GraphSelect: Task-Aware Scene Graph Compression for Efficient 3D Scene Reasoning with LLMs

Anonymous ACL submission

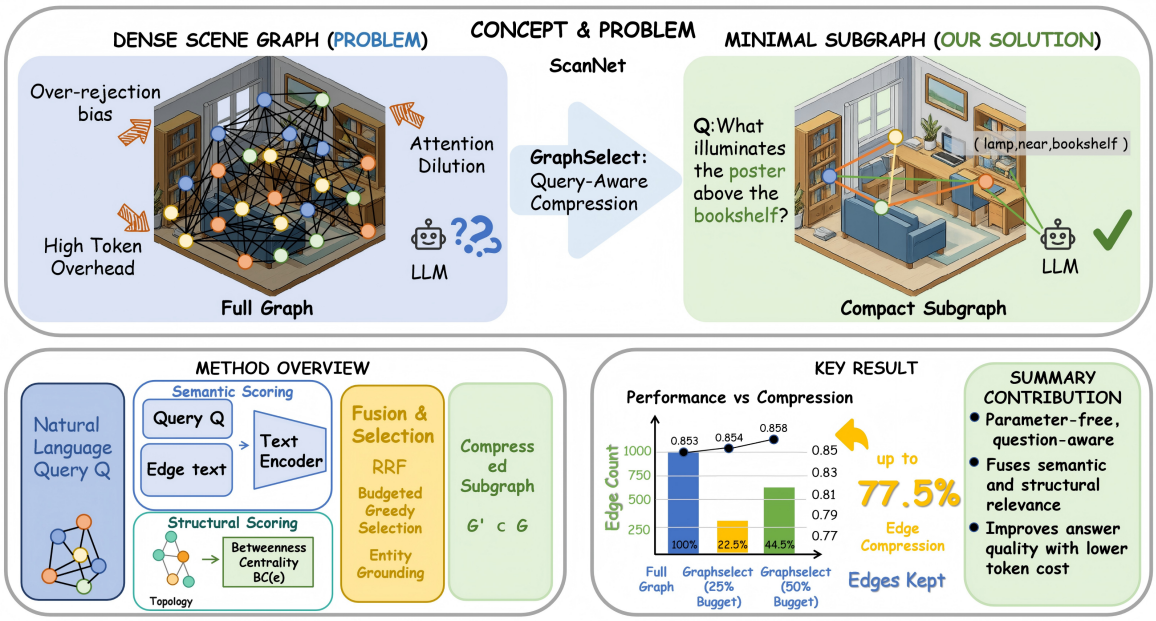

Figure 1: Overview of GraphSelect. The dense scene graph on the left introduces redundant relations, causing token overhead and attention dilution. GraphSelect extracts a compact, query-aware subgraph that preserves critical reasoning paths and removes distractions. This reduces token cost by up to 77.5% and improves answer quality.

Abstract

Large language models excel at 3D scene understanding by discretizing physical environments using 3D scene graphs. However, directly feeding dense full scene graphs leads to massive token overhead, attention dilution, and biases such as the over-rejection of valid spatial instructions. Simple static pruning methods perform poorly as they sever critical semantic multi-hop reasoning paths. To address this issue, we propose GraphSelect, a parameter-free, query-aware scene graph compression algorithm. We evaluate GraphSelect on the ScanQA and SQA3D benchmarks with 3DGraphLLM as the downstream reasoning model. Experimental results demonstrate that our method can reduce prompt cost significantly while further improving answer quality and robustness.

1 Introduction

Large language models (LLMs) have demonstrated exceptional capabilities in 3D scene reasoning tasks, including visual question answering, 3D grounding, and embodied planning (Zemskova and Yudin, 2025). A key step in these tasks is translating the continuous 3D physical environment into a discrete format that LLMs can process. To address this, existing methods commonly use 3D scene graphs as an intermediate representation. By abstracting physical objects as nodes and their spatial relationships as directed edges, scene graphs effectively discretize the 3D space. This structured topological representation provides LLMs with rich semantic context and supports complex relational reasoning across the environment.

To process this representation, current methods typically linearize the nodes and edges of the scene

graph into textual sequences to construct context prompts. While this approach has achieved promising results, directly feeding the complete scene graph into the LLM presents a critical information-density bottleneck. This exhaustive paradigm leads to three primary issues. (1) High token overhead: real-world 3D environments, such as those in ScanNet (Dai et al., 2017), often contain hundreds of objects and thousands of spatial relations. Inputting all these edges creates a long token sequence, which increases computational cost and inference latency. (2) Attention dilution (Kamradt, 2023): when LLMs distribute attention over long input sequences that contain many irrelevant relations, structural noise can distract the model from the essential reasoning path. (3) Over-rejection bias: when the context is overly dense or contains conflicting relations, models may incorrectly reject valid spatial instructions as unanswerable, even when the required evidence is present in the graph.

