corresponding probabilities $P_i(\text{Yes})$ from the 72B verifier. For EG-3DVG, the final prediction is selected using the conservative margin rule with $\tau = 0.1$, keeping the original prediction b_1 unless an alternative candidate receives sufficiently stronger verification evidence. The selected hypothesis $\hat{b}$ is highlighted in orange. These examples illustrate the complete transition from orientation-dependent proposal diversity to candidate-specific viewpoint selection and final VLM-based re-ranking.

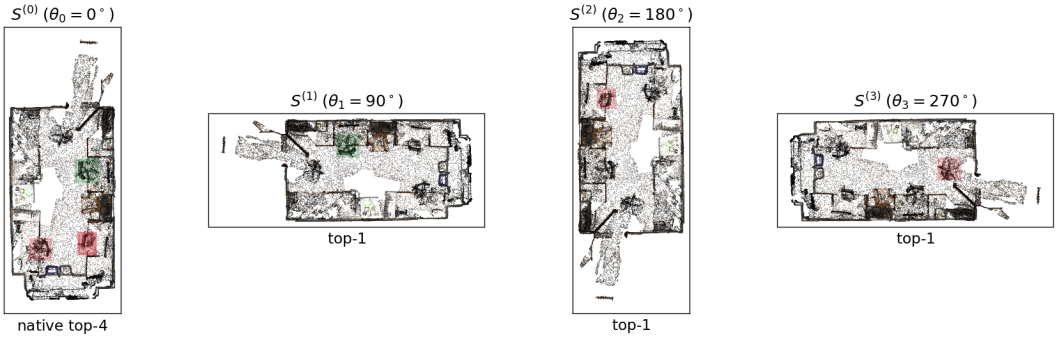
A.3 Efficiency and Robustness Analysis

A.3.1 Computational Cost

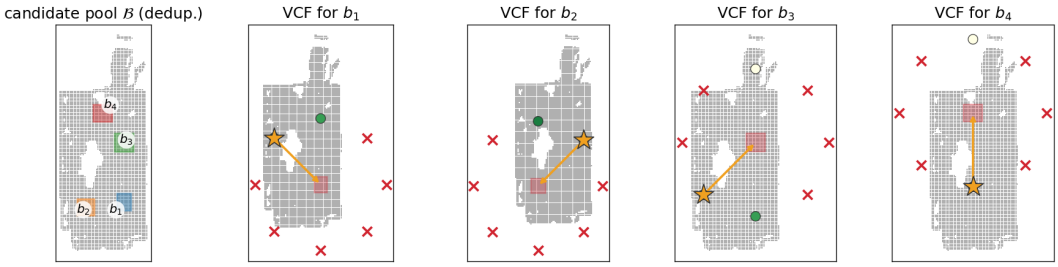
Table A10: **Per-query cost of each RaV stage on ScanRefer with TSP3D (Guo et al., 2025)** (measured on A6000 GPUs). Amortized numbers include the 33.7% of queries closed by the all-agree early exit, which incur no rendering or VLM cost; queries that reach the verifier average 4.0 images.

Stage	Unit cost	Amortized per query
Candidate pool (yaw TTA)	3 extra forward passes of the frozen grounder	3 grounder passes
VCF viewpoint selection	$<$ 1 ms per pose (CPU ray marching)	$<$ 5 ms (CPU)
Rendering + Difix3D+	0.80 GPU-s / image	$2.65 \times 0.80 \approx 2.1$ GPU-s
Verification (7B, batch 48)	0.82 GPU-s / image	$2.65 \times 0.82 \approx 2.2$ GPU-s
Verification (72B, 4-GPU sharded)	6.4 GPU-s / image	$2.65 \times 6.4 \approx 17.0$ GPU-s

