

CAPruner: Conceptual-Adjacent Scene Graph Pruner for Enhancing 3D Spatial Reasoning of Large Language Models

Shengli Zhou¹, Xiangchen Wang¹, Guanhua Chen^{1*}, Feng Zheng^{1,2*}

¹Southern University of Science and Technology ²SpatialTemporal AI

{zhousl2022, wangxc2019}@mail.sustech.edu.cn, chengh3@sustech.edu.cn, f.zheng@ieee.org

Abstract

Large language models (LLMs) have recently been applied to 3D vision-language (3D-VL) tasks, which require spatial reasoning to identify target objects relative to anchors. Scene graphs are commonly employed to represent such relations, but reasoning over complete graphs incurs high token costs and computational inefficiencies, motivating the need for pruning. Existing pruning methods primarily rely on spatial proximity and often remove task-relevant relations, thereby undermining reliable spatial reasoning. To address these limitations, we derive a key requirement for scene graph pruning: preserving spatial relations that are most pertinent to the specific 3D-VL task. Guided by this insight, we propose the **Conceptual-Adjacent Scene Graph Pruner (CAPruner)**. CAPruner integrates fuzzy semantic relevance with spatial proximity to estimate the importance of relations, enabling the selection of critical relations in a task-specific context. Moreover, to avoid costly relation-level annotations, CAPruner is trained by supervising the aggregated scores of each node’s incident edges. Extensive experiments demonstrate that CAPruner effectively preserves relations essential for spatial reasoning, leading to substantial performance improvements of LLMs on 3D-VL tasks. Code is available at <https://github.com/fz-zsl/CAPruner>.

1 Introduction

With the development of large language models (LLMs) and their improved reasoning ability, using pretrained LLMs to assist spatial reasoning has become a new paradigm for solving 3D Vision-Language (3D-VL) tasks (Hong et al., 2023; Huang et al., 2024b,a; Zemskova and Yudin, 2024). These tasks require identifying a target object based on its relative position to anchor objects, making accurate perception of spatial relationships crucial

*Co-corresponding authors.

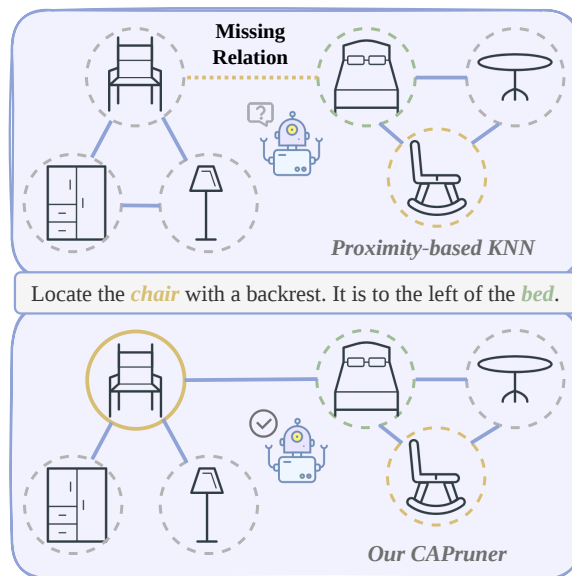


Figure 1: Comparison of scene-graph pruning strategies for LLM spatial reasoning. Previous proximity-based KNN keeps only nearest-neighbor relations and may discard task-critical long-range relations. CAPruner instead combines query-aware semantic cues with spatial proximity to preserve relations that are more useful for downstream reasoning.

for effective spatial reasoning. To better represent relative positions among objects, prior work introduced scene graphs (i.e., an abstract graph whose nodes are objects and whose edges represent relative position relations between objects) for LLMs to perceive the scene. However, as n objects produce $\binom{n}{2}$ pairwise relations, sending all relation descriptions directly to an LLM causes a huge token count. This significantly reduces reasoning efficiency and hinders the extraction of useful information, making scaling impractical. For example, in the InteriorGS dataset (SpatialVerse Research Team, 2025), the average number of objects per scene exceeds 554, which would make the input token count reach the million level and exceed the input length limits of many LLMs. In

real-world scenarios with even more objects, encoding all pairwise relations becomes even less feasible. To address this, 3DGraphLLM (Zemskova and Yudin, 2024) uses a proximity-based K-Nearest-Neighbors (KNN) pruning strategy and encodes only the relations between each object and its two nearest neighboring objects.

However, as shown in Fig. 1, proximity is not strongly correlated with the necessity of a specific binary relation for solving a given task. Combined with the sparsity that KNN pruning produces in the scene graph, this approach cannot guarantee that the relations required to solve a problem are preserved after pruning. When a necessary relation (e.g., the relation between the bed and the chair in the figure) is pruned, the LLM cannot use the anchor object to find the target. Moreover, such an approach does not guarantee the connectivity of the remaining scene graph. Thus, proximity-based KNN cannot fully capture the layout of the entire scene, as the relations between different connected components are missing. As LLMs rely heavily on the retained relations, both shortcomings lead to errors in spatial reasoning.

These observations lead to an important question: **Under a limited budget, which relations in the scene graph should be kept?** In 3D-VL tasks, textual descriptions of relations between objects may include references to anchor objects using their category names and a spatial relation, e.g., “locate the chair (target) next to (relation) the table (anchor)”. Hence, we claim that the importance of a relation can be measured by the attributes of both the incident objects and their positional correlation.

Based on this, we propose a lightweight **Conceptual-Adjacent Scene Graph Pruner (CAPruner)** to select object relations that are potentially useful for solving specific 3D-VL tasks. To reduce the risk of mistakenly pruning relations needed to answer the query, fuzzy matching is applied. For each object, we assign its weight by computing the semantic similarity between its name and words in the natural-language description of a specific task. For example, for the phrase “red chair”, we give higher weight to all chairs in the scene without checking each chair’s actual color. This reduces pruning mistakes that could deteriorate downstream LLM reasoning. For spatial relations between objects, the type of relation (e.g., left, right) can depend on the viewpoint and is hard to accurately determine by a compact pruning model. Hence, we weight edges by object distance follow-

ing the Maxim of Relation (Grice, 1975). Combining these two factors, CAPruner computes the importance of each edge in the scene graph. The network takes the semantic similarity score of the two endpoint objects and their distance as input and outputs an edge weight for pruning.