A naive countermeasure is to prune the graph using static topological metrics, degree centrality. However, this strategy often disrupts multi-hop reasoning chains by discarding low-degree edges that serve as important semantic bridges. For instance, answering "What illuminates the poster above the bookshelf?" may depend on the edge *(lamp, near, bookshelf)*, even if both objects have only a few connections in the full graph. Random edge dropout similarly breaks such critical pathways. Therefore, effective graph compression cannot rely only on static graph properties. It must be dynamically conditioned on the semantic intent of the query.

To address this problem, we propose GraphSelect, a parameter-free algorithm for query-aware scene graph compression. GraphSelect dynamically extracts a compact informative subgraph by computing two complementary relevance signals for each edge. First, it measures semantic alignment between the edge description and the query text using a pretrained encoder. Second, it estimates the structural importance of each edge with betweenness centrality, which helps preserve topological bottlenecks required for multi-hop reasoning. These two signals are fused into a single relevance ranking using reciprocal rank fusion (Cormack et al., 2009), allowing the algorithm to retain the most semantically and structurally salient edges for the compressed subgraph under a token budget. Our experiments show that GraphSelect substantially improves token efficiency while preserving the accuracy of complex 3D reasoning tasks.

This framing makes graph compression more than an efficiency problem. For LLM-based scene reasoning, a shorter input can be more useful when it removes irrelevant or conflicting relations while preserving the evidence needed by the query. GraphSelect is designed around this principle: it compresses the graph not by uniformly shortening the context, but by selecting edges that jointly match the query semantics and maintain the structural paths required for reasoning.

In summary, our main contributions are:

- We propose GraphSelect, a parameter-free graph compression framework that shifts 3D LLM reasoning from full-graph ingestion to dynamic query-aware subgraph extraction.
- We design a dual-stream relevance scoring mechanism that integrates the semantic similarity of textual edge descriptions with the structural topology of the 3D scene.
- We conduct experiments across multiple benchmarks, showing that GraphSelect reduces prompt cost while maintaining or improving downstream answer quality.

2 Related Work

3D Scene Graph Representations. A 3D scene graph represents objects as nodes and their spatial or semantic relations as edges, providing a structured abstraction of indoor environments (Armeni et al., 2019; Wald et al., 2020). Such graphs can be built from 3D reconstructions such as ScanNet (Dai et al., 2017) or generated by open-vocabulary graph builders (Gu et al., 2024; Koch et al., 2024; Hou et al., 2025; Yeo et al., 2025). Recent work further enriches scene representations with object-level descriptions or sparse 3D tokens, such as Descrip3D (Xue et al., 2026) and GaussianVLM (Halacheva et al., 2025). These methods improve the expressiveness of 3D scene representations, but dense graphs can contain thousands of relations, many irrelevant to a given query. GraphSelect is complementary to these representation-building methods: it operates on an existing scene graph and removes query-irrelevant edges while preserving relations needed for downstream reasoning.

Context and Graph Compression. Reducing redundant context has been studied in graph summarization, prompt compression, and task-aware input selection. Classical graph summarization

and connection-subgraph methods preserve global structure (Tian et al., 2008; Navlakha et al., 2008; Tong et al., 2006; Faloutsos et al., 2008), but are not designed for language-conditioned 3D reasoning. Prompt compression methods, such as LLMLingua (Jiang et al., 2023, 2024), Selective Context (Li et al., 2023), and RECOMP (Xu et al., 2023), operate on unstructured text and may break serialized multi-hop relations in a scene graph. SpatialPrompting (Taguchi et al., 2025) shows that task-aware evidence selection benefits spatial reasoning, but selects keyframes rather than graph edges. Query-aware graph reasoning has also been explored in knowledge graphs and GNNs (Sun et al., 2018; Saxena et al., 2020; Hamilton et al., 2017; Veličković et al., 2017), usually with task-specific training. GraphSelect differs from these lines of work by performing parameter-free, query-aware edge selection directly on 3D scene graphs, combining semantic affinity with structural importance (Brandes, 2001; Schütze et al., 2008).

This distinction is important because serialized scene graphs are not independent text spans. Removing one relation can disconnect two otherwise relevant objects and break a multi-hop spatial chain. Thus, selecting graph evidence requires considering both query relevance and topological connectivity, whereas standard prompt compression mainly estimates the importance of individual tokens or text segments. GraphSelect explicitly targets this structured setting by ranking relation edges rather than pruning unstructured text.

