# RelGS: Relation-Aware Gaussian Splatting for Open-Vocabulary 3D Scene Understanding

**Kai Zhao**[1,*] **Yixin Chen**[2] **Dai Shi**[3] **Ye Wang**[4] **Oula Ghannoum**[1,*] **Yi Guo**[1,*]

[1]Western Sydney University, Australia
[2]Adelaide University, Australia
[3]University of Cambridge, United Kingdom
[4]West China Hospital, Sichuan University, China
{90961592, o.ghannoum, y.guo}@westernsydney.edu.au

## Abstract

Open-vocabulary 3D scene understanding provides an important interface for querying and interacting with reconstructed scenes through natural language. Although recent 3DGS methods have enabled text-driven object selection and open-vocabulary segmentation, they still struggle with compositional queries involving attributes, spatial relations, and part-whole relations. Most existing approaches learn per-Gaussian or per-cluster language features, construct query-conditioned referring fields, or perform spatial reasoning only at inference time, but they do not organize the scene itself into a reusable relation-aware representation. As a result, object relations are not explicitly stored as typed and confidence-aware 3D structures, limiting compositional reasoning and multi-hop querying. To address this issue, we propose RelGS, a relation-aware 3D Gaussian Splatting framework for open-vocabulary scene understanding. RelGS learns instance embeddings for Gaussians under multi-view supervision, groups them into cross-view consistent 3D semantic nodes, aggregates mask-level CLIP features into a visual embedding per node, and enriches each node with structured attributes verified by reverse CLIP consistency. It further constructs a confidence-weighted relation graph, where 3D spatial cues propose candidate relations and an LLM verifies their semantic plausibility. At query time, RelGS combines attribute matching, CLIP retrieval, and graph traversal to localize the target object. Extensive experiments demonstrate strong performance on word-level selection, sentence-level referring segmentation, multi-hop relational queries, and point-level semantic segmentation, and a direct graph-level evaluation on annotated relations supports the correctness of the graph itself.

## 1 Introduction

Open-vocabulary 3D scene understanding aims to localize and interact with objects in 3D scenes using free-form natural language. While early queries often focus on categories or attributes, practical instructions are frequently compositional: users may refer to an object by its appearance, its spatial relation to another object, or its part-whole association. Answering such queries requires more than recognizing isolated entities; it requires understanding how entities are organized and related in 3D space. 3D Gaussian Splatting (3DGS) [18] provides an efficient and explicit representation for 3D scenes through point-based primitives and real-time rendering. Recent open-vocabulary 3DGS methods inject language capability into 3DGS by distilling 2D vision-language priors, such as CLIP [36] and SAM [20, 37], into Gaussian-level or cluster-level features [34, 46, 39, 50, 51, 23].

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

These methods enable text-driven object selection and open-vocabulary segmentation, and are effective for category- or attribute-level grounding. However, most existing methods still formulate language grounding as feature retrieval: a text query is encoded into an embedding and matched against Gaussian or cluster features. Under such a retrieval-based paradigm, relational expressions are compressed into a single similarity score. Consequently, the scene does not explicitly store which object is on, inside, attached to, or part of another object, making it difficult to distinguish semantically similar objects that differ mainly by their relations.

Recent works have started to address more complex referring expressions and spatial reasoning. ReferSplat [14] learns Gaussian–text referring fields with position-aware cross-modal interactions, while spatial reasoning frameworks such as SpatialReasoner [29] decompose queries into target objects, anchor objects, and spatial relations for hierarchical localization. These methods improve query-conditioned grounding for complex language, but they still primarily predict a target instance or mask for each input query. The relational evidence used during inference is not organized as a persistent scene structure. As a result, it cannot be directly reused for compositional queries, multi-hop traversal, relation-level uncertainty propagation, or downstream interaction.

We argue that open-vocabulary 3DGS understanding should move beyond query-conditioned prediction toward reusable scene organization. Instead of resolving each language query independently, a reconstructed 3DGS scene should be organized into a relation-aware representation, where object instances are stored as semantic nodes and their 3D relations are stored as typed, directed, and confidence-aware edges. Such an explicit structure allows the scene to be queried repeatedly under different language expressions, supports multi-hop traversal over object relations, and propagates uncertainty from node reliability to relation reliability. To this end, we propose **RelGS**, a relation-aware 3D Gaussian Splatting representation that converts a reconstructed 3DGS scene into a persistent spatial-semantic graph. Each node stores an object-level Gaussian group with visual embeddings, structured attributes, spatial statistics, and semantic reliability, while each edge stores a typed, directed, and confidence-aware inter-object relation. RelGS builds this graph in a geometry-first manner: multi-view instance supervision and node-level CLIP aggregation produce cross-view consistent nodes, 3D spatial cues propose candidate relations, and semantic verification refines their plausibility. At query time, natural language expressions are parsed into node and relation constraints and resolved by graph traversal with CLIP-based reranking. Our contributions are summarized as follows:

- We formulate open-vocabulary 3DGS scene understanding as reusable relation-aware scene organization, moving beyond query-conditioned mask prediction toward persistent node-and-edge representations.
- We introduce instance-embedding node learning with node-level CLIP aggregation to group Gaussians into cross-view consistent object-level nodes with visual embeddings, attributes, spatial statistics, and reliability estimates.
- We propose a geometry-first relation graph construction mechanism, where 3D spatial cues propose typed edges and semantic verification refines them into directed, confidence-weighted relations.
- We show that RelGS supports word-level retrieval, sentence-level referring segmentation, multi-hop querying, click-based interaction, and ScanNet point-level evaluation, demonstrating reusability across queries and benchmarks, and we evaluate the relation graph directly with graph-level metrics.

## 2 Related Work

**Language-Embedded 3D Gaussian Splatting.** 3D Gaussian Splatting [18] provides an explicit, real-time alternative to NeRF [31] and its accelerated variants [3, 32, 6], and has been widely extended to downstream tasks such as segmentation, editing, and generation [13, 8]. A growing body of work injects 2D vision–language priors [20, 36, 33] into 3DGS along two main directions: distilling CLIP and SAM features into individual Gaussians [34, 39, 50, 51, 35, 25], with extensions for hierarchical features [4], cross-view contrastive codebooks [41], and identity-aware consistency [15]; and lifting 2D masks into instance-level Gaussian groups [47, 46, 5, 9, 23, 17], with 3D-aware memory [30] or hierarchical contrastive learning [48] mitigating cross-view drift. These methods typically produce flat per-Gaussian embeddings or loosely grouped clusters, where object identity, semantics, and multi-view consistency remain weakly coupled. In contrast, RelGS learns a contrastively supervised

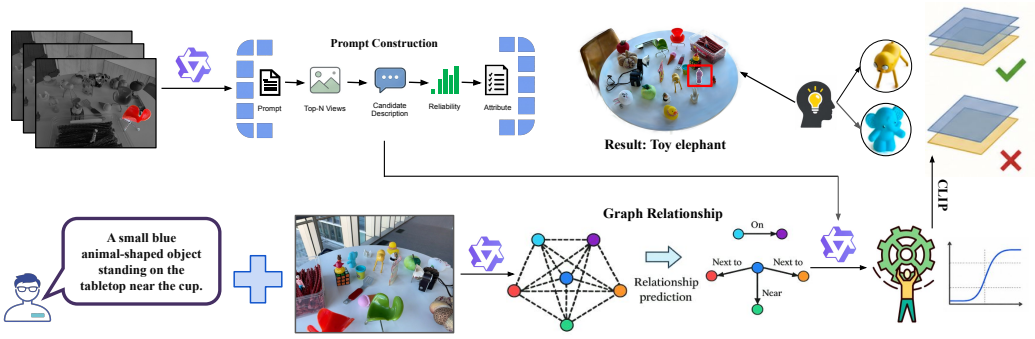



Figure 1: LLM-guided open-vocabulary query process. Given a natural-language query, the system uses multi-view node descriptions and candidate attributes to parse the query, filter candidate nodes with CLIP matching, and localize the target object.

instance field and aggregates raw CLIP features at the node level, organizing Gaussians into stable 3D semantic nodes rather than isolated retrieval units.

**Open-Vocabulary 3D Grounding and Reasoning.** Open-vocabulary 3D grounding localizes 3D objects from free-form language, extending point-cloud-based referring grounding [7, 1] and NeRF-based language fields [19, 21] to Gaussian representations. Recent 3DGS methods move beyond category-level retrieval: ReferSplat [14] formulates referring segmentation via Gaussian–text contrastive learning, FMLGS [40] targets fine-grained part-level queries, and more recent systems [29, 28, 38] couple LVLMs and MLLMs with hierarchical Gaussian features for implicit and relational reasoning. However, these methods either remain dominated by embedding retrieval over per-Gaussian features, or rely on LLMs as post-hoc reasoners over free-form captions detached from explicit 3D structure, so relational expressions like "the duck on the cube" are easily confused with visually similar objects satisfying only the category description. RelGS addresses this by attaching a structured four-field attribute schema to each 3D semantic node, providing symbolic evidence for relation verification while preserving CLIP-based open-vocabulary retrieval.

**3D Scene Graphs.** Scene graphs represent objects as nodes and pairwise relations as edges, supporting compositional reasoning for navigation, manipulation, and scene understanding. Early 3D scene graphs use closed-vocabulary point clouds [42, 45]; recent open-vocabulary methods [12, 44, 26, 22] combine foundation models [20, 36, 27, 33] and LLMs to extract objects and relations from fused 3D maps, with DINO-based map accumulation [33, 26] reducing online perception cost. However, these methods typically infer relations from 2D observations and apply 3D geometry only as a post-filter rather than a constructive prior, and query a static global graph that often contains irrelevant nodes and unstable relations [14, 29, 12, 11, 49]. In contrast, our graph is built directly over 3D Gaussian nodes: geometric cues propose typed candidate edges, semantic attributes verify plausibility, and node reliability propagates to edge reliability. At query time, language is decomposed into target, anchor, and relation constraints, tying symbolic reasoning back to a selected 3D Gaussian set.

## 3 Method

Given posed multi-view RGB images $\{I_s\}_{s=1}^{S}$ of a static 3D scene and a 3D Gaussian Splatting representation [18], denoted by $\mathcal{G} = \{g_i\}_{i=1}^{N}$, where each Gaussian $g_i = \{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i, \mathbf{c}_i\}$ encodes position, covariance, opacity, and appearance, our goal is to organize the scene into a spatial–semantic relation graph $\mathcal{K} = \langle V, E \rangle$ for open-vocabulary relational 3D instance querying. Given a natural language query $q$, the system returns the target instance as a 3D Gaussian subset $\mathcal{G}_{v^*} \subseteq \mathcal{G}$, instead of producing a view-dependent 2D mask.