During training, as current 3D-VL datasets only annotate the target object, and annotating all pairwise relations is prohibitively expensive, we supervise edge-weight learning using only the available target object labels. Concretely, we aggregate the weights of edges around each node and use the aggregated node score for supervision. This encourages edges near the target object to receive higher weights. **At pruning time**, we compute edge weights in the same way for each scene. For every node in the scene graph, we keep incident edges with the highest weights. Since the pruned scene graph only needs to preserve relations required by solving a specific task (i.e., not to fully describe the entire scene), our method avoids the trade-off between preserving global graph connectivity and retaining query-relevant relations.

To sum up, our main contributions are: (1) We conduct qualitative experiments that reveal LLMs perform poorly on spatial relations that are not explicitly mentioned. From this, we summarize the scene-graph requirements when using LLMs for spatial reasoning. (2) We propose CAPruner, a lightweight scene-graph pruning model that uses fuzzy matching on semantics and spatial proximity, plus aggregated node supervision for edge weights. It preserves limited pairwise relations while keeping relations needed to answer a given 3D-VL task. (3) We validate the pruning rationale through extensive experiments. Both quantitative and qualitative results show that our method helps downstream LLMs perform spatial reasoning more accurately.

2 Related Work

2.1 3D LLMs for 3D-VL Tasks

In spatial reasoning tasks, models must accurately perceive the spatial relationships between objects to generate correct answers. Given the limited availability of 3D scene-text paired data, prior research has leveraged the perception and reasoning capabilities of large language models (LLMs) to enhance spatial reasoning. One such approach, 3D-LLM (Hong et al., 2023), encodes the entire 3D scene as a holistic feature. While this method preserves the general layout of the scene, it sacrifices fine-

grained details and mixes features of all objects, making it difficult for the model to identify individual objects and hindering object-level spatial reasoning. To address this, LEO (Huang et al., 2024b) and Chat-Scene (Huang et al., 2024a) segment the scene into distinct objects, encoding the features of each object as input tokens. Although promising, these methods struggle to effectively extract spatial relations between objects, as the absolute positions are prematurely fused with geometric features.

To overcome these limitations, 3DGraphLLM (Zemskova and Yudin, 2024) introduces additional input tokens to explicitly represent spatial relations between objects. However, as the number of relations grows quadratically with object count (often exceeding the input limits of LLMs), 3DGraphLLM adopts a proximity-based KNN strategy, encoding only the relations between each object and its two nearest neighbors. Despite this, the approach remains prone to errors, as it neglects task-specific context and proximity does not always align with task-relevant importance. In contrast, CAPruner considers both semantic relevance and spatial layout when pruning scene graphs, providing LLM with key relations for solving specific 3D-VL tasks.

2.2 Scene Graphs

Scene graphs are structured representations that capture objects and the semantic relations between them. Originally developed in the 2D vision-language domain, scene graphs have been effective for tasks such as image retrieval, referring expression comprehension, and captioning. Extending these concepts to 3D provides a promising way to incorporate structured, relational knowledge in 3D vision-language reasoning.

Scene graphs have also been widely adopted in the 3D domain to address robotics-oriented challenges, including motion planning (Honerkamp et al., 2024; Werby et al., 2024), object localization for navigation (Gu et al., 2024; Honerkamp et al., 2024; Linok et al., 2024; Werby et al., 2024), and robotic manipulation (Honerkamp et al., 2024), as well as 3D scene construction (Gao et al., 2024; Zhai et al., 2024). Aligned with the fast-growing trend of integrating scene graphs for enhancing spatial reasoning in various 3D-VL tasks, several previous methods have leveraged scene graphs to represent spatial relations between objects in the scene. OVSG (Chang et al., 2023) frames the 3D visual grounding problem as a subgraph retrieval

task; 3DGraphQA (Wu et al., 2024) facilitates 3D visual question-answering by introducing a bilinear graph neural network to realize feature fusion between scene graphs and question graphs; FFL-3DOG (Feng et al., 2021) constructs scene graphs based on a textual and visual information and aligns them to obtain the target. Despite these methods having achieved promising results in their respective 3D-VL tasks, they fail to design task context-specific scene graphs tailored to the demands of LLMs (as discussed in Sec. 3), and thus are prone to omitting critical spatial relations that are essential for robust LLM-driven spatial reasoning.

In contrast, CAPruner provides task context-based scene graph pruning. It retains critical spatial relations tailored to LLM reasoning requirements while discarding redundant structural information, thereby enhancing the efficiency and accuracy of LLM-driven spatial reasoning in 3D-VL tasks.

3 Findings

In this section, we explore the role of scene graphs in spatial reasoning tasks for LLMs. Specifically, we have addressed the following two core questions: (1) Why is the construction of scene graphs critical for subsequent reasoning tasks? (2) Why do existing scene graph pruning methods exhibit significant issues?

3.1 Importance of Scene Graph Pruning

To demonstrate the importance of scene graph pruning for LLMs in spatial reasoning tasks, we compare the effects of different edge-selection strategies on downstream task performance. Specifically, we fine-tune LLMs with well-pruned scene graphs and perturbed scene graphs (where some critical spatial relations are removed manually). Then, we test the models on the validation split of the ScanRefer dataset (Chen et al., 2020), which has demanding spatial reasoning requirements, and compare the accuracy of the LLM’s responses.

As shown in fig. 2 (a), model performance declines after perturbation, indicating that the absence of key relations leads to insufficient perception of the scene, resulting in errors in spatial reasoning. This underscores the model’s reliance on the relative positioning of objects encoded in the scene graph. Therefore, to better support spatial reasoning, the pruning model must ensure that spatial relations essential for addressing specific tasks are preserved in the pruned edges.