3 Method

3.1 Overview

Given a 3D scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a natural language query Q, GraphSelect aims to extract a compact subgraph that preserves the relations needed for answering the query. Each node $v \in \mathcal{V}$ denotes an object, and each directed edge $e = (v_i, r, v_j) \in \mathcal{E}$ denotes a spatial or semantic relation r between two objects. Under a token budget B, our goal is to select an edge subset $\mathcal{E}' \subseteq \mathcal{E}$ and form $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ such that:

$$\mathrm{tokens}(\mathcal{G}') \leq B, \qquad \mathrm{QA}(\mathcal{G}', Q) \approx \mathrm{QA}(\mathcal{G}, Q). \tag{1}$$

Here, $\mathrm{tokens}(\cdot)$ denotes the token length of the serialized graph, and $\mathrm{QA}(\cdot, Q)$ denotes the downstream LLM response for query Q.

3.2 Query-Aware Edge Scoring

GraphSelect estimates the utility of each graph edge using two complementary signals: semantic relevance to the input query and structural importance within the scene graph.

Semantic Relevance. For each edge $e = (v_i, r, v_j)$, we convert it into a short textual description $t(e) =$ "$v_i\ r\ v_j$", such as "lamp on table". We compute its semantic alignment with the query using a frozen sentence encoder f_θ:

$$s_{\mathrm{sem}}(e, Q) = \cos\big(f_\theta(Q), f_\theta(t(e))\big). \tag{2}$$

This score selects edges whose relation descriptions are semantically relevant to the query, even when the query and edge text share no exact words.

Structural Relevance. Semantic relevance alone may miss edges that are weakly aligned with the query text but remain important for connecting relevant object clusters during multi-hop reasoning. To preserve such bridge edges, GraphSelect further computes a structural relevance score:

$$s_{\mathrm{struct}}(e) = \mathrm{BC}(e) \cdot w(r, \mathrm{type}(Q)), \tag{3}$$

where $\mathrm{BC}(e)$ is the edge betweenness centrality, and $w(r, \mathrm{type}(Q))$ is a query-type-aware relation weight. Betweenness centrality gives high scores to edges that lie on many shortest paths, which helps preserve topological bottlenecks for multi-hop reasoning. We use a lightweight priority-based heuristic to classify queries into counting, existence, spatial, functional, or mixed types, using keyword rules such as "how many" for counting, question prefixes such as "is there" for existence, and spatial terms such as "left", "behind", or "on" for spatial queries. We group relations into coarse structural types: spatial (e.g., *on*, *near*, *behind*), functional (e.g., *supports*, *provides-light-for*), instance (e.g., *is-a*, *part-of*), and other. Spatial queries upweight spatial relations, functional queries upweight functional relations, and the remaining query types use weaker or uniform weights.

Rank Fusion. The semantic and structural scores operate on different scales and capture distinct forms of relevance. We therefore fuse the two rankings at the rank level using reciprocal rank fusion (RRF) (Cormack et al., 2009):

$$\mathrm{RRF}(e) = \frac{1}{\gamma + \mathrm{rank}_{\mathrm{sem}}(e)} + \frac{1}{\gamma + \mathrm{rank}_{\mathrm{struct}}(e)}, \tag{4}$$

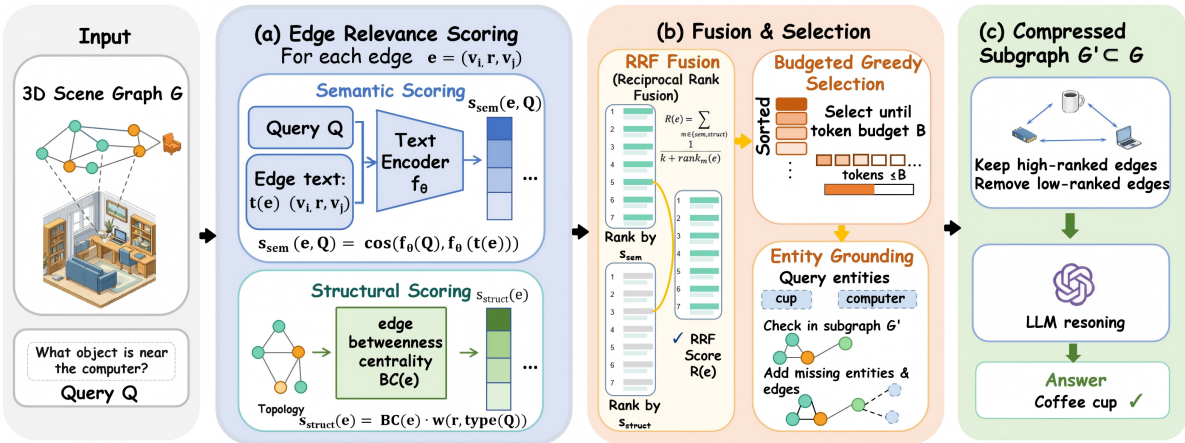

Figure 2: Overview of GraphSelect. The method proceeds in three steps: (a) scoring each edge with semantic and structural relevance, (b) fusing the rankings via reciprocal rank fusion, and (c) greedily selecting high-ranked edges under a token budget, with a lightweight entity-grounding check to retain query-mentioned objects.

where $\mathrm{rank}_{\mathrm{sem}}(e)$ and $\mathrm{rank}_{\mathrm{struct}}(e)$ are the descending ranks of edge e according to s_{sem} and s_{struct}, respectively, and we set $\gamma = 60$ as the smoothing constant, following the standard reciprocal-rank-fusion setting in information retrieval. This rank-level fusion avoids direct score calibration and allows semantic and structural signals to contribute jointly to edge selection.