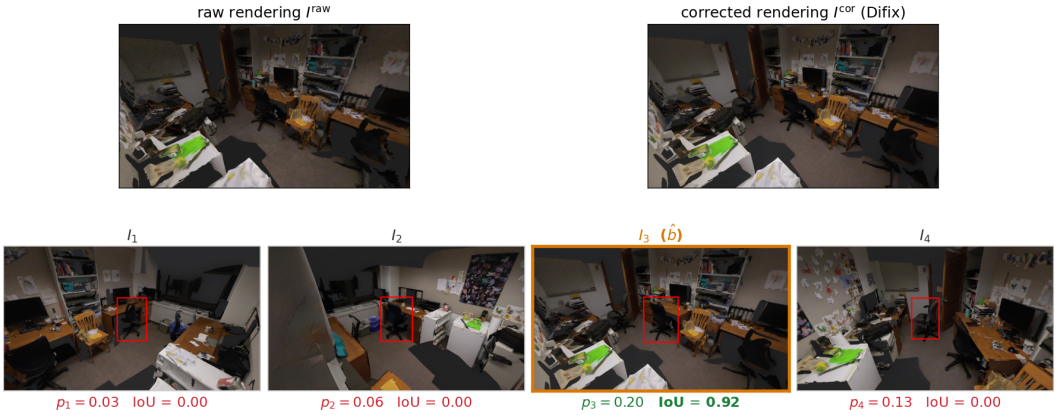
Table A10 reports the measured cost of every stage RaV adds on top of the base grounder. With the 7B verifier, RaV adds approximately 4.3 GPU-seconds per query on average; processing the full ScanRefer validation split ($9{,}508$ queries, $25{,}204$ rendered images) takes about 85 minutes of wall-clock time on eight A6000 GPUs, i.e., roughly 11 GPU-hours. The 72B verifier raises the verification cost to about 17 GPU-seconds per query. The all-agree early exit (33.7% of ScanRefer queries; 33–44% across datasets) is exact rather than approximate, since re-ranking cannot change



(a) Multi-yaw candidate generation.

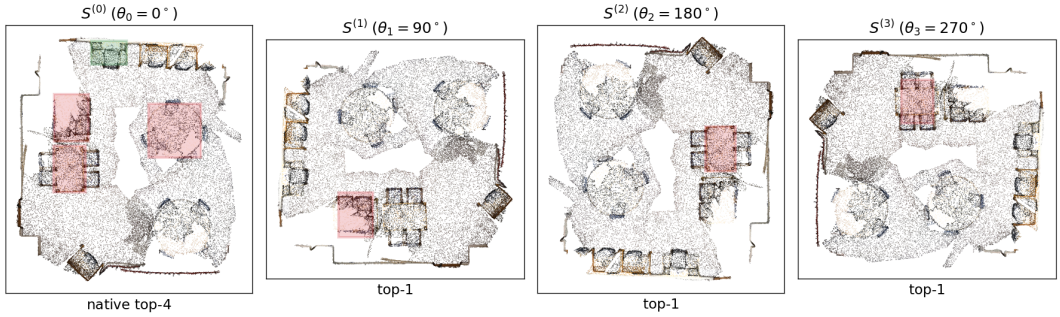


(b) Visibility-Constrained Canonical Framing.

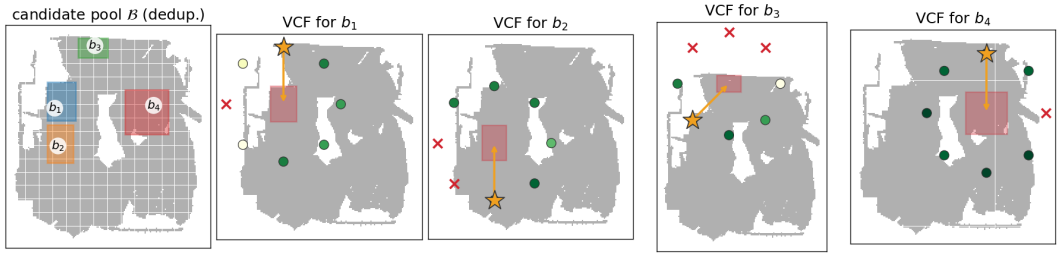


(c) Candidate rendering and VLM verification.

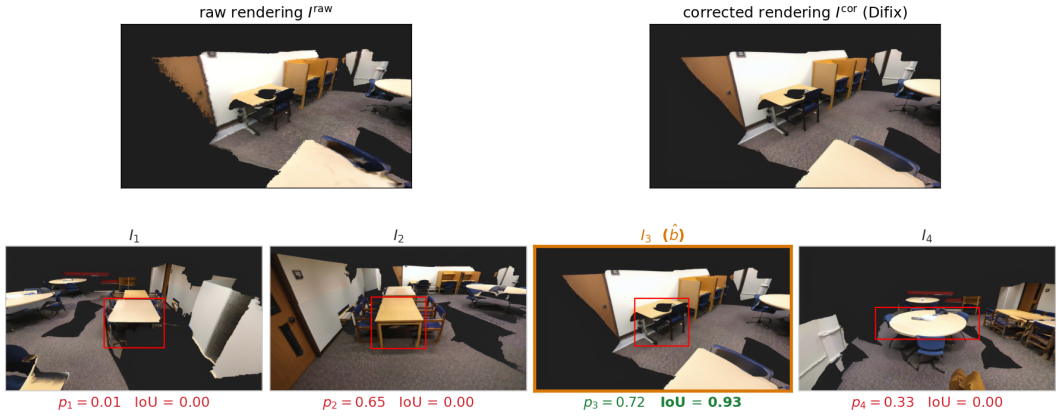
Figure A1: **Qualitative example of RaV with EG-3DVG.** The three panels visualize hybrid candidate construction, VCF viewpoint selection, and 72B VLM verification, respectively. Referring expression: "the chair is the south - west - most one in the room. the chair is black with five legs."