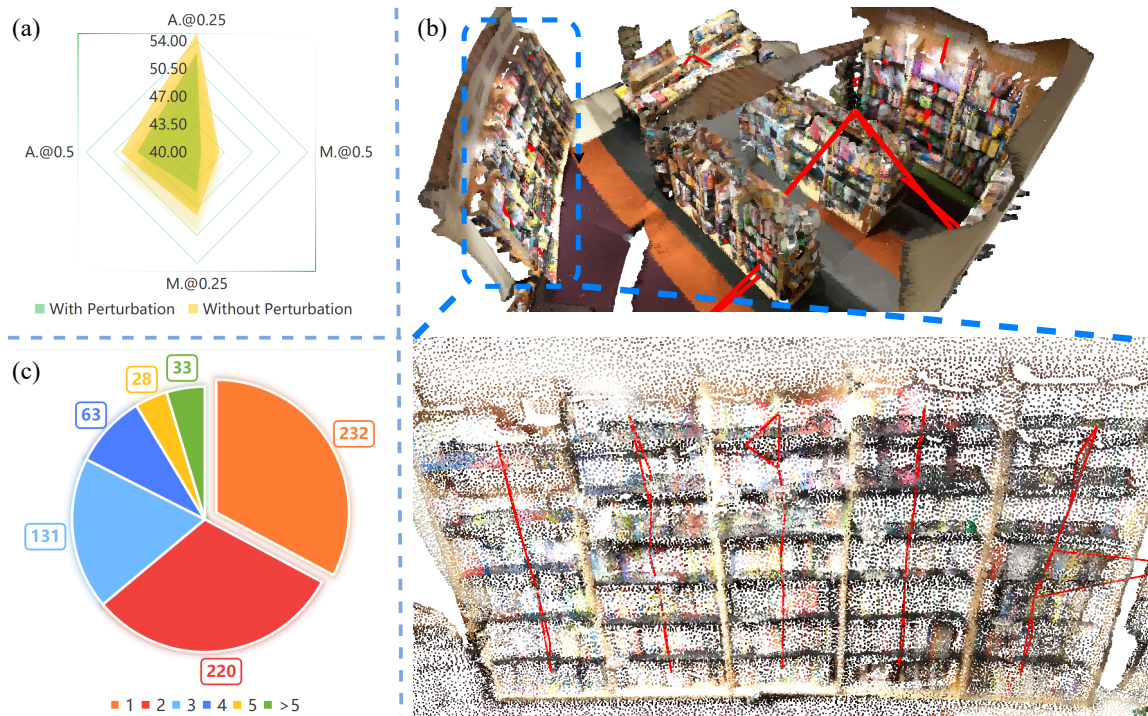


Figure 2: Our findings. (a) Replacing task-critical relations in the scene graph with irrelevant ones consistently degrades downstream accuracy, showing that LLM spatial reasoning relies heavily on the retained relations. (b) Proximity-based KNN overlooks the importance of the context of individual tasks, causing redundancy in local regions (lower part) and global insufficiency (upper part). (c) Over 67% of ScanNet scenes become disconnected after proximity-based KNN pruning, weakening the graph’s ability to represent the global scene layout.

Finding 1: Removing key spatial relationships significantly impairs spatial reasoning.

3.2 Shortcomings of Current Pruners

Existing methods, such as 3DGraphLLM (Zemskova and Yudin, 2024), employ a proximity-based KNN pruning strategy, which preserves the spatial relations between each object and its two nearest neighbors in the scene. Such an approach defines the importance of relations solely based on the distance between objects. However, this approach has two key limitations:

Semantic and Structural Gaps. As the importance of relations is determined entirely by the distance between objects, they neglect the importance of semantic information (e.g., object categories) in the context of specific 3D-VL tasks. For example, in Fig. 2 (b), the proximity between bookshelves and books leads the proximity-based KNN pruning method to establish numerous connections between them, while overlooking relations with other objects. As a result, the LLM provides incorrect answers when tasked with locating “the bookshelf with 7 shelves and 4 sections, located on the wall opposite the table that has books on

it,” as there is no relation between the bookshelf and the wall. Furthermore, the proximity-based KNN pruning strategy does not ensure the connectivity of the pruned scene graph. As illustrated in Fig. 2 (c), among the 707 scenes from the ScanNet data (Dai et al., 2017), only 232 (less than 33%) retain connectivity after pruning. For other scenes, the scene graph fails to represent the relative positions of objects between connected components and, thus, is unable to represent the layout of the entire scene. This severely limits spatial reasoning and the model’s ability to comprehend the scene.

Finding 2: Scene graph pruning should integrate semantic information and maintain structural integrity for the region of interest.

Rigid Pruning Policies. Previous methods (Wang et al., 2023; Zemskova and Yudin, 2024) rely on fixed pruning strategies to select relations for representing the layout of objects within a scene. Such an approach limits their flexibility and ability to adapt to the specific context of a query. This rigidity hinders the model’s ability to prioritize the most context-relevant information, forcing it to encode unnecessary relations. For example, when

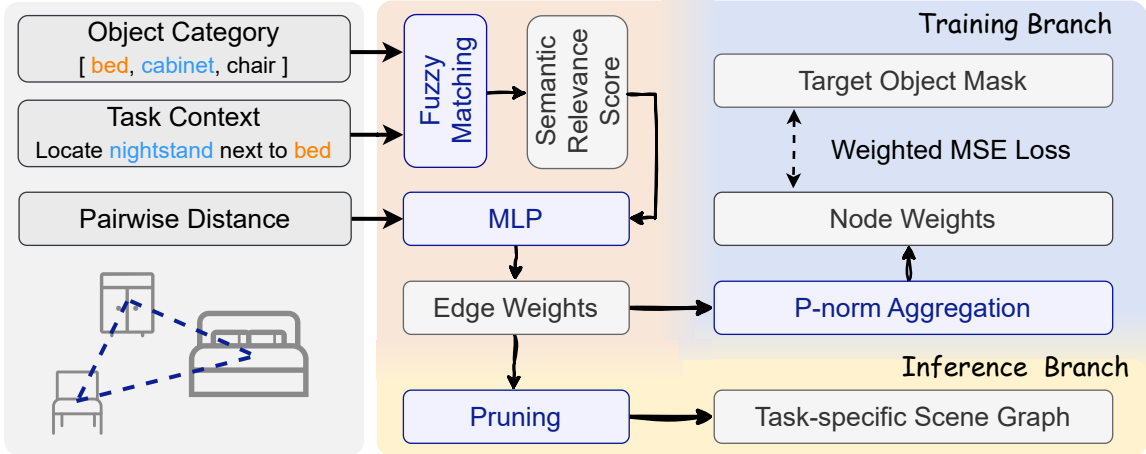


Figure 3: Overview of **CAPruner**. The framework first estimates object-query semantic relevance via fuzzy matching and combines it with geometric cues to predict edge weights. The training branch aggregates edge weights into node weights via p-norm aggregation for node-wise supervision through weighted MSE loss, while the inference branch prunes edges based on edge weights to generate a task-specific scene graph.