Each node $v_k \in V$ represents a semantic instance and stores its Gaussian subset $\mathcal{G}_k$, visual embedding $\mathbf{T}_k$, structured attributes $\mathbf{a}_k$, and 3D spatial statistics $\{\mathbf{x}_k^c, r_k, B_k\}$. The spatial statistics correspond to the node centroid, enclosing radius, and axis-aligned bounding box. Each edge $e_{ab} \in E$ represents a directed relation from node $a$ to node $b$, associated with a relation label $\ell_{ab}$ and a confidence score $\gamma_{ab}$. We maintain a clear separation between semantic and spatial evidence. Semantic evidence describes

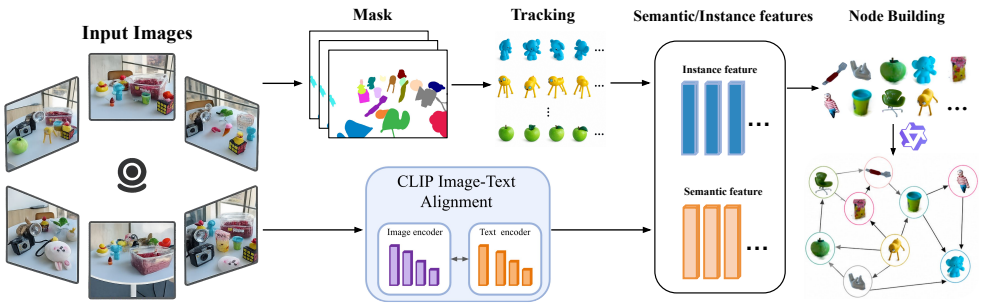



Figure 2: Multi-view supervised semantic node learning. RelGS learns instance embeddings for 3D Gaussians using SAM masks and cross-view tracking, combines them with mask-level CLIP semantic features, and groups Gaussians into stable entity-level nodes for relation-aware scene understanding.

the identity and attributes of an instance, whereas spatial evidence characterizes its geometric relation to other instances in 3D space.

### 3.1 Multi-View Supervised Semantic Node Learning

To construct object-level nodes from 3DGS, we augment each Gaussian with a learnable $\ell_2$-normalized instance embedding $\mathbf{h}_i$ that captures object identity. During training, SAM masks provide region-level supervision for this embedding, whereas semantics enter at the node level through CLIP aggregation after training. The original 3DGS geometry, opacity, and color parameters remain constrained by the RGB reconstruction objective, preserving scene geometry and appearance while the instance field is learned; the field propagates no gradient to geometry.

Since SAM masks are generated independently in each view, we use SAM2 [37] to propagate masks across views and assign globally consistent identities. Propagated masks are matched to local SAM masks by IoU with a threshold of 0.5, and unmatched masks are excluded from per-mask supervision. This yields cross-view mask sets $\{M_{s,k}\}_s$ that correspond to the same underlying object. Although the instance terms below are defined within a single view, $\mathbf{h}_i$ is one per-Gaussian vector rendered into every visible view, so consistent mask decompositions prevent different views from pulling the same Gaussians toward incompatible means. Through differentiable alpha compositing, instance embeddings are rendered into pixel-level instance features $\hat{\mathbf{H}}_s(u)$. The mean instance feature within mask $M_{s,k}$ is defined as $\bar{\mathbf{H}}_{s,k} = |M_{s,k}|^{-1} \sum_{u \in M_{s,k}} \hat{\mathbf{H}}_s(u)$. Details on the compositing weights and the $\ell_2$ normalization of $\bar{\mathbf{H}}_{s,k}$ used in Eq. (1) are provided in Appendix A.2. We optimize the following objective:

$$\mathcal{L} = \lambda_1 \sum_{s,k} \frac{1}{|M_{s,k}|} \sum_{u \in M_{s,k}} \underbrace{\left\| \hat{\mathbf{H}}_s(u) - \bar{\mathbf{H}}_{s,k} \right\|_2^2}_{\text{instance compactness}} + \lambda_2 \sum_{s} \sum_{\substack{k,l \in \mathcal{M}_s \\ k \neq l}} \underbrace{\max\left(0, \cos(\bar{\mathbf{H}}_{s,k}, \bar{\mathbf{H}}_{s,l}) - m\right)}_{\text{instance separation}}. \tag{1}$$

Here, $\mathcal{M}_s$ is the set of mask identities in view $I_s$. The objective contains two complementary terms. The instance compactness term encourages features within the same mask to form a coherent instance representation. The instance separation term pushes different masks in the same view apart using a cosine hinge margin $m$. After training, we construct a $k$-NN graph over Gaussians using a joint similarity that combines instance-embedding cosine similarity and 3D spatial proximity. Each connected component is treated as a semantic node $v_k$. Semantics are attached at this level: for a SAM mask $M_{s,k}$ in view $I_s$, we crop the masked region and encode it with the CLIP image encoder $E_{\text{img}}(\cdot)$ to obtain a unit-norm mask-level feature $\mathbf{t}_{s,k}$. For each node, we compute its Gaussian subset $\mathcal{G}_k$, visible view set $\mathcal{V}_k$, spatial statistics $\{\mathbf{x}_k^c, r_k, B_k\}$, and visual embedding $\mathbf{T}_k$. The node visual embedding is obtained by averaging the mask-level CLIP embeddings $\{\mathbf{t}_{s,k}\}_{s \in \mathcal{V}_k}$ over its visible views, which is where cross-view semantic aggregation takes place.

### 3.2 Node Attribute Enrichment

The node visual embedding $\mathbf{T}_k$ supports open-vocabulary retrieval, but relation reasoning also requires discrete and interpretable semantic fields. We therefore enrich each node with structured attributes. For node $v_k$, representative views are selected according to SAM mask stability and relative visible area:

$$Q(s,k) = q_{s,k}^{\mathrm{SAM}} \cdot \frac{|M_{s,k}|}{\max_{s' \in \mathcal{V}_k} |M_{s',k}|}. \tag{2}$$

In this score, $q_{s,k}^{\mathrm{SAM}}$ denotes the SAM stability score, $|M_{s,k}|$ denotes the mask area in the current view, and $\mathcal{V}_k$ denotes the visible view set of node $v_k$. The score favors views where the target mask is both stable and visually complete. We crop the masked regions from the top-$N_v$ selected views and feed them into a VLM to obtain structured attributes $\mathbf{a}_k$ and a short node caption $c_k$. The attribute schema includes category, color, material or shape, and function. These fields describe the node itself and are not used to directly extract inter-node relations.

Free-form VLM captions may contain view-dependent spatial expressions such as *left*, *right*, *in front of*, and *behind*, which do not necessarily correspond to stable 3D relations. To avoid introducing view-dependent bias, we restrict VLM outputs to per-node attributes and derive inter-node relations from 3D spatial statistics in Sec. 3.3. To suppress obvious VLM naming errors, we compute a semantic reliability weight by checking the consistency between the node visual embedding and the CLIP text embedding of its caption:

$$w_k^{\mathrm{sem}} = \mathbf{1}[\cos(\mathbf{T}_k, E_{\mathrm{text}}(c_k)) \geq \tau_{\mathrm{sem}}]\,. \tag{3}$$

Here, $E_{\mathrm{text}}(\cdot)$ denotes the CLIP text encoder. The resulting weight $w_k^{\mathrm{sem}}$ modulates relation confidence in subsequent graph construction.

### 3.3 Spatial–Semantic Relation Graph Construction

The relation graph aims to establish explicit, directed, and confidence-weighted relations between semantic nodes. We decompose relation construction into two stages. The first stage uses 3D spatial statistics to propose candidate node pairs and discretizes continuous geometric measurements into symbolic spatial cues. The second stage uses an LLM to verify candidate relations using endpoint attributes and symbolic cues. This design grounds relation candidates in computable 3D structure, while allowing relation labels to be determined with semantic context.

For each node pair, node $a$ is treated as the candidate target node and node $b$ as the candidate reference node. Candidate pairs are generated by spatial proximity:

$$\mathcal{P} = \{(a,b) : \|\mathbf{x}_a^c - \mathbf{x}_b^c\|_2 \leq \rho \cdot \max(r_a, r_b)\}\,. \tag{4}$$

In this criterion, $\mathbf{x}_a^c$ and $\mathbf{x}_b^c$ are node centroids, $r_a$ and $r_b$ are enclosing radii, and $\rho$ controls the neighborhood scale. For each candidate pair, we compute spatial descriptors including vertical offset, horizontal distance, 3D distance, and AABB (Axis-Aligned Bounding Boxes) overlap. These descriptors are discretized into symbolic spatial cues, including `INSIDE`, `ABOVE`, `BELOW`, `ATTACHED`, and `ADJACENT`. All cues are defined along the candidate edge direction from node $a$ to node $b$.

We use a compact predicate set covering common part-whole, attachment, containment, support, vertical, and proximity relations. The set deliberately excludes viewpoint-dependent relations, which are ambiguous without a canonical object orientation. Relation phrases in the input query are normalized to this predicate set for graph traversal. For each candidate edge, the LLM predicts the relation label and verification confidence from endpoint attributes, candidate predicates, and the symbolic spatial cue:

$$(\ell_{ab}, \gamma_{ab}^{\mathrm{ver}}) = \mathrm{LLM}_{\mathrm{ver}}\left(\mathbf{a}_a, \mathbf{a}_b, \mathcal{R}, \mathrm{RelSym}(a,b)\right). \tag{5}$$

The structured attributes of the endpoint nodes are denoted by $\mathbf{a}_a$ and $\mathbf{a}_b$, the candidate predicate set by $\mathcal{R}$, and the symbolic spatial cue by $\mathrm{RelSym}(a,b)$. We further compute a soft spatial compatibility score $\gamma_{ab}^{\mathrm{sp}} \in [0,1]$ from normalized inter-node distance, vertical offset, and AABB overlap according to the symbolic cue. The final edge confidence is defined as

$$\gamma_{ab} = \gamma_{ab}^{\mathrm{sp}} \cdot \gamma_{ab}^{\mathrm{ver}} \cdot w_a^{\mathrm{sem}} \cdot w_b^{\mathrm{sem}}, \tag{6}$$

Table 1: Comparison of word-level 3D selection on LERF-OVS [34]. Results are reported in mIoU and mAcc. Best results are in **bold**.