(a) Multi-yaw candidate generation.



(b) Visibility-Constrained Canonical Framing.



(c) Candidate rendering and VLM verification.

Figure A2: **Qualitative example of RaV with EG-3DVG.** The three panels visualize hybrid candidate construction, VCF viewpoint selection, and 72B VLM verification, respectively. Referring expression: "this is a brown table. it is behind the chair ."

the output when all candidates coincide, and it is the largest single source of cost reduction. VCF itself is essentially free: viewpoint scoring uses occupancy-grid lookups only, so each candidate is rendered exactly once after its viewpoint has been decided. RaV therefore introduces an explicit accuracy–compute trade-off: unlike the base grounders, it additionally invokes a rendering corrector and a frozen VLM at inference time. Our comparisons should thus be interpreted as measuring the accuracy gained by a training-free verification stage, rather than as compute-matched replacements for the underlying grounders.

A.3.2 Sensitivity to the Verification Margin

Table A11: **Sensitivity of Overall Acc@0.25 to the verification margin** τ (72B verifier). * marks the value used in all main results, shared across grounders and verifier scales

. Absolute accuracies are computed within our evaluation run.

τ	ScanRefer (EG-3DVG (Park et al., 2026))	Nr3D (PV-Ground (Shang et al., 2026))	Sr3D (PV-Ground)
0.0	61.57	52.35	52.36
0.05	61.67	52.98	53.84
0.1	61.67*	53.13	54.78
0.2	61.23	53.27*	55.73
0.3	61.16	53.27	56.18
0.4	60.90	53.15	56.39*
Base	57.07	51.12	56.18

Table A11 sweeps the margin over a fixed grid. On ScanRefer and Nr3D the accuracy is flat over a broad plateau: every $\tau \in [0.05, 0.4]$ improves over the baseline, and the chosen values sit inside the plateau rather than at a sharp optimum, so the reported gains do not depend on a finely tuned margin. Sr3D behaves differently because the verifier's discrimination is weakest there (pairwise probability of scoring a correct candidate above an incorrect one is only 58%, versus 66% on ScanRefer): accuracy increases monotonically with τ, and the margin converts what an ungated rule ($\tau = 0$) would turn into a -3.8-point loss into a $+0.2$-point gain. This is exactly the intended behavior of the conservative rule: when the verifier provides weak evidence, RaV declines to lose rather than manufacturing a gain.

A.3.3 Semantic Fidelity of Artifact Correction

Table A12: **Prediction changes induced by Difix3D+** (within our evaluation run). Each pair of runs shares identical viewpoints and differs only in artifact correction; we count queries whose final prediction changes, split into fixed (wrong→correct), broken (correct→wrong), and neutral (both wrong or both correct).

Configuration	IoU	Changed	Fixed	Broken	Neutral	Net
TSP3D (VCF viewpoints)	0.25	3,503 / 9,508	339	285	2,879	+54
TSP3D (VCF viewpoints)	0.5	3,503 / 9,508	379	308	2,816	+71
EG-3DVG (fixed viewpoints)	0.25	1,201 / 9,508	300	288	613	+12
EG-3DVG (fixed viewpoints)	0.5	1,201 / 9,508	274	263	664	+11

A potential concern with diffusion-based artifact correction is that it may alter semantic evidence, such as object identity, color, or surrounding objects, and thereby mislead the verifier. We therefore examine the effect of Difix3D+ (Wu et al., 2025) at both the prediction and image levels.

First, Table A12 compares paired runs that use identical candidate pools and viewpoints and differ only in whether artifact correction is applied. Difix3D+ changes the final prediction for $3{,}503$ of $9{,}508$ TSP3D (Guo et al., 2025) queries, reflecting that even modest changes in verification scores can affect near-tie candidate decisions. However, these changes do not exhibit a systematic harmful bias. At Acc@0.25, artifact correction fixes 339 predictions while breaking 285, and the same pattern holds at Acc@0.5 (379 fixed versus 308 broken). EG-3DVG (Park et al., 2026) shows a similar behavior, with 300 fixed versus 288 broken predictions at Acc@0.25. These results are consistent with the modest positive contribution of Difix3D+ observed in the component ablation.