tasked with a 3D visual grounding query identifying “the table to the north of the two bookshelves in the center of the room, which is a long, creamy brown rectangle” in the scene shown in Fig. 2 (b), the fixed pruning strategy fails to emphasize the relations crucial to the query (e.g., table-bookshelf). Instead, it retains redundant connections, such as those between sections of bookshelves, wasting budget on irrelevant relations. Consequently, the model inefficiently uses computational resources by processing irrelevant information that does not contribute to the specific task at hand.

Finding 3: Pruning strategy needs to be adaptive to task-specific context for prioritizing relevant information.

Based on the analysis above, we can propose a basic paradigm for scene graph pruning algorithms: (1) To better extract scene structural information for LLM spatial reasoning, it is essential to retain the spatial relations between objects that are crucial for answering specific 3D-VL tasks in the pruned scene graph; (2) During scene graph construction, the pruning model should integrate the context of specific 3D-VL tasks, the semantic information of objects, and the spatial relations between objects to assess the importance of each edge.

4 Method

4.1 Fuzzy Matching

According to the above-mentioned paradigm, we propose **Conceptual-Adjacent Scene Graph**

Pruner (CAPruner). CAPruner is designed to prioritize relations that are crucial in the context of specific 3D-VL tasks according to the semantic properties of objects and their spatial relations in the scene. While LLMs are highly sensitive to missing key relations, they are relatively robust to small amounts of redundant information. Thus, the goal of CAPruner is to minimize the risk of erroneous pruning under a limited budget of retained edges.

For individual objects, textual references typically include the object’s name (e.g., “table”, “cup”) and its properties (e.g., “rectangular”, “red”). Due to current limitations in aligning 3D point cloud features with textual data (Li et al., 2025a; Hadgi et al., 2025), filtering referents according to the object’s detailed properties can easily lead to incorrect pruning decisions. For example, when attempting to match a “red, round table next to three chairs,” any error in determining the object’s color, shape, or relative positioning can result in the actual referent being mistakenly deemed unimportant, triggering false rejection in pruning. To address this, we match objects semantically based on their categories. For instance, when the query mentions a “table,” the scene’s table and semantically similar objects (e.g., “desk”) receive extra weight according to their semantic similarity.

Regarding the spatial relationships between objects, many relational expressions are perspective-dependent (e.g., “front”, “back”, “left”, and “right”). Such a property has made models struggle with understanding these relational categories (Yu et al., 2025). To prevent incorrect pruning in such

cases, we assign higher weights to objects that are closer in proximity, in accordance with the Maxim of Relation theory (Grice, 1975).

4.2 Model Architecture

In this section, we introduce the architecture of the CAPruner model, as shown in Fig. 3. The backbone of CAPruner first calculates semantic relevance scores, and then obtains the weight of each edge as a function of the distance and semantic relevance scores of the objects at its ends. The training branch aggregates edge weights to node weights for node-wise supervision, while the inference branch prunes according to edge weights.

Semantic Relevance Calculation. The semantic relevance of an object is determined by the degree to which its category aligns with the context of the 3D-VL task description. Let \mathcal{T} represent the set of tokens in the natural language description of the 3D-VL task, and c_i denote the category of object i . The semantic relevance score s_i for object i is calculated as the maximum similarity between the object’s category and the task tokens, i.e.,

$$s_i = \max_{t \in \mathcal{T}} \{\text{Similarity}(c_i, t)\} \quad (1)$$

Edge Weight Calculation. The weight of an edge between two objects i and j is determined by their semantic relevance scores and spatial proximity within the scene. Let P_i and P_j denote the positions of objects i and j in the 3D scene. The edge weight w_{ij} is defined as a function of the semantic relevance scores s_i, s_j and the Euclidean distance $\|P_i - P_j\|_2$ between the objects. Formally, $w_{ij} = f(s_i, s_j, \|P_i - P_j\|_2)$, where $f(\cdot)$ is a multi-layer perceptron that aggregates these features to compute the edge weight.

Node-wise Supervision. Currently, many mainstream 3D-VL datasets (Chen et al., 2020; Azuma et al., 2022; Ma et al., 2023) provide annotations for the target object with respect to the textual description. However, none of them provides the importance of edges between each pair of objects. Moreover, as the labeling effort of such annotations scales quadratically with the number of objects, it is impractical to manually label task-specific inter-object relations. Hence, CAPruner employs a node-wise supervision that aggregates edge weights to node weights, supervises node weights, and propagates the effect back to edge weights. Specifically, the weight of each node is calculated using a graph neural network (GNN) approach that aggregates

the weights of the edges incident to it. This aggregation is performed using the p-norm method $v_i = \text{sigmoid} \left(\sum_j w_{ij}^p \right)^{1/p}$, where v_i is the node weight of the i -th object, and p is a hyperparameter that controls the p-norm aggregation.

Weighted MSE loss. Since the number of target objects in the 3D-VL task is typically much smaller than the number of non-target objects, we balance the contributions of target and non-target objects using a weighted mean squared error (WMSE) loss. For the target object set \mathcal{O} and the non-target object set $\tilde{\mathcal{O}}$, the loss function \mathcal{L} is defined as:

$$\mathcal{L} = \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} (v_i - 1)^2 + \frac{1}{|\tilde{\mathcal{O}}|} \sum_{i \in \tilde{\mathcal{O}}} v_i^2 \quad (2)$$

As the sigmoid function limits v_i in the range of $(0, 1)$, the first term encourages nodes corresponding to target objects to have higher weights, and the second term encourages nodes corresponding to non-target objects to have lower weights.