| | | Figurines | | Ramen | | Teatime | | Waldo_Kitchen | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Method | mIoU | mAcc | mIoU | mAcc | mIoU | mAcc | mIoU | mAcc | mIoU | mAcc |
| 2D evaluation | Feature-3DGS [50] | 40.50 | 73.40 | 43.70 | 69.80 | 58.80 | 77.20 | 39.60 | 87.60 | 45.65 | 77.00 |
| | GS-Grouping [47] | 40.70 | 74.30 | 45.50 | 68.60 | 60.90 | 75.00 | 38.70 | 88.20 | 46.45 | 76.53 |
| | LEGaussian [39] | 40.80 | 75.20 | 46.00 | 67.50 | 60.30 | 75.60 | 39.40 | 90.30 | 46.63 | 77.15 |
| | LangSplat [34] | 44.70 | 80.40 | 51.20 | 73.20 | 65.10 | 88.10 | 44.50 | **95.50** | 51.38 | 84.30 |
| | LangSplatV2 [25] | 56.40 | 82.10 | **51.80** | 74.70 | **72.20** | **93.20** | 59.10 | 86.40 | 59.88 | 84.10 |
| | **Ours** | **61.72** | **86.31** | 51.35 | **76.33** | 71.25 | 92.30 | **67.20** | 89.43 | **62.88** | **86.09** |
| 3D evaluation | LangSplat [34] | 10.16 | 8.93 | 7.92 | 11.27 | 11.38 | 20.34 | 9.18 | 9.09 | 9.66 | 12.41 |
| | LEGaussian [39] | 17.99 | 23.21 | 15.79 | 26.76 | 19.27 | 27.12 | 11.78 | 18.18 | 16.21 | 23.82 |
| | OpenGaussian [46] | 39.29 | 55.36 | 31.01 | 42.25 | 60.44 | 76.27 | 22.70 | 31.82 | 38.36 | 51.43 |
| | VoteSplat [16] | **68.62** | 85.71 | 39.24 | 61.97 | 66.71 | **88.14** | 25.84 | 33.68 | 50.10 | 67.38 |
| | Dr.Splat [17] | 53.36 | 80.36 | 24.70 | 35.21 | 57.20 | 80.36 | 39.07 | 63.64 | 43.58 | 64.89 |
| | InstanceGaussian [23] | 55.11 | 79.50 | 30.55 | 42.87 | 61.23 | 78.11 | 36.84 | 62.54 | 45.93 | 65.76 |
| | ReferSplat [14] | 19.16 | 24.87 | 16.91 | 28.03 | 20.45 | 28.76 | 12.94 | 19.63 | 17.37 | 25.32 |
| | **Ours** | 57.47 | **86.80** | **40.07** | **62.38** | **70.11** | 86.50 | **42.73** | **67.10** | **52.60** | **75.70** |

where $w_a^{\text{sem}}$ and $w_b^{\text{sem}}$ are the semantic reliability weights of the endpoint nodes. If $\gamma_{ab} \geq \tau$, we add a directed edge $e_{ab}$ with relation label $\ell_{ab}$ and confidence $\gamma_{ab}$ to the relation graph.

### 3.4 Open-Vocabulary 3D Instance Querying

Given a natural language query $q$, a prompted LLM parser converts it into node-level and relation-level constraints:

$$\mathcal{Q}(q) = (\mathcal{Q}_{\text{node}}, \mathcal{Q}_{\text{rel}}) . \tag{7}$$

The node constraint $\mathcal{Q}_{\text{node}}$ follows the node attribute schema and may include category, color, material or shape, and function. The relation constraint is denoted by $\mathcal{Q}_{\text{rel}} = \{(p_n, \mathcal{Q}_{\text{ref},n})\}_{n=1}^{L}$, where $p_n$ is a normalized relation predicate, $\mathcal{Q}_{\text{ref},n}$ describes the reference node, and $L$ is the number of relation constraints. The parser retains only information that can be mapped to node attributes or relation predicates. Subjective or weakly grounded modifiers are excluded from retrieval constraints. After parsing, we first match $\mathcal{Q}_{\text{node}}$ against node attributes $\mathbf{a}_v$ to obtain a candidate target set $\mathcal{C}_{\text{node}}$. When relation constraints are present, each candidate node is further checked against graph neighbors that satisfy the required predicates and reference-node descriptions, producing a relation-consistent candidate set $\mathcal{C}(q)$.

Finally, candidates in $\mathcal{C}(q)$ are reranked by CLIP text–visual similarity:

$$s_v^{\text{clip}} = \cos\left(E_{\text{text}}(q_{\text{core}}), \mathbf{T}_v\right) . \tag{8}$$

Here, $q_{\text{core}}$ denotes the parser-retained core query text, and $\mathbf{T}_v$ denotes the visual embedding of candidate node $v$. CLIP reranking improves robustness to open-vocabulary variations, including synonymous and near-synonymous category descriptions. It does not replace graph constraints; instead, it selects the visually most compatible instance among candidates that already satisfy the structured constraints, and it cannot reintroduce nodes that violate them. The final output is the Gaussian subset $\mathcal{G}_{v^*}$ of the highest-ranked candidate. Further implementation details are provided in Appendix A.2.

## 4 Experiment

**Datasets and metrics.** We evaluate RelGS on three benchmarks covering word-level selection, sentence-level referring segmentation, and 3D semantic segmentation. First, we use LERF-OVS [34] for word-level open-vocabulary 3D selection, reporting results under both 2D projection-based and direct 3D evaluation protocols. Second, we evaluate sentence-level referring segmentation on LERF-OVS [34], following the 2D evaluation protocol of ReferSplat [14]. Third, we conduct 19-class




Figure 3: Qualitative results of sentence-level 3D referring segmentation on LERF-OVS [34]. Predicted target regions are highlighted by red bounding boxes. FAIL indicates unsuccessful localization.

semantic segmentation on ScanNet [10] under a direct 3D point-level evaluation protocol, following OpenGaussian [46]. For 2D evaluation, the selected 3D Gaussians are rendered into ground-truth views and compared with 2D ground-truth masks. For 3D evaluation, predictions are compared directly with 3D annotations at the Gaussian or point level, depending on the benchmark, without any 2D projection. We report mIoU and mAcc throughout.

**Baselines.** We compare against two families of representative methods: distillation-based approaches including LangSplat [34], LangSplatV2 [25], LEGaussians [39], and Feature-3DGS [50]; and clustering- or indexing-based approaches including GS-Grouping [47], OpenGaussian [46], InstanceGaussian [23], VoteSplat [16], Dr.Splat [17], and ReferSplat [14].

### 4.1 Word-Level Open-Vocabulary 3D Object Selection

Given a single-word query, we encode it with the CLIP text encoder, compute its similarity with each node's visual embedding $\mathbf{T}_k$ defined in Sec. 3.1, and render the selected node into multiple views. Tab. 1 reports both 2D and 3D results on LERF-OVS. RelGS achieves the best overall performance under both evaluation protocols, with particularly strong gains on the object-dense *Figurines* and heavily occluded *Waldo_Kitchen*, indicating that node-level aggregation effectively suppresses cross-view noise while preserving semantic discrimination. The advantage becomes more evident in 3D, where RelGS delivers more reliable point-level localization across scenes. In contrast, LangSplat and LEGaussians suffer from the discriminability loss of compressed CLIP features, while post-hoc clustering methods such as OpenGaussian, InstanceGaussian, and Dr.Splat still inherit per-Gaussian semantic drift. By aggregating raw CLIP image features into $\mathbf{T}_k$ and learning instance embeddings, RelGS forms cross-view consistent retrieval units for more reliable 3D localization. RelGS does not lead 2D mAcc in every scene because mAcc does not penalize predictions that spill beyond the ground-truth region, whereas mIoU penalizes both missed and excess selection. A method biased toward over-selection can therefore score very high mAcc but low mIoU, as LangSplat and LEGaussian do on *Waldo_Kitchen*. RelGS returns a hard object-level subset with conservative boundaries, conceding a little mAcc in individual scenes for tighter selection; its mean 2D mAcc remains the best.

### 4.2 Sentence-Level Open-Vocabulary 3D Referring Segmentation

Sentence-level queries often contain attribute modifiers and spatial relations, such as object appearance, anchor objects, and relational constraints. Following Sec. 3.4, we decompose each query into node constraints $\mathcal{Q}_{\text{node}}$ and relation constraints $\mathcal{Q}_{\text{rel}}$, and retrieve the target node by combining structured matching with CLIP reranking (Eq. (8)). As shown in Tab. 2, RelGS achieves the best performance on all four scenes and substantially outperforms ReferSplat in both average mIoU and mAcc.

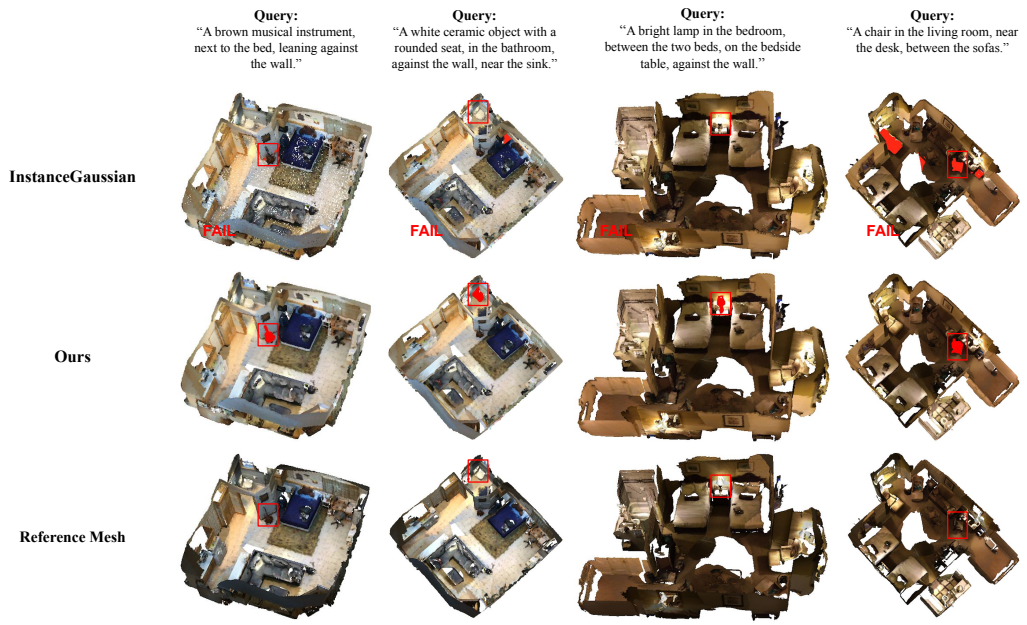



Figure 4: Qualitative results of sentence-level 3D referring segmentation on ScanNet [10]. Predicted target regions are highlighted by red bounding boxes. FAIL indicates unsuccessful localization.

Table 2: Comparison of sentence-level 3D referring segmentation on LERF-OVS [34]. Results are reported in mIoU and mAcc. Best results are in **bold**.

| | | Figurines | | Ramen | | Teatime | | Waldo_Kitchen | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Method | mIoU | mAcc | mIoU | mAcc | mIoU | mAcc | mIoU | mAcc | mIoU | mAcc |
| 2D evaluation | LangSplat [34] | 18.63 | 47.48 | 12.76 | 41.96 | 8.42 | 35.12 | 16.94 | 46.83 | 14.19 | 42.85 |
| | OpenGaussian [46] | 19.57 | 51.36 | 16.84 | 47.63 | 13.08 | 42.87 | 18.73 | 50.21 | 17.06 | 48.02 |
| | InstanceGaussian [23] | 18.43 | 48.18 | 13.05 | 42.73 | 9.47 | 36.84 | 16.82 | 47.51 | 14.44 | 43.82 |
| | ReferSplat [14] | 24.82 | 62.38 | 33.46 | 70.24 | 29.61 | 66.07 | 23.74 | 60.03 | 27.91 | 64.68 |
| | **Ours** | **46.83** | **81.64** | **55.96** | **88.37** | **52.23** | **85.59** | **28.62** | **73.18** | **45.91** | **82.20** |

These gains show the benefit of explicitly modeling spatial constraints with a 3D-statistics–derived relation graph, rather than relying only on CLIP similarity to handle prepositions such as *on*, *in*, and *under*. By combining spatial compatibility, LLM-based verification, and node reliability in Eq. (6), RelGS can preserve plausible relations under imperfect 3D node statistics while suppressing semantically inconsistent edges.