Prediction-level changes alone, however, do not establish that the corrected images preserve semantic content. We therefore additionally audit randomly sampled raw/corrected rendering pairs using an independent VLM judge from a different model family than the verification VLM (InternVL3-8B (Zhu et al., 2025a), queried with binary questions read from first-token logits). The audit separately checks whether correction changes the target object identity, materially alters its color, or adds or removes surrounding objects. Among 200 audited pairs, 96.0% preserve all three semantic properties, while object-identity, material-color, and object-addition/removal changes are flagged in only 1.0%, 1.5%, and 2.0% of the samples, respectively. These results indicate that Difix3D+ primarily removes rendering degradation while rarely modifying the semantic evidence used for candidate verification.

A.4 Additional Implementation Details

This section reports the remaining settings omitted from Sec. 3 and Sec. 4.1.

Baseline Selectors . The learned cross-view selector is a text-conditioned transformer (4 layers, d=256) over candidate–view evidence tokens, trained with a cross-entropy objective while both backbones remain frozen. The DINOv2 baseline projects each candidate box into four real RGB frames with approximately orthogonal yaw coverage and mean-pools frozen patch features within the projected region. The SigLIP baseline instead extracts multi-scale candidate crops and scores them against the referring expression using frozen language-aligned embeddings. These feature-level baselines use real RGB views, whereas RaV uses mesh renderings; thus, the comparison is intended to contrast candidate-local and contextual evidence rather than imaging domains.

Correct and distractor candidates are often visually similar because they belong to the same semantic class. Their projected crops have a mean pairwise DINOv2 cosine similarity of 0.75, compared with 0.24 for random object pairs. This suggests that the weak performance of candidate-local selectors stems from limited discriminative evidence rather than insufficient encoder capacity.

Rendering. Scenes are rendered from the ScanNet meshes, transformed into the axis-aligned frame used by the grounders. Rendering is performed with an offscreen OpenGL renderer through EGL, which requires no display server and allows the renderer to be replicated across GPUs. The occupancy grid used by VCF is built once per scene and cached, so it is reused across all candidates and all queries of that scene rather than being recomputed per hypothesis.

Artifact correction. Difix3D+ (Wu et al., 2025) is run in half precision at a fixed timestep with classifier-free guidance disabled, so a single forward pass is sufficient. The candidate marker is a five-pixel-wide red rectangle, and the same width is used for every candidate, grounder, and dataset.

Verifier. Beyond Qwen2.5-VL-7B and Qwen2.5-VL-72B (Bai et al., 2025), we additionally evaluate InternVL3-8B, InternVL3-78B (Zhu et al., 2025a), and LLaVA-OneVision-7B (Li et al., 2024) on identical renders, pools, and prompts. Every verifier is used off-the-shelf in half precision with its default processor configuration. We do not normalize input resolution across families, so each model sees the renders exactly as its own preprocessing pipeline prepares them. The affirmative and negative token ids are resolved separately for each tokenizer, since the surface forms of "Yes" and "No" are not tokenized identically across families. The 7B/8B models are replicated one per GPU, while the 72B and 78B models are sharded across four GPUs with layer-wise model parallelism.

Re-ranking. The verification margin is $\tau = 0.1$ on ScanRefer, $\tau = 0.2$ on Nr3D, and $\tau = 0.4$ on Sr3D, held fixed across all grounders and both verifier scales. The margin rule operates on the de-duplicated hybrid pools of PV-Ground (Shang et al., 2026) and EG-3DVG (Park et al., 2026), whose candidates carry a native ranking with the original prediction as the default hypothesis. TSP3D (Guo et al., 2025) provides exactly one prediction per orientation and no native ranking among them, so its pool is not de-duplicated and the final prediction is selected directly as $\hat{n} = \arg\max_n P_n(\text{Yes})$, as described in the main text. Sensitivity to τ is reported in Appendix A.3.2.

Compute and reproducibility. All experiments run on NVIDIA RTX A6000 GPUs (48 GB). Rendering is parallelized across GPUs with two workers each, and 7B/8B verification runs one

process per GPU. Every verification run is accompanied by an automated audit that asserts the number of output rows equals the number of queries and reports the number of candidates skipped by the early exit, the number of missing renders, and the number of out-of-memory batch splits. We report only runs for which the missing-render count is zero. When a batch exceeds device memory, it is halved and retried recursively so that no sample is silently dropped. Software versions are PyTorch 2.5.1 (CUDA 12.1) with Transformers 5.14.1 for grounding and verification, and Diffusers 0.25.1 for artifact correction.