These weights are then propagated back through the p-norm aggregation operation on the scene graph, encouraging edges incident to target objects to have higher edge weights, while lowering the edge weights of edges incident to non-target objects. Such a supervision approach helps the model to strike a better balance between semantic relevance and proximity.

By the end of training, the scene graph can be flexibly pruned according to edge weights with respect to the task context, preserving the most relevant relationships for the task at hand while discarding redundant or irrelevant connections.

5 Experiments

5.1 Experimental Settings

Implementation Details. For semantic relevance calculation, we classify objects into general categories defined in the NYUv2 dataset (Nathan Silberman and Fergus, 2012). The object receives a semantic similarity score of 1 only when there exists an object of the same category in the textual description. When computing edge weights using the semantic relevance score and objects’ pairwise distances, we utilize a 3-layer MLP with 1219 parameters, which yields high computational efficiency. To enhance the robustness of our model while making a fair comparison with the previous proximity-based KNN model, 3DGraphLLM

Model	ScanRefer				ScanQA	SQA3D
	A.@0.25	A.@0.5	M.@0.25	M.@0.5	BLEU-4	EM@1
FFL-3DOG (Feng et al., 2021)	41.3	34.0	35.2	25.7	–	–
SeeGround (Li et al., 2025b)	44.1	39.4	34.0	30.0	–	–
CSVG (Yuan et al., 2025)	49.6	39.8	38.4	27.3	–	–
AugRefer (Wang et al., 2025)	55.7	44.0	50.0	39.1	–	–
MA2TransVG (Xu et al., 2024)	57.9	45.7	53.8	41.4	–	–
3D-VisTA (Zhu et al., 2023)	50.6	45.8	43.7	39.1	13.1	48.5
TSP3D (Guo et al., 2025)	56.5	46.7	–	–	–	–
Scene-LLM (Fu et al., 2025)	–	–	–	–	12.0	54.2
Chat-Scene-7B (Huang et al., 2024a)	55.5	50.2	–	–	14.3	54.6
PQ3D (Zhu et al., 2025)	–	51.2	–	46.2	–	47.1
QuatRoPE (Zhou et al., 2026)	58.2	52.5	54.3	49.2	–	55.2
3DGraphLLM-1B (Zemskova and Yudin, 2024)	52.5	47.5	45.0	40.5	12.2	52.6
CAPruner + Llama-3.2-1B (Ours)	55.0	49.6	48.0	42.8	13.0	52.8
3DGraphLLM-8B (Zemskova and Yudin, 2024)	60.2	54.6	54.7	49.4	12.5	55.2
CAPruner + Llama-3-8B (Ours)	61.7	56.0	55.3	49.9	13.2	56.3

Table 1: Comparison on ScanRefer, ScanQA, and SQA3D. With the same backbone LLM, CAPruner consistently outperforms base methods and achieves the strongest or highly competitive results. A. and M. denote accuracy on the overall and “multi” splits, respectively. Scores for 3DGraphLLM-1B and in italic are evaluated on our machine.

(Zemskova and Yudin, 2024), we preserve two incident edges with the highest weights for each node in the scene graph. In the experiments, Llama-3.2-1B is used for 1B models, and Llama-3-8B is used for 8B models (Grattafiori et al., 2024). When parsing the pruned scene graph into LLM for inference, we arrange a series of tokens in a sequence. The sequence pattern follows the same setting in 3DGraphLLM, encoding the features of texts, individual objects, and selected relations.

Training Approach. During training, we first train the CAPruner model for pruning the scene graph. The model is trained for 50 epochs with a batch size of 16, a learning rate of 10^{-3} , and $p = 3$ for p-norm aggregation. Then, we fine-tune the LLM using LoRA with $r = 16$ for 3 epochs with a batch size of 8 and a learning rate of 2×10^{-5} .

5.2 Comparative Experiment

In this experiment, we aim to verify the effectiveness of CAPruner by comparing its performance against previous models. The experiments are conducted on the ScanRefer (Chen et al., 2020) dataset for 3D visual grounding (3D VG), the ScanQA (Azuma et al., 2022) dataset for 3D visual question-answering (3D VQA), and the SQA3D (Ma et al., 2023) dataset for situated 3D VQA. We report localization accuracy on ScanRefer, BLEU-4 on ScanQA, and EM@1 on SQA3D.

The results in Tab. 1 demonstrate that models using scene graphs pruned by CAPruner have

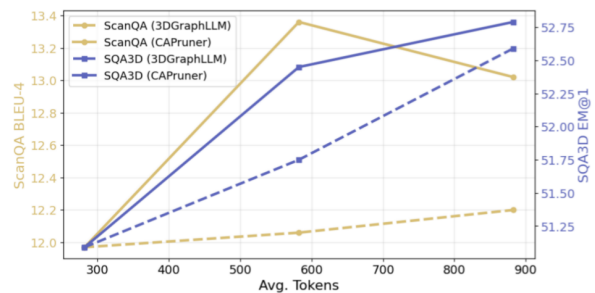


Figure 4: Comparison between CAPruner (1B) and 3DGraphLLM (proximity-based KNN, 1B)

achieved large gains throughout all metrics compared to models using proximity-based KNN pruning (i.e., 3DGraphLLM) when using the same LLM, especially on datasets with higher spatial reasoning demands like ScanRefer. Our model has also achieved the highest scores for most metrics, showing the superiority of our method in spatial reasoning and comprehensive 3D-VL task-solving ability. Additionally, CAPruner’s inference time for each 3D-VL sample is 0.75 ms, which is negligible compared to the inference time of LLM (e.g., 1731 ms per sample for the 8B model).

Meanwhile, as shown in Fig. 4, CAPruner not only outperforms proximity-based KNN across all token budgets but also achieves higher token efficiency for 3D-VL tasks. For example, CAPruner, with an average of 582 tokens, performs better than the baseline model, which uses an average of 882 tokens, saving 34% in token usage.