This choice keeps the method parameter-light and transferable. A weighted sum would require tuning a dataset-specific trade-off between semantic and structural scores, whereas RRF only depends on the relative ranking induced by each signal. Thus, the same fusion rule applies across datasets and budget levels without additional validation data or dataset-specific calibration.

3.3 Budgeted Subgraph Selection

After ranking all edges by $\mathrm{RRF}(e)$, GraphSelect extracts a compact subgraph through budgeted greedy selection followed by entity grounding.

Greedy Edge Selection. We sort all edges in descending order of $\mathrm{RRF}(e)$ and initialize an empty selected set $\mathcal{E}' = \emptyset$. We then iterate through the ranked list and add an edge only if the serialized selected subgraph remains within the budget:

$$\mathrm{tokens}(\mathcal{E}' \cup \{e\}) \leq B. \tag{5}$$

Edges exceeding the budget are skipped. This keeps the highest-ranked edges while enforcing the token constraint during selection.

Entity Grounding. A strict token budget may remove objects explicitly mentioned in the query. To reduce this risk, GraphSelect performs a final grounding check. It first extracts query-mentioned entities using a predefined object vocabulary. If an extracted entity is absent from the selected subgraph, GraphSelect considers its adjacent edges in the original graph and adds the highest-ranked feasible edge. This step also follows the hard budget constraint in Equation (5). Thus, entity grounding improves coverage of query entities without increasing the token budget. Entity grounding is also budget-aware: for a missing query entity, GraphSelect adds its best adjacent edge only if the updated subgraph still satisfies $\mathrm{tokens}(E' \cup \{e\}) \leq B$; otherwise the edge is skipped. In a 405-run audit spanning synthetic scenes, ScanQA, and SQA3D, entity grounding never exceeded the target budget.

4 Experiments

We evaluate GraphSelect on ScanQA and SQA3D using the 3DGraphLLM protocol (Zemskova and Yudin, 2025). GraphSelect only modifies the serialized scene-graph input and leaves the downstream LLM and decoding pipeline unchanged.

4.1 Experimental Setup

Datasets. We use the ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2023) validation sets, both built on ScanNet (Dai et al., 2017). ScanQA contains 4,675 question-answer pairs across 312 indoor scenes, while SQA3D contains 3,261 situated questions grounded to agent poses in the same validation scenes. We also construct a small synthetic set for sanity checks and controlled ablations.

Synthetic Evaluation Set. For sanity checks and controlled ablations, we also construct a synthetic

Algorithm 1 GraphSelect

Require: Scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, query Q, token budget B, encoder f_θ
Ensure: Selected edge set $\mathcal{E}'$

Compute $s_{\text{sem}}(e, Q)$ and $s_{\text{struct}}(e)$ for each $e \in \mathcal{E}$
Compute $\text{RRF}(e)$ for each $e \in \mathcal{E}$
Sort $\mathcal{E}$ by $\text{RRF}(e)$ in descending order
Initialize $\mathcal{E}' \leftarrow \emptyset$
for each edge e in sorted $\mathcal{E}$ **do**
  if $\text{tokens}(\mathcal{E}' \cup \{e\}) \leq B$ **then**
    $\mathcal{E}' \leftarrow \mathcal{E}' \cup \{e\}$
  end if
end for
Extract query-mentioned entities from Q
for each extracted entity absent from $\mathcal{E}'$ **do**
  Find its highest-ranked adjacent edge $e^\star$ in $\mathcal{G}$
  if $\text{tokens}(\mathcal{E}' \cup \{e^\star\}) \leq B$ **then**
    $\mathcal{E}' \leftarrow \mathcal{E}' \cup \{e^\star\}$
  end if
end for
return $\mathcal{E}'$

evaluation set of 10 indoor scenes from three templates (living room, kitchen, and bedroom), yielding 102 QA pairs in total. Each scene contains 12 objects and 24.6 edges on average, and each template defines a fixed object inventory, base relations, and manually written questions covering spatial, existence, counting, and functional reasoning patterns. To further increase difficulty, we add up to five noise edges per scene using generic spatial relations such as *near*, *on*, and *next-to*.