Tab. 4 further reports performance by relational hop count on LERF-OVS. RelGS exceeds ReferSplat on every reported subset, with the largest absolute margin on the 2-hop subset. The subsets differ in query composition, so we draw no causal conclusion about degradation with hop count. Fig. 3 further illustrates this advantage on compound queries that require first localizing an anchor object and then traversing the corresponding relation. In these cases, baseline methods often fail to identify the intended target, whereas the recursive relation traversal in Sec. 3.4 enables RelGS to localize the correct object more reliably.

### 4.3 Point Cloud Understanding and Click-Based Selection

**Point cloud understanding.** On ScanNet, we evaluate RelGS under the 19-class 3D point-level semantic segmentation protocol. As shown in Tab. 3, RelGS outperforms OpenGaussian and InstanceGaussian in both mIoU and mAcc, showing that the proposed node-level representation also benefits direct 3D semantic understanding. By aggregating multi-view semantic evidence into stable semantic nodes, RelGS reduces fragmented predictions and improves class-level consistency in complex indoor scenes. The qualitative examples in Fig. 4 show similar trends, where RelGS produces more complete and semantically coherent 3D localizations for objects under clutter, occlusion, and relational context.

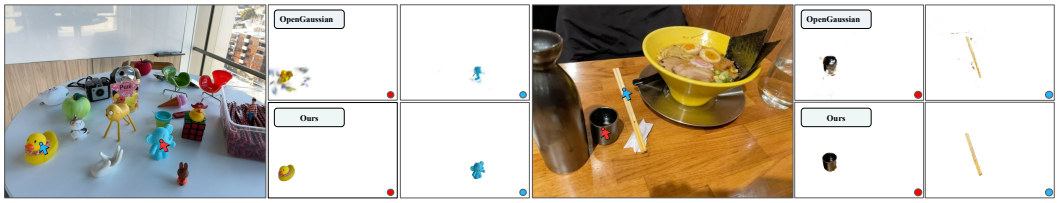



Figure 5: Qualitative comparison of click-based 3D object selection. Our method yields more complete and compact object-level selections, avoiding fragmented predictions and redundant regions.

Table 3: Quantitative evaluation on ScanNet [10]. 19-class semantic segmentation under the same 3D point-level protocol.

| Method | mIoU ↑ | mAcc ↑ |
|---|---|---|
| LangSplat [34] | 3.78 | 9.11 |
| OpenGaussian [46] | 24.73 | 41.54 |
| InstanceGaussian [23] | 40.66 | 54.01 |
| **Ours** | **43.72** | **56.10** |

Table 4: Query complexity on LERF-OVS [34], 2D mIoU. 0/1/2-hop: zero, one, or two relation-graph hops.

| Method | 0-hop ↑ | 1-hop ↑ | 2-hop ↑ |
|---|---|---|---|
| LangSplat [34] | 51.38 | 15.71 | 13.11 |
| ReferSplat [14] | 57.60 | 37.63 | 28.33 |
| **Ours** | **62.88** | **47.62** | **48.23** |

Table 5: Component ablation on LERF-OVS [34], sentence-level mIoU. **Node**: node learning; **Caption**: VLM enrichment; **Graph**: relation graph.

| Case | Node | Caption | Graph | mIoU ↑ |
|---|---|---|---|---|
| Base | | | | 27.91 |
| #1 | ✓ | | | 31.73 |
| #2 | ✓ | ✓ | | 38.60 |
| Full | ✓ | ✓ | ✓ | **45.91** |

Table 6: MLLM ablation for node attribute generation on LERF-OVS [34], sentence-level mean mIoU.

| MLLM | Size | mIoU ↑ |
|---|---|---|
| BLIP-2 [24] | 2.7B | 26.57 |
| LLaVA-1.5 [27] | 7B | 32.41 |
| Qwen2-VL [43] | 8B | 40.80 |
| Qwen3-VL-4B [2] | 4B | 41.76 |
| Qwen3-VL-8B [2] | 8B | **45.91** |

**Click-based selection.** We also evaluate interactive selection using the learned instance embedding $\mathbf{h}_i$. Given a click in any view, RelGS identifies the dominant Gaussian at the clicked pixel, retrieves its node, and renders the selected node across views, without additional online SAM prediction or feature extraction. As shown in Fig. 5, RelGS produces more complete and compact selections than OpenGaussian, avoiding fragmented regions and redundant object parts.

### 4.4 Ablation Study and Analysis

**Pipeline components.** We ablate the three core components of RelGS on LERF-OVS: semantic node learning, node attribute enrichment, and relation graph construction. As shown in Tab. 5, these components bring progressive improvements in mean mIoU. Semantic nodes provide cross-view consistent retrieval units, structured attributes impose explicit constraints beyond CLIP similarity, and the relation graph further supports spatial and compositional reasoning. This demonstrates that the three components contribute complementary cues for sentence-level 3D referring segmentation. Case #2 and Full share identical nodes, attributes, models, and reranking, so their difference isolates the net contribution of the relation graph.

**MLLM choice.** We further evaluate different MLLMs for node attribute generation while keeping all other components fixed. As shown in Tab. 6, the choice of MLLM has a clear impact on sentence-level referring segmentation. Compared with BLIP-2 and LLaVA-1.5, Qwen-based models produce stronger results, indicating that more reliable structured attribute generation benefits node verification and relation reasoning. Among them, Qwen3-VL-8B achieves the best performance, improving the mean mIoU to 45.91. This ablation replaces only the attribute model. The gain from the relation graph is positive under all five attribute models and grows with model capability, though not strictly monotonically (Tab. 15). The spread stems from filtering rather than description quality: BLIP-2 cannot reliably produce the required four-field JSON, so its malformed fields enter node matching as incorrect constraints and exclude the correct node, placing it below Case #1: wrong attributes are

Table 7: Direct evaluation of the relation graph on Ref-LERF [14] expressions (four LERF-OVS scenes, macro-averaged). mIoU: sentence-level.

| Graph construction variant | Triplet Rec. (%) | Predicate Acc. (%) | Edge Prec. (%) | # Edges | mIoU |
|---|---|---|---|---|---|
| Geometric cues only (no LLM verification) | 61.2 | 66.4 | 60.3 | 64 | 42.18 |
| + LLM verification, w/o reliability weight | 73.5 | 86.1 | 81.7 | 44 | 44.63 |
| Full RelGS (LLM verification + node reliability) | 72.7 | 88.9 | 87.6 | 39 | 45.91 |

Table 8: Efficiency on *teatime* (RTX A6000). Extra storage excludes the RGB 3DGS checkpoint.

| Method | Embed. training | One-time post-proc. | Word query | Sentence query | Peak GPU | Extra storage |
|---|---|---|---|---|---|---|
| LangSplat [34] | 1.7 h | autoencoder 32 min | 23 ms | 23 ms | 11.5 GB | 1977 MB |
| InstanceGaussian [23] | 1.3 h | clustering 46 min | 3 ms | 3 ms | 16.7 GB | 114 MB |
| **Ours** | 0.9 h | attributes + graph 4 min | 3 ms | LLM parse ≈1.5 s + 3 ms | 14.1 GB | 124 MB |

worse than none. With attributes fixed, replacing the shared verification-and-parsing model with a smaller open-source instruction model changes the result by clearly less than the graph contribution.

**Direct graph evaluation.** Since downstream mIoU cannot alone establish that relations are correct, Tab. 7 evaluates the graph itself against relations annotated from Ref-LERF expressions (protocol in Appendix A.4). LLM verification is the decisive step, markedly raising edge precision, predicate accuracy, and triplet recall. The reliability gate further improves precision, at a small drop in triplet recall that is not discriminative at the present sample size. Downstream mIoU moves in the same direction as graph quality.

**Node learning and reranking.** Replacing instance grouping with spatial proximity lowers all reported mIoU values, although this variant serves only as a spatial-only reference (Appendix A.4). After graph filtering, some queries still retain more than one candidate, because nodes may share attributes and relations. Selecting by graph confidence alone lowers sentence-level mIoU, but by clearly less than removing the relation graph.

**Efficiency.** In Tab. 8, RelGS has the shortest embedding training, and graph construction is a one-off cost shared by all query types. A word-level query invokes no language model, whereas a sentence-level query first requires one LLM parse, the price of structured query capability.

## 5 Conclusion

We presented RelGS, a relation-aware 3D Gaussian Splatting framework for open-vocabulary 3D scene understanding. RelGS moves beyond feature retrieval and query-conditioned mask prediction by organizing a 3DGS scene into reusable semantic nodes connected by typed relation edges, enabling explicit compositional reasoning over objects, attributes, and relations. It learns instance embeddings for individual Gaussians, groups them into cross-view consistent 3D nodes with aggregated CLIP embeddings and structured attributes, and constructs a confidence-weighted relation graph, where geometric cues propose candidate relations and an LLM verifies their semantic plausibility. During inference, attribute matching, CLIP retrieval, and graph traversal are jointly used to localize the target Gaussian subset. Extensive experiments on LERF-OVS and ScanNet demonstrate state-of-the-art performance on word-level selection, sentence-level referring segmentation, and 3D semantic segmentation, while leading on every reported multi-hop query subset. Despite these promising results, RelGS remains dependent on the quality of 3DGS reconstruction, camera poses, and multi-view coverage, and errors from foundation-model components may affect node construction and graph reasoning. Absolute performance is sensitive to language-model quality, particularly that of the attribute model, although the graph gain remains positive under all five tested models. The fixed six-predicate set expresses neither viewpoint-dependent relations nor multi-reference relations such as *between*; extending it requires no retraining or node reconstruction, only corresponding geometric cues and query mappings. Vertical relations such as *on* assume a consistent up axis, so incorrectly oriented inputs may cause errors in those relations. Extending relation-aware 3D semantic understanding to larger-scale, dynamic, and more complex real-world environments remains an important direction for future work.

## Acknowledgments

Financial support was provided by the Future Food Systems Cooperative Research Centre.

## References


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

[2] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[4] Y. Bhalgat, I. Laina, J. F. Henriques, A. Zisserman, and A. Vedaldi. N2f2: Hierarchical scene understanding with nested neural feature fields. In *European Conference on Computer Vision*, pages 197–214. Springer, 2024.

[5] J. Cen, J. Fang, C. Yang, L. Xie, X. Zhang, W. Shen, and Q. Tian. Segment any 3d gaussians. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 1971–1979, 2025.