Pruning Method	ScanRefer				ScanQA		SQA3D		
	A.@0.25	A.@0.5	M.@0.25	M.@0.5	B.-3	B.-4	EM@1	ROUGE	CIDEr
Proximity-based KNN	52.5	47.5	45.0	40.5	17.9	12.2	52.6	53.8	138.3
CAPruner (MST)	54.4	49.0	47.1	42.0	18.1	11.7	52.4	53.7	139.0
Gain	1.9	1.5	2.1	1.5	0.2	-0.5	-0.2	-0.1	0.7
CAPruner (KNN)	55.0	49.6	48.0	42.8	18.5	13.0	52.8	54.1	139.1
Gain	2.5	2.1	3.0	2.3	0.6	0.8	0.2	0.4	0.8

Table 2: Comparison of pruning policies after learning CAPruner edge weights. Applying KNN to the learned scores performs best. A. and M. denote accuracy on the overall and “multi” splits, respectively; B.-3 and B.-4 denote BLEU-3 and BLEU-4. All models are based on Llama-3.2-1B.

Proximity	Semantic Similarity	Accuracy
✗	None	7.73
✗	Bert (Devlin et al., 2019)	8.21
✗	Strict Matching	17.71
✗	Fuzzy Matching	24.44
✓	None	7.73
✓	Bert (Devlin et al., 2019)	8.57
✓	Strict Matching	24.38
✓	Fuzzy Matching	24.57

Table 3: Ablation on proximity cues and semantic matching strategies. Strict matching helps when object categories are explicitly mentioned, but fuzzy matching is consistently stronger; combining fuzzy matching with proximity yields the best accuracy.

5.3 Comparison on Pruning Approaches

When using the scene graph to represent the layout of the entire scene, the connectivity after pruning is crucial to maintaining structural integrity, but it also hinders the model from focusing on more important relations. As CAPruner is designed to represent key relations in a specific 3D-VL task context, it only needs to focus on the region of interest, lowering the requirement on structural integrity.

In this experiment, after training the CAPruner model, we use KNN and MST (first select the edges on the maximum-weight spanning tree of the scene graph, then each node selects one more remaining incident edge with the largest weight) strategies for pruning. Finally, we fine-tune the downstream LLM and compare the performances.

The results in Tab. 2 demonstrate that the KNN strategy performs better, verifying our advantage of forgoing the requirement of structural integrity by characterizing local regional features.

5.4 Ablation Studies

We perform ablation studies on model settings and measure the accuracy of CAPruner as the per-

p	1	2	3	4	5	6
Acc.	7.57	19.22	24.57	24.51	24.57	24.36

Table 4: Ablation on p used in p -norm aggregation.

centage of samples where it can correctly predict $|\mathcal{O}|$ (i.e., number of targets) objects with top node weights without involving LLM. Results for the effects of proximity, the semantic similarity computation method, and the choice of p in p -norm aggregation are presented in Tab. 3 and Tab. 4.

The results lead to three conclusions: (1) Fuzzy matching outperforms other strategies, including Bert embeddings, whose discriminability is relatively low. (2) Proximity helps identify important edges and target objects, but its effect is much smaller than semantics, which also aligns with the Maxim of Relation that “in reference, semantics are given priority, while proximal objects are considered when conditions are equal”. (3) Accuracy grows as p grows for $p \leq 3$ and remains stable afterward, indicating that edges with large weights play a more important role than the sum of the edge weights in predicting target objects.

6 Conclusion

We presented CAPruner, a lightweight scene graph pruning model for LLM-based spatial reasoning in 3D-VL tasks. By combining task-specific semantic cues with spatial proximity and training with aggregated node-level supervision, CAPruner preserves relations that are most useful for downstream reasoning without requiring relation-level annotation. Experiments show that CAPruner consistently outperforms proximity-based pruning with negligible overhead, highlighting task-specific scene graph pruning as an effective and scalable strategy.

Limitations

Though edges in scene graphs can correspond to most expressions in 3D-VL task descriptions, they cannot represent complex relations (e.g., the fourth chair counting from the left in a row of chairs behind the fifth row of desks in the classroom). Therefore, a more versatile and generalizable data structure should be considered to better represent complex relations in real-world scenarios.

Acknowledgements

This work was supported by the National Key Research and Development Program of China under Grant 2024YFE0203100.

References

- Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. 2022. Scanqa: 3d question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Haonan Chang, Kowndinya Boyalakuntla, Shiyang Lu, Siwei Cai, Eric Jing, Shreesh Keskar, Shijie Geng, Adeeb Abbas, Lifeng Zhou, Kostas Bekris, and 1 others. 2023. Context-aware entity grounding with open-vocabulary 3d scene graphs. *arXiv preprint arXiv:2309.15940*.
- Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. 2020. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 202–221. Springer.
- Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mingtao Feng, Zhen Li, Qi Li, Liang Zhang, Xiang-Dong Zhang, Guangming Zhu, Hui Zhang, Yaonan Wang, and Ajmal Mian. 2021. Free-form description guided 3d visual graph network for object grounding in point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3722–3731.
- Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. 2025. Scene-llm: Extending language model for 3d visual reasoning. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 2195–2206.
- Gege Gao, Weiyang Liu, Anpei Chen, Andreas Geiger, and Bernhard Schölkopf. 2024. Graphdreamer: Compositional 3d scene synthesis from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21295–21304.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and 1 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- H. Paul Grice. 1975. [Logic and conversation](#). *Syntax and Semantics*, 3:41–58.
- Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, and 1 others. 2024. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE.
- Wenxuan Guo, Xiuwei Xu, Ziwei Wang, Jianjiang Feng, Jie Zhou, and Jiwen Lu. 2025. Text-guided sparse voxel pruning for efficient 3d visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 3666–3675.
- Souhail Hadgi, Luca Moschella, Andrea Santilli, Diego Gomez, Qixing Huang, Emanuele Rodolà, Simone Melzi, and Maks Ovsjanikov. 2025. Escaping plato’s cave: Towards the alignment of 3d and text latent spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19825–19835.
- Daniel Honerkamp, Martin Buchner, Fabien Despinoy, Tim Welschhold, and Abhinav Valada. 2024. [Language-grounded dynamic scene graphs for interactive object search with mobile manipulation](#). *IEEE Robotics and Automation Letters*, 9:8298–8305.
- Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 2023. 3d-llm: Injecting the 3d world into large language models. *NeurIPS*.
- Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, and 1 others. 2024a. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada*.
- Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. 2024b. An

- embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- Haoyuan Li, Yanpeng Zhou, Yufei Gao, Tao Tang, Jianhua Han, Yujie Yuan, Dave Zhenyu Chen, Jiawang Bian, Hang Xu, and Xiaodan Liang. 2025a. Does your 3d encoder really work? when pretrain-sft from 2d vlms meets 3d vlms. *arXiv preprint arXiv:2506.05318*.
- Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. 2025b. [Zero-shot 3d visual grounding from vision-language models](#). *ArXiv*, abs/2505.22429.
- Sergey Linok, Tatiana Zemskova, Svetlana Ladanova, Roman Titkov, Dmitry A. Yudin, Maxim Monastyrny, and Aleksei Valenkov. 2024. [Beyond bare queries: Open-vocabulary object grounding with 3d scene graph](#). *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13582–13589.
- Xiaojuan Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. 2023. [Sqa3d: Situated question answering in 3d scenes](#). In *International Conference on Learning Representations*.
- Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. 2012. Indoor segmentation and support inference from rgb-d images. In *ECCV*.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, and 1 others. 2024. [DINOv2: Learning robust visual features without supervision](#). *Transactions on Machine Learning Research*. Featured Certification.
- Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. 2023. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. *International Conference on Robotics and Automation (ICRA)*.
- Manycore Tech Inc. SpatialVerse Research Team. 2025. InteriorGS: A 3d gaussian splatting dataset of semantically labeled indoor scenes. <https://huggingface.co/datasets/spatialverse/InteriorGS>.
- Xinyi Wang, Na Zhao, Zhiyuan Han, Dan Guo, and Xun Yang. 2025. [Augrefer: Advancing 3d visual grounding via cross-modal augmentation and spatial relation-based referring](#). *CoRR*, abs/2501.09428.
- Ziqin Wang, Bowen Cheng, Lichen Zhao, Dong Xu, Yang Tang, and Lu Sheng. 2023. VI-sat: Visual-linguistic semantics assisted training for 3d semantic scene graph prediction in point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21560–21569.
- Abdelrhman Werby, Chenguang Huang, Martin Buehner, Abhinav Valada, and Wolfram Burgard. 2024. [Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation](#). In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands.
- Zizhao Wu, Haohan Li, Gongyi Chen, Zhou Yu, Xiaoling Gu, and Yigang Wang. 2024. 3d question answering with scene graph reasoning. In *ACM Multimedia 2024*.
- Can Xu, Yuehui Han, Rui Xu, Le Hui, Jin Xie, and Jian Yang. 2024. Multi attributes interactions matters for 3d visual grounding. In *CVPR*.
- Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. 2025. Inst3d-Imm: Instance-aware 3d scene understanding with multi-modal instruction tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14147–14157.
- Qihao Yuan, Kailai Li, and Jiaming Zhang. 2025. [Solving zero-shot 3d visual grounding as constraint satisfaction problems](#). In *36th British Machine Vision Conference 2025, BMVC 2025, Sheffield, UK, November 24-27, 2025*. BMVA.
- Tatiana Zemskova and Dmitry Yudin. 2024. [3dgraphllm: Combining semantic graphs and large language models for 3d scene understanding](#). *Preprint*, arXiv:2412.18450.
- Guangyao Zhai, Evin Pinar Örnek, Shun-Cheng Wu, Yan Di, Federico Tombari, Nassir Navab, and Benjamin Busam. 2024. Commonsenses: Generating commonsense 3d indoor scenes with scene graphs. *Advances in Neural Information Processing Systems*, 36.
- Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. 2024. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations (ICLR)*.
- Shengli Zhou, Minghang Zheng, Feng Zheng, and Yang Liu. 2026. [Scalable object relation encoding for better 3d spatial reasoning in large language models](#). *Preprint*, arXiv:2603.24721.
- Ziyu Zhu, Xiaojuan Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 2023. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2911–2921.
- Ziyu Zhu, Zhuofan Zhang, Xiaojuan Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. 2025. Unifying 3d vision-language understanding via promptable queries. In *Computer Vision – ECCV 2024*, pages 188–206, Cham. Springer Nature Switzerland.

A Additional Experiments

A.1 Ablation Studies

In this section, we provide additional ablation studies on the choice of loss and similarity function.

The choice of loss function. We choose weighted MSE loss as it is more robust when there is a large difference between the number of target and non-target objects (which is common for 3D-VL tasks). To validate its effectiveness, we also train CAPruner using binary cross-entropy (BCE) loss. The accuracy on the validation set when using the BCE loss is 19.72, which is much lower than the accuracy when trained with the weighted MSE loss (24.57), verifying the superiority of the weighted MSE loss.

The choice of similarity function. In this experiment, we compare the model’s accuracy when inferred using category-based fuzzy matching or the cosine similarity between DINOv2 (Oquab et al., 2024) / Uni3D (Zhou et al., 2024) feature vectors and the Bert-Large-Uncased (Devlin et al., 2019) embedding as the similarity function. The results in Tab. 5 demonstrate that when using the dot product of the embedding vectors (i.e., cross-modal alignment scores) to measure similarity, the accuracy of the model deteriorates, indicating lower quality of the pruned scene graph. Such results further demonstrate the superiority of using category-based fuzzy matching as the similarity function.

Matching Feature	ScanRefer Acc.@0.25	ScanQA EM@1	Multi3DRef F1@0.25
Category	55.04	21.02	55.60
DINOv2	54.05	20.81	55.12
Uni3D	54.53	20.79	55.25

Table 5: Comparison on model’s accuracy when inferred using category-based fuzzy matching or cosine similarity of DINOv2 (Oquab et al., 2024) / Uni3D (Zhou et al., 2024) feature vectors as the similarity function.