Scene Graph Construction. Since ScanQA and SQA3D do not directly provide scene graphs, we follow the 3DGraphLLM preprocessing pipeline (Zemskova and Yudin, 2025). Object nodes are instantiated from Mask3D-derived ScanNet attributes, which provide category labels and 3D bounding boxes for up to 100 candidate instances per scene. Edges are generated over ordered object pairs, and duplicate triples (o_i, r, o_j) are removed while preserving distinct relations between the same pair. We use seven geometric relations: *near*, *next-to*, *on*, *above*, *below*, *in-front-of*, and *behind*, assigned by fixed distance, overlap, and relative-position rules. In the ScanNet validation scenes used by ScanQA and SQA3D, each scene retains about 70–95 valid labeled objects after filtering and contains 2,008.3 edges on average.

Evaluation Protocol and Metrics. We use the 3DGraphLLM evaluation pipeline with a 4-bit quantized LLaMA-3-8B checkpoint trained on ScanQA and SQA3D. The input is a serialized scene graph concatenated with the question text. We report Exact Match (EM1), refined Exact Match with synonym matching (EM1_ref), ROUGE-L, and CIDEr using the official ScanQA evaluation script. For SQA3D, we also report per-type EM1_ref across six question categories.

Implementation Details. GraphSelect is implemented in Python 3.10. We use BAAI/bge-large-en-v1.5 (Xiao et al., 2024) as the frozen sentence encoder via HuggingFace Transformers (Wolf et al., 2020), compute betweenness centrality with NetworkX (Hagberg et al., 2007), and use the LLaMA tokenizer from the 3DGraphLLM pipeline (Zemskova and Yudin, 2025) for token counting. All experiments run on a single NVIDIA RTX 3090 GPU for BGE encoding, while the remaining pipeline is CPU-only. On the real ScanNet setting, encoding 300,000 edge texts takes about 7 minutes, followed by about 5 minutes for selection and evaluation.

Compression Measurement. GraphSelect selects scene-graph edges under a target token budget. We report the number of selected edges, the target token budget, and the edge compression ratio. Because each edge is serialized in a nearly uniform text format, edge count tracks graph-token count; on a 200-sample ScanQA audit, the correlation between token compression and edge compression is $r = 0.9991$. The realized compression ratio also closely matches the intended target across audited datasets, with at most a 2.0-point deviation on synthetic scenes and at most a 0.1-point deviation on ScanQA and SQA3D. Unless otherwise stated, the 25%, 50%, and 75% settings refer to target token budgets relative to the full serialized graph.

Full-Graph Scale. The full serialized scene graphs are long before compression. In an audit of 200 samples per real benchmark, full-graph prompts average 10.8k tokens on ScanQA and 14.4k tokens on SQA3D, as summarized in Table 1.

Budget Accuracy. The realized compression ratio closely matches the intended target across audited datasets, with at most a 2.0-point deviation on synthetic scenes and at most a 0.1-point deviation on ScanQA and SQA3D (Table 2).

Table 1: Full-graph token statistics across datasets. Synthetic scenes contain about 25 edges, while ScanQA and SQA3D contain about 2,000 edges on average.

Dataset	Mean Tokens	Std. Dev.	Range
Synthetic (102 QA pairs)	156.9	12.9	134–171
ScanQA	10,838.6	5,717.4	3,701–18,530
SQA3D	14,380.9	9,065.5	1,398–43,407

Table 2: Realized compression ratio versus target budget. GraphSelect stays within 2.0 percentage points of the intended compression target across all audited settings.

Dataset	Budget	Compression	Deviation
Synthetic	25%	76.8%	+1.8%
Synthetic	50%	51.9%	+1.9%
Synthetic	75%	27.0%	+2.0%
ScanQA	25%	75.1%	+0.1%
ScanQA	50%	50.1%	+0.1%
ScanQA	75%	25.1%	+0.1%
SQA3D	25%	75.0%	+0.0%
SQA3D	50%	50.0%	+0.0%
SQA3D	75%	25.0%	+0.0%

Baselines. We compare GraphSelect with four baselines: (1) **Full Graph**, which uses all edges without compression; (2) **Degree Centrality**, which selects edges by endpoint degree and represents static topology-based pruning; (3) **Random**, which samples edges until the budget is filled; and (4) **Entity-only**, which keeps edges adjacent to query-mentioned objects but ignores multi-hop paths. We also evaluate two ablations: **w/o Semantic**, which removes the query-edge semantic signal, and **w/o Structural**, which removes the betweenness-based structural signal.

4.2 Quantitative Analysis

The results show that GraphSelect preserves downstream QA performance under strong compression on both ScanQA and SQA3D. On ScanQA, the 50% budget reduces the graph from 2,008 to 894 edges, achieving 55.5% edge compression, while retaining 99.1% of the full-graph EM1 score (0.225 vs. 0.227) and slightly improving CIDEr from 0.853 to 0.858. On SQA3D, all budget settings stay close to the full graph, with EM1 ranging from 0.545 to 0.553 around the full-graph baseline of 0.547. These results show that dense scene graphs contain substantial query-irrelevant redundancy, and that many edges can be removed without materially degrading answer quality.