[6] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022.

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

[8] G. Chen and W. Wang. A survey on 3d gaussian splatting. *ACM Computing Surveys*, 2024.

[9] S. Choi, H. Song, J. Kim, T. Kim, and H. Do. Click-gaussian: Interactive segmentation to any 3d gaussians. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[11] S. Dai, Y. Qu, Z. Li, X. Li, S. Zhang, and L. Cao. Training-free hierarchical scene understanding for gaussian splatting with superpoint graphs. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 3673–3682, 2025.

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

[13] S. He, P. Ji, Y. Yang, C. Wang, J. Ji, Y. Wang, and H. Ding. A survey on 3d gaussian splatting applications: Segmentation, editing, and generation. *arXiv preprint arXiv:2508.09977*, 2025.

[14] S. He, G. Jie, C. Wang, Y. Zhou, S. Hu, G. Li, and H. Ding. Refersplat: Referring segmentation in 3d gaussian splatting. *arXiv preprint arXiv:2508.08252*, 2025.

[15] S. Jang and W. Kim. Identity-aware language gaussian splatting for open-vocabulary 3d semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20467–20476, 2025.

[16] M. Jiang, S. Jia, J. Gu, X. Lu, G. Zhu, A. Dong, and L. Zhang. Votesplat: Hough voting gaussian splatting for 3d scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6456–6465, 2025.

[17] K. Jun-Seong, G. Kim, K. Yu-Ji, Y.-C. F. Wang, J. Choe, and T.-H. Oh. Dr. splat: Directly referring 3d gaussian splatting via direct language embedding registration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14137–14146, 2025.

[18] B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[19] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19729–19739, 2023.

[20] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[21] S. Kobayashi, E. Matsumoto, and V. Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in neural information processing systems*, 35:23311–23330, 2022.

[22] S. Koch, N. Vaskevicius, M. Colosi, P. Hermosilla, and T. Ropinski. Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14193, 2024.

[23] H. Li, Y. Wu, J. Meng, Q. Gao, Z. Zhang, R. Wang, and J. Zhang. Instancegaussian: Appearance-semantic joint gaussian representation for 3d instance-level perception. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14078–14088, 2025.

[24] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[25] W. Li, Y. Zhao, M. Qin, Y. Liu, Y. Cai, C. Gan, and H. Pfister. Langsplatv2: High-dimensional 3d language gaussian splatting with 450+ fps. *arXiv preprint arXiv:2507.07136*, 2025.

[26] S. Linok, T. Zemskova, S. Ladanova, R. Titkov, D. Yudin, M. Monastyrny, and A. Valenkov. Beyond bare queries: Open-vocabulary object grounding with 3d scene graph. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13582–13589. IEEE, 2025.

[27] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[28] Z. Liu, Y. Wang, S. Zheng, T. Pan, L. Liang, Y. Fu, and X. Xue. Reasongrounder: Lvlm-guided hierarchical feature splatting for open-vocabulary 3d visual grounding and reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3718–3727, 2025.

[29] Z. Liu, S. Zheng, S. Chen, C. Zhao, L. Liang, X. Xue, and Y. Fu. A neural representation framework with llm-driven spatial reasoning for open-vocabulary 3d visual grounding. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 1042–1051, 2025.

[30] W. Lyu, X. Li, A. Kundu, Y.-H. Tsai, and M.-H. Yang. Gaga: Group any gaussians via 3d-aware memory bank. *arXiv preprint arXiv:2404.07977*, 2024.

[31] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[32] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[33] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[34] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024.

[35] Y. Qu, S. Dai, X. Li, J. Lin, L. Cao, S. Zhang, and R. Ji. Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 5328–5337, 2024.

[36] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[37] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[38] C. Shi, M. Chen, Y. Mao, C. Yang, X. Hu, J. Ding, and Z. Yu. Realm: An mllm-agent framework for open world 3d reasoning segmentation and editing on gaussian splatting. *arXiv preprint arXiv:2510.16410*, 2025.

[39] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024.

[40] X. Tan, Y. Ji, H. Zhu, and Y. Xie. Fmlgs: Fast multilevel language embedded gaussians for part-level interactive agents. *arXiv preprint arXiv:2504.08581*, 2025.

[41] L. Tian, X. Li, L. Ma, H. Yin, Z. Zheng, H. Huang, T. Li, H. Lu, and X. Jia. Ccl-lgs: Contrastive codebook learning for 3d language gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9855–9864, 2025.

[42] J. Wald, H. Dhamo, N. Navab, and F. Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3961–3970, 2020.

[43] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[44] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

[45] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7515–7525, 2021.

[46] Y. Wu, J. Meng, H. Li, C. Wu, Y. Shi, X. Cheng, C. Zhao, H. Feng, E. Ding, J. Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024.

[47] M. Ye, M. Danelljan, F. Yu, and L. Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European conference on computer vision*, pages 162–179. Springer, 2024.

[48] H. Ying, Y. Yin, J. Zhang, F. Wang, T. Yu, R. Huang, and L. Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024.

[49] H. Zhai, H. Li, Z. Li, X. Pan, Y. He, and G. Zhang. Panogs: Gaussian-based panoptic segmentation for 3d open vocabulary scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14114–14124, 2025.

[50] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024.

[51] X. Zuo, P. Samangouei, Y. Zhou, Y. Di, and M. Li. Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *International Journal of Computer Vision*, 133(2):611–627, 2025.

# A Appendix

## A.1 Implementation Details

### A.1.1 Training Time

We train each scene on a single NVIDIA RTX A6000 GPU. The instance-embedding node learning stage (Sec. 3.1) is trained for $30{,}000$ iterations, taking approximately $1$ hour per scene on LERF-OVS and approximately $30$ minutes per scene on ScanNet. Node attribute enrichment (Sec. 3.2) and relation graph construction (Sec. 3.3) are one-time post-processing steps requiring approximately 2–5 minutes per scene, with no recomputation needed at inference time. At query time, a word-level query invokes no language model and encodes the word with the CLIP text encoder. A sentence-level query first requires one LLM parse (Eq. (7)), followed by the structured stage of attribute matching against $\mathcal{Q}_{\text{node}}$, graph traversal under $\mathcal{Q}_{\text{rel}}$, and CLIP reranking (Eq. (8)), whose latencies are reported separately in Tab. 8. In that table, extra storage covers the language and instance fields plus precomputed caches, and both baselines operate at the vocabulary level, so their two query columns coincide.

### A.1.2 ScanNet Dataset Evaluation

For ScanNet evaluation, we use $9$ scenes, namely `scene0000_00`, `scene0062_00`, `scene0070_00`, `scene0097_00`, `scene0200_00`, `scene0347_00`, `scene0400_00`, `scene0590_00`, and `scene0645_00`. We use the $19$ semantic categories defined by ScanNet as text queries: `wall`, `floor`, `cabinet`, `bed`, `chair`, `sofa`, `table`, `door`, `window`, `bookshelf`, `picture`, `counter`, `desk`, `curtain`, `refrigerator`, `shower curtain`, `toilet`, `sink`, and `bathtub`. For each category query, the model returns the corresponding 3D Gaussian subset or semantic node prediction, which is then evaluated against ScanNet ground-truth semantic labels. We report standard 3D point-level metrics, including mean Intersection-over-Union (mIoU) and mean accuracy (mAcc). When necessary, Gaussian predictions are transferred to ScanNet points by nearest-neighbour assignment in 3D space.

## A.2 Method Supplementary

This section supplements the main method (Sec. 3) with implementation-level details that are referenced by, but not described in, the main paper.

**Pre-trained 3DGS and learnable embeddings.** Consistent with prior 3DGS-based open-vocabulary works, we assume each scene is provided with a pre-trained 3DGS reconstruction obtained under the LangSplat protocol, and attach an $\ell_2$-normalized instance embedding $\mathbf{h}_i \in \mathbb{R}^{32}$ ($\|\mathbf{h}_i\|_2 = 1$) to each Gaussian (Sec. 3.1). The embedding is trained for $30{,}000$ iterations under the loss in Eq. (1). During this stage, the original 3DGS parameters $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, $\alpha_i$, and $\mathbf{c}_i$ remain optimized under the RGB reconstruction objective to preserve geometry and appearance, while the SAM supervision affects only $\mathbf{h}_i$ and propagates no gradient to geometry. This decoupling lets RelGS act directly on any 3DGS pre-trained under the LangSplat protocol.

**SAM, CLIP, and cross-view mask tracking.** We adopt SAM (ViT-H) for region proposals and OpenCLIP ViT-B/16 for masked-region encoding. While LangSplat extracts three SAM granularity levels (small, middle, large), our implementation uses only the large level, since coarser regions better align with object-level semantic nodes. Since SAM produces masks independently per view, we use the zero-shot video tracker SAM2 [37] to propagate masks from the first view to all other views. In each target view, propagated masks are matched to local SAM masks by IoU; matches with IoU $> 0.5$ inherit a globally consistent identity $k$, while unmatched local masks (IoU $\leq 0.5$) are excluded from per-mask supervision in Eq. (1). Both instance terms of Eq. (1) are defined within a single view; the cross-view coupling arises because $\mathbf{h}_i$ is a single per-Gaussian vector rendered into every visible view, so mask geometry consistency on the shared 3D field is the operative factor. The consistent identities further define the visible view set $\mathcal{V}_k$, the representative view selection of Eq. (2), and the node representation $\mathbf{T}_k$. Each mask-level CLIP feature $\mathbf{t}_{s,k}$ is view-dependent by design; cross-view aggregation takes place when $\mathbf{T}_k$ is formed after training, not inside the loss.

**Feature rendering and $\ell_2$ normalization in Eq. (1).** The pixel-level instance feature $\hat{\mathbf{H}}_s(u)$ is obtained by the same differentiable alpha compositing as the color rendering of 3DGS, but applied to the per-Gaussian embedding $\mathbf{h}_i$ instead of $\mathbf{c}_i$, with stop-gradient composition weights:

$$\hat{\mathbf{H}}_s(u) = \sum_{i \in \mathcal{N}(u)} \mathrm{sg}(\beta_i\, \alpha_i)\, \mathbf{h}_i, \qquad \beta_i = \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{9}$$

where $\mathcal{N}(u)$ is the depth-sorted set of Gaussians contributing to pixel $u$ in view $I_s$, $\mathrm{sg}(\cdot)$ is the stop-gradient operator, and $\beta_i$ is the accumulated transmittance, written here as $\beta$ to distinguish it from the node visual embedding $\mathbf{T}_k$. The per-Gaussian instance embedding $\mathbf{h}_i$ is kept on the unit sphere. The composition weights are non-negative and sum to $\sum_i \beta_i \alpha_i = 1 - \beta_{\mathrm{end}} \leq 1$, where $\beta_{\mathrm{end}}$ is the transmittance remaining after the last Gaussian, so the rendered feature is a sub-convex combination of unit vectors: $\hat{\mathbf{H}}_s(u)$ is not necessarily a unit vector, but lies in the same embedding space. For Eq. (1), $\bar{\mathbf{H}}_{s,k}$ is $\ell_2$-normalized before the instance separation hinge, so that the margin $m \in [-1, 1]$ is a cosine margin. The instance compactness term keeps the squared-Euclidean form, following standard practice in language-embedded 3DGS works [34, 46], since the rendered $\hat{\mathbf{H}}_s(u)$ stays close to $\bar{\mathbf{H}}_{s,k}$ within a single mask and the two formulations are locally equivalent up to a positive constant.