A.2 Generalizability Verification

In this section, we perform an experiment to investigate cross-dataset generalization. We train CAPruner on a single dataset and directly evaluate it on all datasets without retraining. In Tab. 6, the row indicates the dataset used for training, the column indicates the dataset used for evaluation, and the scores are the accuracies of the model. From the results, we can observe that CAPruner trained on only one dataset can achieve performance very close to the model trained on all datasets across

all evaluation settings. This clearly demonstrates that CAPruner has strong cross-dataset generalization ability, and the model trained on one dataset can transfer to other datasets effectively without retraining.

Training Set	ScanRefer	Multi3DRef	ScanQA
ScanRefer	21.80	27.53	22.48
Multi3DRef	21.71	27.49	22.21
All	21.85	27.61	22.74

Table 6: The model’s accuracy when trained using a single dataset or the combination of all datasets and tested on the validation sets.

B Qualitative Results

In this section, we visualize scene graphs pruned by CAPruner and compare them with scene graphs obtained through proximity-based KNN pruning. The context is chosen from the text descriptions in the 3D visual grounding (3D VG) dataset ScanRefer (Chen et al., 2020). The red edges in the figure correspond to those in the pruned scene graph. To enhance the clarity of visualization, we adopt a setting where only one incident edge with the highest weight (for CAPruner) / shortest distance (for proximity-based KNN) is retained.

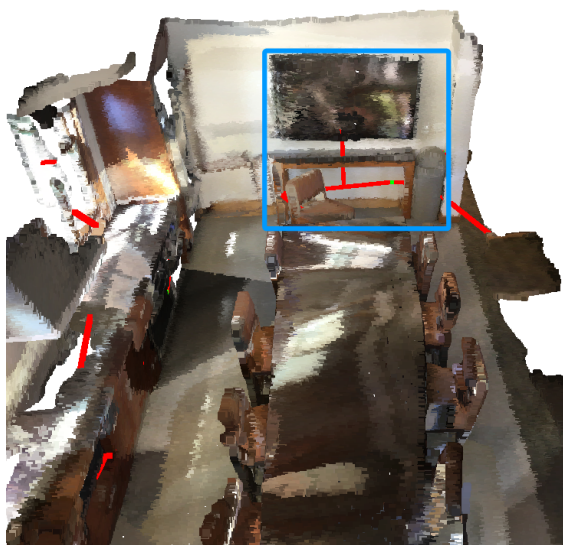
Case (a) When using proximity-based KNN to prune the scene graph, the resulting graph on the left is overall sparse, hindering the model’s ability to perceive the scene layout. In contrast, by introducing more edges with larger lengths, CAPruner on the right can better shape the structure of the scene. Moreover, when handling the task of locating “it is a gray trash can, the trash can sits in the corner by where the TV is”, the KNN-pruned scene graph in the left figure fails to represent the spatial relation between the trash can and the TV, as the trash can are connected with the table and the wall, which are closer to it. By considering the categories of objects and the semantics of the task description, CAPruner gets rid of the limitations of spatial proximity and retains the edge between the trash can and the TV to reach the spatial reasoning requirement of the description.

Case (b) While proximity-based KNN pruning on the left can only focus on local spatial relations, CAPruner on the right can focus on long-range relations according to the requirement of the task. Such a characteristic enables the model to handle spatial relations with longer distances, e.g., “opposite to” and “farthest”.

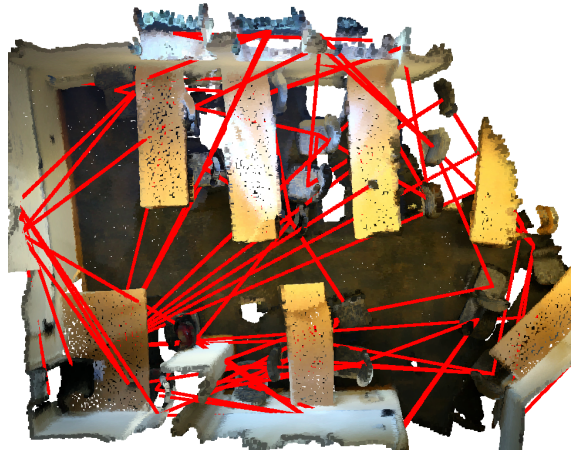
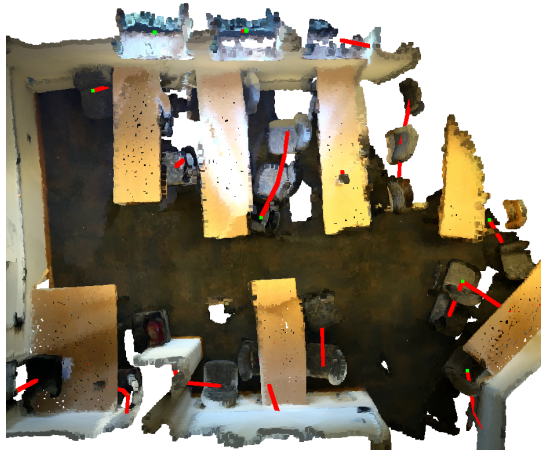
Proximity-based KNN

CAPruner (Ours)

(a) [scene0011_00] 3D VG: It is a gray trash can, the trash can sits in the corner by where the TV is.



(b) [scene0015_00] 3D VG: It is a long brown table located opposite the crossed table on the other side.



(c) [scene0208_00] 3D VG: It is a long brown table located opposite the crossed table on the other side.



Table 7: Qualitative Results

Case (c) As shown in the lower part of Fig. 2 (b) and Case (c), there are lots of edges that have both ends within the same bookshelf. Such a phenomenon is caused by the preliminary step for LLM spatial reasoning, i.e., scene instance segmentation. When the segmentation model, e.g., Mask3D (Schult et al., 2023), segments large objects (e.g., bookshelf) into multiple small objects (e.g., books or sections of the bookshelf). Such a segmentation makes proximity-based KNN prone to preserving edges between different parts of the same object. In contrast, because CAPruner also considers semantic similarity, the pruned scene graph is more robust to segmentation results.

C Details for Datasets

The number of examples in the datasets used for training and validation is as follows:

Dataset	Training	Validation
ScanRefer	32338	9508
ScanQA	26138	4675
Multi3DRef	37695	11120
SQA3D	26623	3261

Table 8: Number of examples in the datasets.