The improvement in CIDEr suggests that compression can improve not only inference efficiency but also the quality and specificity of generated responses. Full graphs often include many relations that are valid in the scene but irrelevant to the current question and its required reasoning path. When such relations are serialized into the prompt, they can compete with the target evidence and encourage the model to generate less specific or weakly grounded answers. By filtering these distracting edges, GraphSelect provides a cleaner, more focused context that can better align the generated answer with the reference answer.

Table 3: Baseline comparison on ScanQA at a 50% budget.

Method	Edges	Compression	EM1	EM1_ref	CIDEr
Full graph	2,008	0%	0.227	0.397	0.853
GraphSelect 50%	894	55.5%	0.225	0.402	0.858
Degree 50%	789	60.7%	0.222	0.398	0.849

Table 4: Ablation study on SQA3D at a 50% budget.

Method	Nominal Tokens	Edges	CIDEr	EM1
GraphSelect (full method)	8,250	894	1.481	0.545
w/o Semantic signal	8,250	880	1.455	0.543
w/o Structural signal	8,250	910	1.453	0.542

4.3 Qualitative Analysis

Figure 3 provides three representative examples under the same 50% budget. In all three cases, the full graph contains the necessary evidence but also many irrelevant relations. GraphSelect preserves the key reasoning paths and returns the correct answer, whereas degree centrality and random pruning often remove critical bridge edges, such as lamp→bookshelf or desk→office-chair, causing the reasoning chain to break. This comparison suggests that GraphSelect does not simply retain high-frequency or high-degree relations, but instead preserves the structurally useful and query-relevant connections needed for multi-hop scene reasoning.

4.4 Ablation Studies

Effect of Semantic and Structural Signals. Table 4 isolates the two scoring signals on SQA3D under the 50% budget. Removing either signal reduces both CIDEr and EM1, which shows that semantic relevance and structural importance capture complementary evidence. The semantic signal selects relations that are directly relevant to the query, while the structural signal preserves bridge edges needed for multi-hop reasoning.

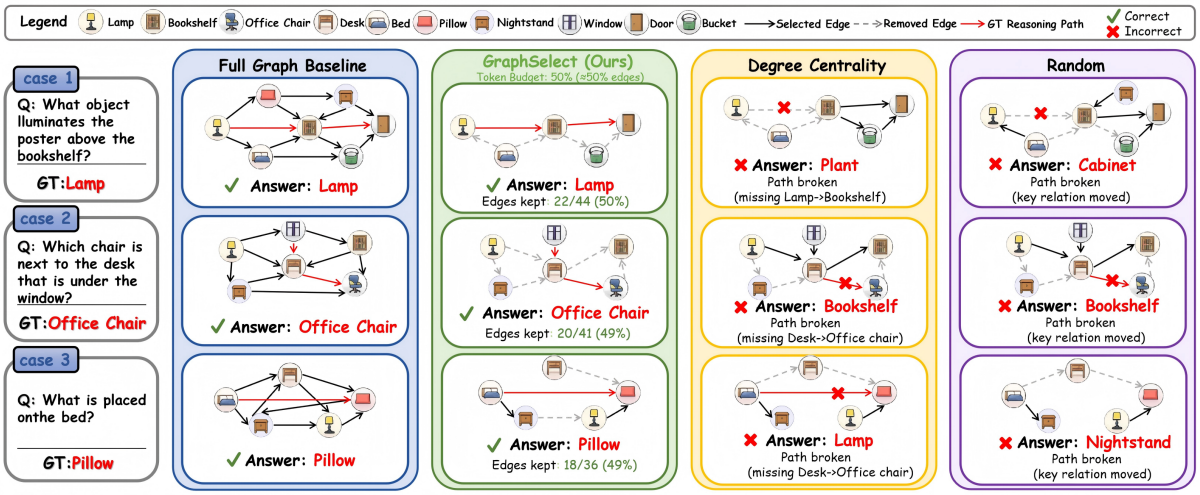

Figure 3: Qualitative comparison at a matched 50% budget. GraphSelect preserves query-relevant paths and recovers correct answers, while static or random pruning often removes key relations and breaks reasoning chains.

Table 5: Complete results at all budget levels on **synthetic indoor scenes** (10 scenes, 102 QA pairs). Full-graph accuracy is 77.0%. GraphSelect maintains 97% of the full-graph score at 50% compression, while degree centrality drops to 79% and entity-only to 50%.