**Node construction.** After convergence, the $k$-NN graph mentioned at the end of Sec. 3.1 uses $k = 16$ nearest neighbours per Gaussian and the joint similarity

$$s_{ij} = w_{\mathrm{inst}} \cdot \cos(\mathbf{h}_i, \mathbf{h}_j) + w_{\mathrm{spa}} \cdot \exp\!\left(-\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2 / \sigma\right), \tag{10}$$

with $w_{\mathrm{inst}} = 0.7$, $w_{\mathrm{spa}} = 0.3$, and bandwidth $\sigma$ adaptively set to the median nearest-neighbour Gaussian-to-Gaussian distance in the scene (i.e., the median over $\{\min_{j \neq i} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2\}_i$). Edges with $s_{ij} < 0.5$ are pruned before extracting connected components. Components containing fewer than $N_{\min} = 50$ Gaussians, or whose visible-view set covers less than $\eta = 2\%$ of the training cameras, are discarded as noise; the remaining components form the node set $V$. The node visual embedding $\mathbf{T}_k$ is the $\ell_2$-normalized mean of mask-level CLIP features $\{\mathbf{t}_{s,k}\}_{s \in \mathcal{V}_k}$ over the visible views.

**View selection and VLM call.** In Eq. (2) we use $N_v = 10$ representative views per node. The corresponding masked crops (with a context margin of $10\%$ of the mask bounding-box side length) are passed in a single VLM call, which returns the four-field attribute schema $\mathbf{a}_k = (\texttt{category}, \texttt{color}, \texttt{material/shape}, \texttt{function})$ together with a short caption $c_k$ in strict JSON format (see the prompt in Tab. 9). By default, we use Qwen3-VL-8B as the VLM, which gives the best performance in Tab. 6. The reverse CLIP threshold in Eq. (3) is fixed to $\tau_{\mathrm{sem}} = 0.22$ across all experiments. Nodes with $w_k^{\mathrm{sem}} = 0$ are kept in the graph and remain available for direct attribute matching and CLIP retrieval, but contribute zero weight to incident edges through Eq. (6), removing the need for any extra threshold-based pruning.

**Soft spatial compatibility $\gamma_{ab}^{\mathrm{sp}}$.** The soft spatial compatibility $\gamma_{ab}^{\mathrm{sp}} \in [0, 1]$ in Eq. (6) is computed from the geometric cues underlying $\mathrm{RelSym}(a, b)$. Let $z_a^c$ and $z_b^c$ be the up-axis coordinates of $\mathbf{x}_a^c$ and $\mathbf{x}_b^c$, $\Delta z_{ab} = z_a^c - z_b^c$, $d_{ab}^{xy}$ the in-plane (XY) distance, $d_{ab}^{\mathrm{3D}} = \|\mathbf{x}_a^c - \mathbf{x}_b^c\|_2$, $\mathrm{Ovl}(B_a, B_b)$ the AABB overlap ratio, and $\bar{r} = \frac{1}{2}(r_a + r_b)$. We define

$$\gamma_{ab}^{\mathrm{sp}} = \begin{cases} \mathrm{Ovl}(B_a, B_b), & \mathrm{RelSym} = \texttt{INSIDE}, \\ \exp(-d_{ab}^{xy} / \bar{r}), & \mathrm{RelSym} \in \{\texttt{ABOVE}, \texttt{BELOW}\}, \\ \exp(-d_{ab}^{\mathrm{3D}} / (r_a + r_b)) \cdot \big(\mathrm{Ovl}(B_a, B_b) + \epsilon\big), & \mathrm{RelSym} = \texttt{ATTACHED}, \\ \exp(-d_{ab}^{\mathrm{3D}} / (\rho\, \bar{r})), & \mathrm{RelSym} = \texttt{ADJACENT}, \end{cases} \tag{11}$$

with $\epsilon = 0.05$. This continuous score acts as a soft constraint that tolerates imperfect 3D statistics, while the final relation label $\ell_{ab}$ is determined by the LLM in symbolic space (Eq. (5)).

$\mathrm{RelSym}$ **discretization thresholds.** The five symbolic spatial cues are derived from the continuous descriptors as follows. `INSIDE` is assigned when $\mathrm{Ovl}(B_a, B_b) > 0.6$ and $\mathrm{vol}(B_a) < 0.5\,\mathrm{vol}(B_b)$; `ABOVE`/`BELOW` when $|\Delta z_{ab}| > 0.5(r_a + r_b)$ and the in-plane overlap is below $0.1$, with the sign of $\Delta z_{ab}$ deciding between the two; `ATTACHED` when $\mathrm{Ovl}(B_a, B_b) \in (0.05,\, 0.4]$ and $d_{ab}^{\mathrm{3D}} < r_a + r_b$; the remaining pairs within the proximity neighborhood of Eq. (4) are labelled as `ADJACENT`.

**Predicate set $\mathcal{R}$ and edge threshold.** The candidate predicate set is $\mathcal{R} = \{\texttt{on}, \texttt{in}, \texttt{under}, \texttt{attached_to}, \texttt{part_of}, \texttt{near}\}$, covering the part-whole, attachment, containment, support, vertical, and proximity relations referenced in Sec. 3.3. The same set is used for relation verification (Eq. (5)) and for normalising natural-language relation phrases during query parsing (Eq. (7); see Tab. 11). The candidate-pair neighborhood scale in Eq. (4) is $\rho = 2.0$. The final relation graph $\mathcal{K} = \langle V, E \rangle$ retains all directed edges $(a, \ell_{ab}, b, \gamma_{ab})$ with $\gamma_{ab} \geq \tau$, where $\tau = 0.35$.

**Query parsing and graph traversal.** The parser in Eq. (7) is implemented as an LLM call with the prompt in Tab. 11. The match between $\mathcal{Q}_{\text{node}}$ and node attributes $\mathbf{a}_v$ proceeds field-by-field over the four-field schema: `category` and `color` are used for recall, while `material/shape` and `function` are used for filtering; any field left unspecified in $\mathcal{Q}_{\text{node}}$ is skipped, yielding the candidate target set $\mathcal{C}_{\text{node}}$. When $\mathcal{Q}_{\text{rel}} \neq \emptyset$, we check, for every $v \in \mathcal{C}_{\text{node}}$ and every relation constraint $(p_n, \mathcal{Q}_{\text{ref},n})$, whether $v$ has a directed neighbour in $\mathcal{K}$ with predicate $p_n$ whose attributes match $\mathcal{Q}_{\text{ref},n}$. For multi-hop queries ($L > 1$), the procedure is applied recursively along the relation chain, producing the relation-consistent candidate set $\mathcal{C}(q)$. Finally, $\mathcal{C}(q)$ is reranked by Eq. (8) using $q_{\text{core}}$, and the Gaussian subset $\mathcal{G}_{v^*}$ of the top-1 candidate is returned.

**LLM and VLM configuration.** RelGS makes three language-model calls: attribute generation (Sec. 3.2), relation verification (Eq. (5)), and query parsing (Eq. (7)), all of which default to Qwen3-VL-8B. Tab. 6 replaces only the first. The relation verifier and the query parser share the same LLM, used in a single-turn, zero-shot configuration with greedy decoding to maximize structured-output reliability. Full system prompts for the VLM and the LLM are provided in Sec. A.3.

### A.3 Prompt Templates

We release the full system prompts of the three VLM/LLM calls used by RelGS for reproducibility.

Table 9: VLM node-attribute extraction prompt (Sec. 3.2, generation of $\mathbf{a}_k$ and $c_k$).

```
NODE_ATTRIBUTE_SYSTEM_PROMPT = """You are a structured attribute
extractor for 3D semantic nodes in an open-vocabulary scene
understanding pipeline.

INPUTS
- A grid of N_v cropped multi-view images of ONE 3D object node;
  the masked region is highlighted, the outside area is darkened.
- The same object appears in all crops; do NOT describe occluders
  or background.

SCOPE
- Describe ONLY the highlighted object across views.
- Be factual; do not guess hidden details.
- Use view-independent wording; never use left/right/front/back/
  top/bottom or any camera-relative term.

ATTRIBUTE FIELDS (must all be filled)
1) category       : a concrete singular class noun (e.g., "rubber
                    duck", "ceramic mug", "wooden chair"). Forbidden
                    generics: object/thing/item/stuff/region/area.
2) color          : 1-3 dominant colors, comma-separated.
3) material_shape : material AND geometric shape, joined by " / "
                    (e.g., "ceramic / cylindrical with handle").
4) function       : the typical function or affordance of the object
                    (e.g., "container for liquids", "sitting surface").

CAPTION FIELD
- caption : ONE sentence, 12-20 words, view-independent,
            comma-separated phrases, no period. Begin with the
            category noun, then appearance details, then a function
            or part-of clause. Do NOT include inter-object spatial
            relations (no "next to", "on the table", "above the lamp"
            etc.); inter-node relations are constructed separately.
```

```
PRINTED-CONTENT RULE
- Scan ALL crops within the highlighted region for visible text,
  logos, or printed characters.
- If any are visible, transcribe the readable tokens exactly,
  preserving case and punctuation; if uncertain, describe the
  graphic neutrally (e.g., "yellow cartoon character with red
  cheeks").

OUTPUT (STRICT)
Return EXACTLY ONE JSON line, no extra text, no code fences,
no reasoning:
{"category": "...", "color": "...", "material_shape": "...",
 "function": "...", "caption": "..."}

CONSTRAINTS
- No chain-of-thought.
- Do not infer the location of the object in the room.
- If any field cannot be determined visually, write "unknown"
  for that field; never hallucinate.
"""
```