Method	Accuracy	% Full Graph	Compression	Avg. Edges
Full graph	77.0%	100%	0%	25.0
GraphSelect 25%	68.6%	89%	77%	5.5
GraphSelect 50%	75.0%	97%	52%	11.9
GraphSelect 75%	77.0%	100%	27%	18.2
Random 25%	62.3%	81%	77%	5.3
Random 50%	73.0%	95%	52%	11.5
Random 75%	76.0%	99%	27%	17.9
Degree 25%	44.1%	57%	77%	5.4
Degree 50%	60.8%	79%	52%	11.4
Degree 75%	71.1%	92%	28%	17.5
Entity-only	38.7%	50%	86%	3.2
Ablation at 50% budget				
– Semantic	63.2%	82%	52%	12.1
– Structural	77.0%	100%	52%	11.5

Robustness Across Question Types. Table 6 reports per-type EM1_ref on SQA3D. GraphSelect remains close to the full graph across spatial, relative, scene understanding, yes/no, direction, and mixed questions. This suggests the method is not tied to a question format and can preserve performance across different query types.

Controlled Synthetic Evaluation. To complement the real-scene benchmarks, we further evaluate GraphSelect on the synthetic set described above under controlled scene structures and relation patterns. On this controlled set, GraphSelect retains 97% of the full-graph score at 50% compression (75.0% vs. 77.0%), whereas degree centrality retains only 79% and entity-only retains 50%. This result supports the intuition that query-aware selection better preserves the reasoning structure required for answering spatial questions than static topology-based pruning strategies that ignore query intent and local reasoning context. The complete synthetic-scene results are summarized in Table 5.

Generalization to a Text-Only LLM. To test whether GraphSelect depends on 3DGraphLLM, we further evaluate it with a text-only Qwen-based LLM on 200 ScanQA validation samples. The model receives only serialized scene-graph text, without geometric coordinates, GNN features, or visual inputs from the original 3D pipeline. The full graph yields 0.235 EM1, while GraphSelect at a 50% budget achieves 0.205 EM1, retaining about 87% of full-graph performance.

Although the absolute score is lower than that of 3DGraphLLM, this result indicates that GraphSelect still preserves useful graph evidence beyond a single downstream architecture.

This result is also useful because the text-only model does not receive geometric coordinates, GNN features, or visual inputs. The retained performance therefore comes from the serialized relational evidence itself. Although stronger 3D-specific backbones remain preferable, the experiment indicates that GraphSelect is not tightly coupled to a particular multimodal architecture.

5 Conclusion

We presented GraphSelect, a parameter-free and query-aware scene graph compression method for efficient 3D scene reasoning with LLMs. By combining query-edge semantic relevance with structural importance, GraphSelect removes query-irrelevant relations and preserves the compact evidence needed for downstream reasoning. Experi-

Table 6: Per-type EM1_ref on SQA3D.

Method	Spatial	Relative	Scene U.	Yes/No	Direction	Mixed
Full graph	0.570	0.658	0.542	0.645	0.500	0.539
GraphSelect 25%	0.584	0.658	0.542	0.645	0.470	0.531
GraphSelect 50%	0.581	0.638	0.517	0.629	0.485	0.570
GraphSelect 75%	0.591	0.651	0.550	0.661	0.470	0.531

ments with 3DGraphLLM on ScanQA and SQA3D show that pruning redundant graph edges can improve answer quality rather than merely reduce input length. For example, on ScanQA, GraphSelect at the 50% budget improves CIDEr from 0.853 to 0.858 while using a much smaller graph. This suggests that dense 3D scene graphs contain not only redundant but also distracting relations, and that query-aware edge selection can provide cleaner context for LLM reasoning. The current method uses a keyword-based query-type heuristic, which may introduce coarse or noisy structural weights. Future work will explore learned query typing, LLM backbones, and token-efficient interfaces for multimodal scene understanding.

6 Limitations

GraphSelect has several limitations. First, its structural score uses a lightweight keyword-based query-type heuristic, which may assign noisy types to ambiguous queries. Second, entity grounding relies on a predefined object vocabulary and near-exact string matching, making it brittle under open-vocabulary or varied phrasing. Third, our main evaluation uses 3DGraphLLM as the downstream backbone, so broader validation across other LLM-based 3D reasoning systems remains future work. These limitations define the current scope but do not change the main finding that query-aware edge selection can reduce prompt size while preserving reasoning quality. Future work will explore learned query typing, more robust entity linking, and broader backbone evaluations.

References

Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 2019. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5664–5673.

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*, pages 19129–19139.

Ulrik Brandes. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

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

Christos Faloutsos, Kevin S. McCurley, and Andrew Tomkins. 2008. Fast Discovery of Connection Subgraphs. *Knowledge and Information Systems*, 14(1):81–108.

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.

Aric Hagberg, Pieter J Swart, and Daniel A Schult. 2007. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL).