Table 10: LLM relation verification prompt (Eq. (5)).

```
RELATION_VERIFY_SYSTEM_PROMPT = """You verify a candidate spatial
relation between two 3D semantic nodes A and B in a scene graph,
based on their structured attributes and a symbolic 3D spatial cue.

INPUTS
- attr_a    : JSON of node A's attributes
              {category, color, material_shape, function, caption}.
- attr_b    : JSON of node B's attributes (same schema).
- candidates: the predicate set R =
              {on, in, under, attached_to, part_of, near}.
- rel_sym   : ONE symbolic 3D spatial cue from
              {INSIDE, ABOVE, BELOW, ATTACHED, ADJACENT},
              describing the geometric relation of A relative to B.
- direction : the candidate edge is directed from A to B.

POLICY
- Treat rel_sym as a strong geometric prior; the chosen predicate
  must be CONSISTENT with rel_sym:
    INSIDE   -> in or part_of
    ABOVE    -> on (if A is supported by B) or near
    BELOW    -> under or near
    ATTACHED -> attached_to or part_of
    ADJACENT -> near
- Use semantic plausibility from attr_a and attr_b to disambiguate
  (e.g., a "drawer" ATTACHED to a "cabinet" -> part_of;
         a "magnet" ATTACHED to a "fridge"  -> attached_to;
         a "duck"   ABOVE   a "Rubik's cube" -> on;
         a "lamp"   ABOVE   a "shelf" without contact -> near).
- If no candidate is plausible, return predicate = "none".

OUTPUT (STRICT)
Return EXACTLY ONE JSON line, no extra text, no code fences,
no reasoning:
{"predicate": "<one of candidates or 'none'>",
 "confidence": <float in [0,1]>,
 "rationale_tag": "<one short tag, e.g., 'support_geometry',
                   'intrinsic_part', 'magnetic_attachment',
                   'no_contact'>"}

CONSTRAINTS
- No chain-of-thought; only the JSON line.
- Confidence MUST reflect both rel_sym consistency AND semantic
  plausibility; mismatched cases should be < 0.4.
- Do not invent attributes beyond those provided.
```

```
"""
```

Table 11: LLM query parsing prompt (Eq. (7)).

```
QUERY_PARSE_SYSTEM_PROMPT = """You parse an open-vocabulary 3D
referring query into a structured constraint set for a relation-aware
scene graph.

INPUTS
- query         : a free-form natural-language description of ONE
                  target object in a 3D scene, possibly mentioning
                  attributes and spatial relations.
- predicate_set : R = {on, in, under, attached_to, part_of, near}.

OUTPUT SCHEMA
- node      : constraints on the TARGET node, aligned with the node
              attribute schema.
              {
                "category"      : "<class noun or 'unknown'>",
                "color"         : "<color or 'unknown'>",
                "material_shape": "<material/shape phrase or
                                   'unknown'>",
                "function"      : "<function/affordance or
                                   'unknown'>"
              }
- relations : ordered list of relation constraints
              (target -> reference). Each item:
              {
                "predicate" : "<one of predicate_set>",
                "reference" : { same schema as 'node' }
              }
- core_query: a short, attribute-only phrase used for CLIP reranking,
              with all relational prepositions and reference-object
              descriptions REMOVED. 4-10 words, lowercase,
              no punctuation.

PROTOCOL
- Map only content that aligns with the node-attribute schema or the
  predicate set. Subjective or weakly grounded modifiers (e.g.,
  "cute", "interesting") MUST be dropped.
- Normalize prepositions:
    "on top of" / "sitting on" / "above (in contact)" -> on
    "inside"    / "within"      / "in"                -> in
    "below"     / "underneath"                        -> under
    "attached to" / "stuck on" / "mounted on" /
    "hanging from"                                    -> attached_to
    "part of"   / "belongs to"                        -> part_of
    "near" / "beside" / "next to" / "adjacent to"     -> near
- If the query has multiple relations (multi-hop), preserve them in
  order from the target outward to the outer-most reference.
- If a field is not present in the query, set it to "unknown".

OUTPUT (STRICT)
Return EXACTLY ONE JSON line, no extra text, no code fences,
no reasoning:
{"node": {...}, "relations": [...], "core_query": "..."}

CONSTRAINTS
- No chain-of-thought.
- Never invent attributes or relations not present in the query.
- The 'core_query' MUST NOT contain any relational preposition or
  any reference-object description; it describes ONLY the target.
"""
```

### A.4 Additional Evaluations

This section reports the protocol and the supporting analyses referenced by Sec. 4.4. Unless stated otherwise, all settings are fixed at their defaults.

**Relation-graph annotation protocol.** Ref-LERF [14] provides referring expressions and object masks on the four LERF-OVS scenes, but no relation annotations, so the reference triplets of Tab. 7 are our own. We use Ref-LERF purely as test material, and use neither its model, its weights, nor its training pipeline. A deterministic lexical screen, which makes no language-model call and does not reuse the output of the query parser (Eq. (7)), first identified candidate expressions from the full expression set under the six-predicate ontology, using the lexical normalization rules in Tab. 11 and, where geometric cues are involved, the predicate definitions in Tab. 10. Two annotators, one author and one non-author, then independently checked all candidate expressions and all extracted target–predicate–reference instances, excluding uses that describe a scene region or a location rather than an object-to-object relation (e.g., "in the bathroom" and "in the center of the table"); only labels agreed upon by both annotators were retained. Tab. 12 lists the resulting relation-bearing expressions and reference relation instances. One expression uses *between*, a two-reference relation that the ontology does not cover; we record it as *near*. An external scene-graph benchmark was not used because transfer would require additional node matching and label mapping, folding mapping error into the conclusion.

Edge Precision was judged by the same two annotators, following the general evaluation approach of ConceptGraphs [12], though not its three-annotator majority protocol. Its denominator is the set of predicted edges, while # Edges reports the number of retained predicted edges per scene. Triplet Recall and Predicate Accuracy are computed against the reference relation instances, with one predicted predicate per directed edge and no R@$k$ ranking, and are macro-averaged over the four scenes. Query parsing is fixed to the default model across all variants.

Table 12: Reference relation annotations per LERF-OVS scene.

| | Figurines | Ramen | Teatime | Waldo_Kitchen | Total |
|---|---|---|---|---|---|
| Relation-bearing expressions | 49 | 51 | 41 | 56 | 197 |
| Reference relation instances | 52 | 59 | 51 | 64 | 226 |

**Separated ablation of node learning.** Tab. 13 isolates the node-learning embeddings. We also evaluated an auxiliary per-Gaussian semantic embedding supervised by mask-level CLIP targets. It leaves all three metrics unchanged at the reported precision: it lies on no inference path, since nodes are grouped from $\mathbf{h}_i$ and spatial proximity, word- and sentence-level retrieval uses $\mathbf{T}_k$, and click-based selection uses $\mathbf{h}_i$; and it shares no gradient with $\mathbf{h}_i$, because the composition weights are stop-gradient. RelGS therefore omits it. The spatial-proximity variant also adjusts the spatial weight and the connectivity criterion, so it is not a single-variable swap, and we present it only as a spatial-only reference against full instance grouping.

Table 13: Separated ablation of the node-learning embeddings.

| Variant | Word-level 3D mIoU | Sentence-level mIoU | ScanNet 19-class mIoU |
|---|---|---|---|
| RelGS ($\mathbf{h}_i$ + $\mathbf{T}_k$) | 52.60 | 45.91 | 43.72 |
| RelGS + auxiliary semantic embedding | 52.60 | 45.91 | 43.72 |
| Spatial proximity instead of $\mathbf{h}_i$ | 45.38 | 36.12 | 38.04 |

**Cross-view tracking and mask filtering.** Tab. 14 shows that SAM2 propagation and IoU filtering each improve the results. Beyond making supervision compatible on the shared 3D field, the filtering excludes masks without an IoU match from per-mask supervision.

Table 14: Ablation of cross-view tracking and mask filtering on LERF-OVS.

| Mask handling | Nodes per scene | Word-level 3D mIoU | Sentence-level mIoU |
|---|---|---|---|
| Per-view SAM masks only | 59 | 48.14 | 40.72 |
| SAM2 propagation, no IoU filtering | 53 | 50.86 | 43.57 |
| SAM2 + IoU filtering (full) | 46 | 52.60 | 45.91 |

**Graph gain under different attribute models.** Tab. 15 runs the graph-free and full-graph configurations for every attribute MLLM of Tab. 6. The relation graph improves performance under all five models with their ranking unchanged, so absolute accuracy is bounded by the attribute model, whereas the gain from the graph is not tied to any specific one.

Table 15: Gain from the relation graph under different attribute MLLMs (LERF-OVS, sentence-level mIoU).

| Attribute MLLM | w/o relation graph | Full relation graph | Gain |
|---|---|---|---|
| BLIP-2 | 23.94 | 26.57 | +2.63 |
| LLaVA-1.5 | 28.67 | 32.41 | +3.74 |
| Qwen2-VL | 35.23 | 40.80 | +5.57 |
| Qwen3-VL-4B | 36.44 | 41.76 | +5.32 |
| Qwen3-VL-8B (default) | 38.60 | 45.91 | +7.31 |

**Node granularity.** RelGS nodes are open-vocabulary, class-agnostic 3D grouping units covering all scene content, so their number inherently exceeds the number of annotated query targets. The value of $k$ sets the candidate neighbourhood of each Gaussian, while the node partition is further constrained by the similarity threshold and pruning of Appendix A.2. A small $k$ therefore tends to fragment complete objects, a large $k$ adds cross-object edges, and the intermediate range is stabilized by the threshold. In Tab. 16, the number of nodes decreases monotonically with $k$, confirming that $k$ controls granularity, whereas sentence-level mIoU varies only slightly and is best at the default value, indicating that the similarity threshold and pruning rules already secure grouping quality.

Table 16: Sensitivity to $k$ on LERF-OVS, with all other settings fixed.

| $k$ | Nodes per scene | Sentence-level mIoU | $\Delta$ vs. default |
|---|---|---|---|
| 8 | 62 | 44.72 | $-1.19$ |
| 12 | 52 | 45.48 | $-0.43$ |
| 16 (default) | 46 | 45.91 | 0.00 |
| 24 | 40 | 45.37 | $-0.54$ |
| 32 | 34 | 44.61 | $-1.30$ |

**Reconstruction under node learning.** During node learning, geometry, opacity, and color continue to be optimized by the RGB objective. In Tab. 17, all three reconstruction metrics improve, yet the number of Gaussians increases slightly, so no compression occurs. We do not attribute the improvement to instance supervision, because geometry receives only RGB gradients; it follows from the continued optimization of the RGB objective. The result we can establish is a negative one: adding groupable instance structure costs neither the underlying reconstruction nor an inflated representation size. Compactness is not an objective of this design, since semantic supervision acting on geometry would break the comparability of the underlying reconstruction with the baselines.

Table 17: Reconstruction quality before and after node learning (*teatime*).

| Variant | PSNR ↑ | SSIM ↑ | LPIPS (VGG) ↓ | # Gaussians |
|---|---|---|---|---|
| Pre-trained 3DGS (before node learning) | 30.59 | 0.9175 | 0.2325 | 1,912,041 |
| After node learning (ours) | 30.82 | 0.9263 | 0.2249 | 1,927,872 |

**Representative failure cases.** Three failure modes follow from the analyses above. A weak attribute model injects incorrect symbolic constraints and excludes the correct node during attribute matching, as with the malformed BLIP-2 fields in Tab. 6. A granularity mismatch between the VLM and the query, such as cup versus mug, leaves ambiguity for reranking to resolve. Finally, incorrectly oriented inputs may cause errors in vertical relations such as *on*, because these rely on the up-axis assumption.

### A.5 More Visual Results

To further demonstrate the effectiveness of RelGS, we provide additional qualitative results on LERF-OVS [34] and ScanNet [10]. Fig. 6 shows word-level 3D referring segmentation results on

LERF-OVS, where RelGS accurately localizes target objects from category-level queries. Fig. 7 presents more challenging sentence-level referring segmentation results on LERF-OVS, involving attributes and spatial relations. Fig. 8 further shows qualitative results on ScanNet, demonstrating that RelGS can generalize to point-cloud-based indoor scenes and handle open-vocabulary relational queries in 3D. Across these examples, RelGS produces more complete and compact target regions, while reducing failures caused by fragmented predictions or relation ambiguity.

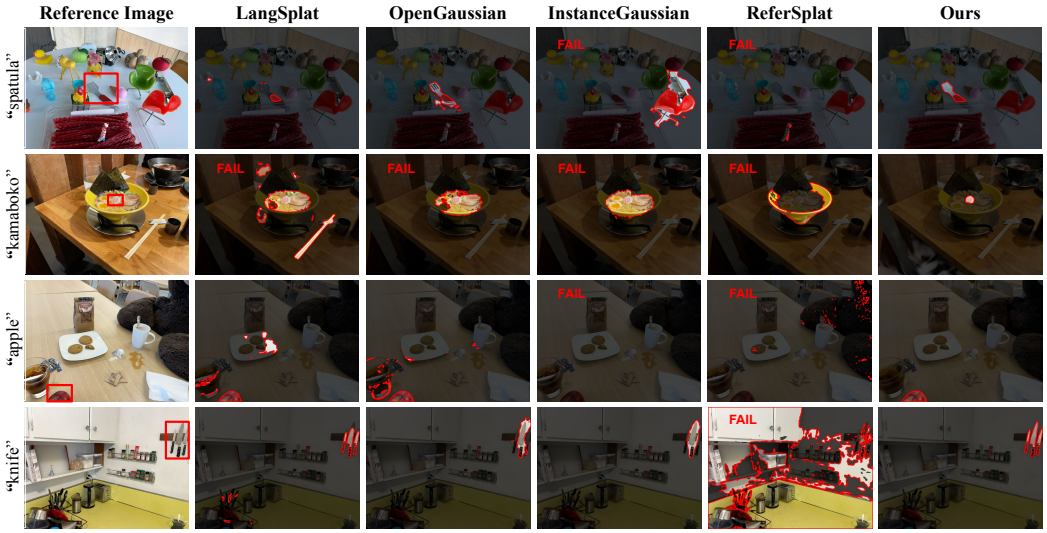



Figure 6: Qualitative results of word-level 3D referring segmentation on LERF-OVS [34]. Predicted target regions are highlighted by red bounding boxes. FAIL indicates unsuccessful localization.

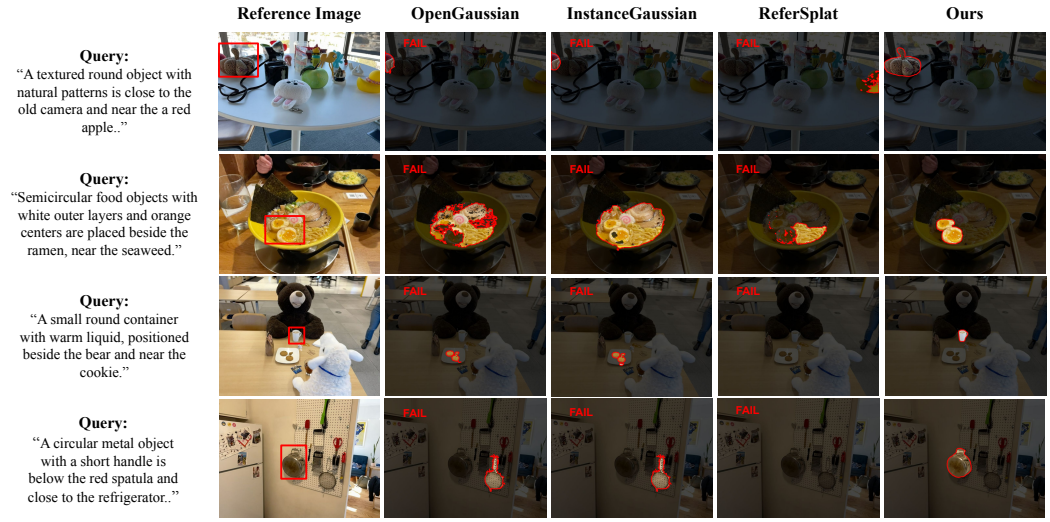



Figure 7: Qualitative results of sentence-level 3D referring segmentation on LERF-OVS [34]. Predicted target regions are highlighted by red bounding boxes. FAIL indicates unsuccessful localization.

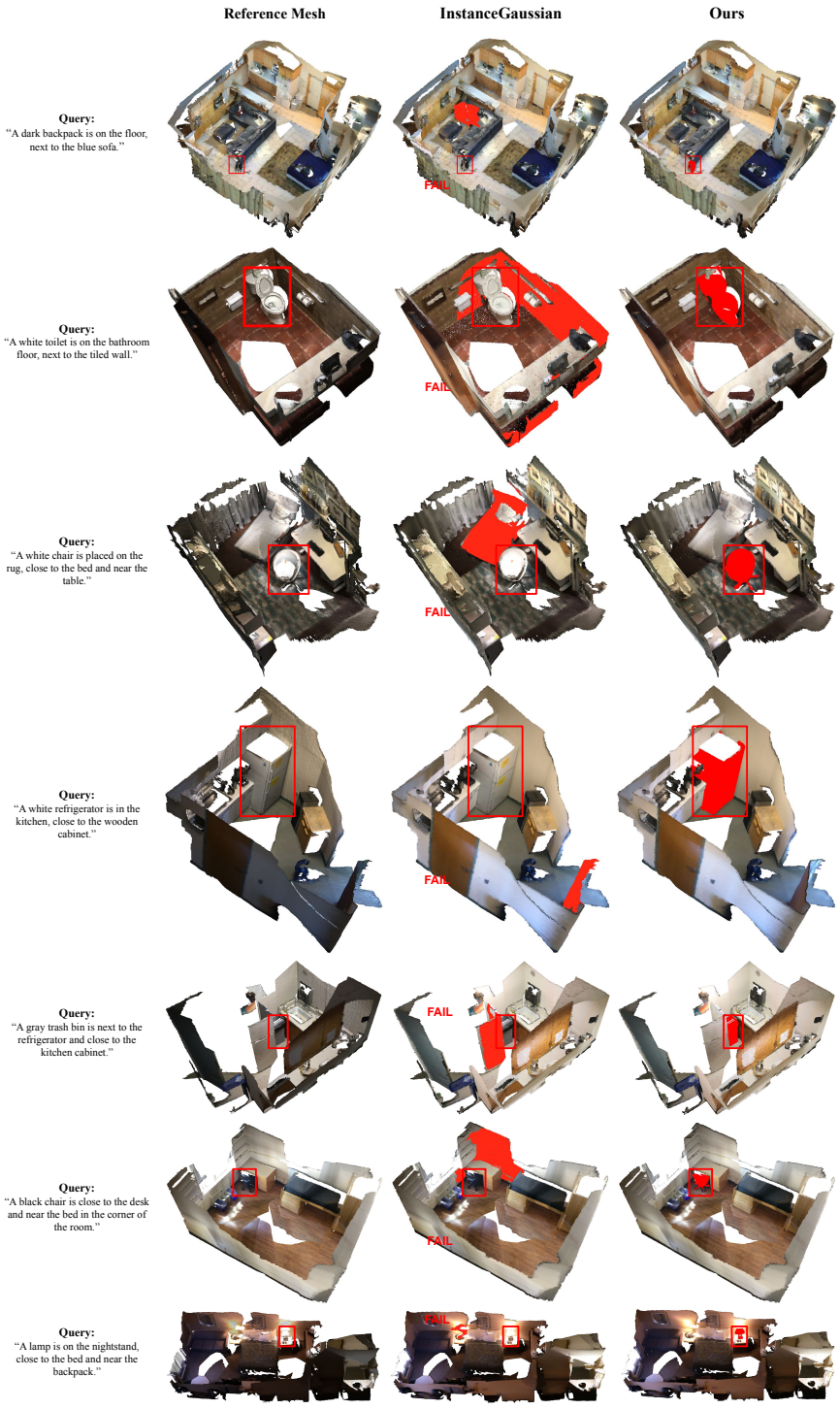



Figure 8: Qualitative results of sentence-level 3D referring segmentation on ScanNet [10]. Predicted target regions are highlighted by red bounding boxes. FAIL indicates unsuccessful localization.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions of the paper: constructing a reusable relation-aware 3DGS representation, learning cross-view consistent semantic nodes, enriching nodes with structured attributes, building a typed and confidence-weighted relation graph, and resolving compositional queries through attribute matching, CLIP retrieval, and graph traversal. These claims are consistent with the method section and are supported by the reported experiments on word-level selection, sentence-level referring segmentation, multi-hop relational queries, ScanNet 3D semantic segmentation, and click-based interaction. The paper does not appear to overclaim beyond its evaluated scope, since the stated contributions are framed around open-vocabulary 3D scene understanding with 3DGS and are validated on the benchmarks discussed in the experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated limitations discussion. It acknowledges that RelGS depends on the quality of the underlying 3DGS reconstruction, camera poses, and multi-view coverage, and that errors from SAM, CLIP, VLMs, and LLM-based relation verification may affect node construction and relation graph reliability. It also discusses the scope of the evaluated settings, noting that the method is mainly tested on LERF-OVS and ScanNet and that generalization to larger outdoor scenes, dynamic scenes, and highly cluttered environments remains future work. It further states the sensitivity to the attribute model, the restrictions of the fixed six-predicate set, and the up-axis assumption for vertical relations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the relevant assumptions, definitions, and formula-based derivations in Sec. 3. The paper does not introduce standalone theorems or lemmas requiring additional formal proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the proposed method in Sec. 3 and provide the datasets, metrics, baselines, and experimental settings in Sec. 4. Additional implementation details are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The experimental settings and implementation details are described in Sections 3–4 and the appendix.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The paper provides the main experimental settings in Sec. 4, including datasets, evaluation protocols, metrics, baselines, and task settings. The training and testing details, including data splits, optimizer settings, model choices, and implementation details, are further provided in the appendix.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experimental section does not include error-bar-related experiments or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include the compute-resource details in the supplementary material, including the GPU type, memory usage, and execution time required to reproduce the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: We conduct experiments on publicly available datasets.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: Our work focuses on 3D scene understanding and is not directly related to social or sensitive decision-making applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We does not pose such risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit the original creators of all existing assets used in this work, including public datasets, pretrained models, and open-source code. The corresponding citations, licenses, and terms of use are respected and followed according to their official releases.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We don't have new asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not involve crowdsourcing, human subjects, or human participant studies in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not involve crowdsourcing, human subjects, or human participant studies in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: This work only involves the use of off-the-shelf LLMs for query parsing and relation verification.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.