Anna-Maria Halacheva, Jan-Nico Zaech, Xi Wang, Danda Pani Paudel, and Luc Van Gool. 2025. Gaussianvlm: Scene-centric 3d vision-language models using language-aligned gaussian splats for embodied reasoning and beyond. *IEEE Robotics and Automation Letters*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Hao-Yu Hou, Chun-Yi Lee, Motoharu Sonogashira, and Yasutomo Kawanishi. 2025. Fross: Faster-than-real-time online 3d semantic scene graph generation from rgb-d images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28818–28827.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 13358–13376.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.

Gregory Kamradt. 2023. Needle In A Haystack — pressure testing LLMs. *GitHub Repository*.

Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. 2024. 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.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 6342–6353.

Xiaojian 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*.

Saket Navlakha, Rajeev Rastogi, and Nisheeth Shrivastava. 2008. Graph summarization with bounded error. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 419–432.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4498–4507.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4231–4242.

Shun Taguchi, Hideki Deguchi, Takumi Hamazaki, and Hiroyuki Sakai. 2025. Spatialprompting: Keyframe-driven zero-shot spatial reasoning with off-the-shelf multimodal large language models. *arXiv preprint arXiv:2505.04911*.

Yuanyuan Tian, Richard A Hankins, and Jignesh M Patel. 2008. Efficient aggregation for graph summarization. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 567–580.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, pages 613–622. IEEE.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. 2020. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Jintang Xue, Ganning Zhao, Jie-En Yao, Hong-En Chen, Yue Hu, Meida Chen, Suya You, and C-C Jay Kuo. 2026. Descrip3d: Enhancing large language model-based 3d scene understanding with object-level text descriptions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1746–1756.

Qi Xun Yeo, Yanyan Li, and Gim Hee Lee. 2025. Statistical confidence rescoring for robust 3d scene graph generation from multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24999–25008.

Tatiana Zemskova and Dmitry Yudin. 2025. 3dgraphllm: Combining semantic graphs and large language models for 3d scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8885–8895.

A Additional Details

A.1 Real ScanNet Scene Graph Construction

Section 4.1 summarizes the real-scene graph construction pipeline. We report the relation-specific thresholds used to instantiate the seven geometric relations. The *near* relation is triggered when the 3D center distance is below 3.0 m, with a fallback threshold of 4.5 m when no other relation applies. The *next-to* relation uses a horizontal-distance threshold of 1.5 m and a vertical-offset constraint to exclude stacked objects. The *on*, *above*, and *below* relations require horizontal overlap between projected boxes; *on* uses a tight vertical gap threshold of 0.3 m, whereas *above* and *below* allow a signed height difference up to 1.5 m. The *in-front-of* and *behind* relations use a 2D proximity threshold of 2.0 m and a minimum signed front/back offset of 0.3 m. Synthetic-only relations, including *provides-light-for*, *in*, and *under*, are used only in synthetic scenes. For structural relevance, we collapse the directed relation graph into an undirected simple graph in NetworkX so that all relations between the same unordered object pair share a single topological edge. Betweenness centrality is then computed on this collapsed topology and mapped back to the original directed edges.

A.2 Implementation and Token-Budget Audits

The core implementation details are provided in Section 4.1. Synthetic scene experiments complete in under 5 minutes, including BGE loading. Real ScanNet experiments require approximately 7 minutes for BGE encoding of 300,000+ edge texts, followed by 5 minutes of selection and evaluation.

To validate our compression metrics and ensure fair comparison across methods, we audit token-level statistics on three datasets: 10 synthetic scenes with 102 QA pairs, 200 ScanQA samples, and 200 SQA3D samples. Token counting uses the LLaMA tokenizer from 3DGraphLLM.

Token vs. Edge Compression. On the 200-sample ScanQA audit, token compression and edge compression are nearly perfectly aligned: the Pearson correlation coefficient is $r = 0.9991$, the mean difference between the two ratios is -0.001, and the standard deviation is 0.009. This behavior reflects the near-uniform edge serialization format, which makes edge count a reliable proxy for token count throughout the main experiments.

Method Fairness. At the fixed 50% budget, all methods have nearly identical realized token counts. On synthetic scenes, GraphSelect averages 75.7 tokens, while Random and Degree use 75.4 and 75.8 tokens. On the ScanQA audit subset, GraphSelect and Random both use 5,415.4 tokens, versus 5,416.9 for Degree. The maximum mean difference is only 1.5 tokens, confirming that performance differences arise from edge-selection quality rather than hidden budget advantages.

Entity-Only Baseline. The entity-only baseline keeps only edges adjacent to query-mentioned entities and therefore operates well below the nominal budget. It uses 20.8 tokens on average on synthetic scenes, corresponding to 86.7% compression, and 826.5 tokens on the ScanQA audit subset, corresponding to 92.4% compression. The high variance on ScanQA, with a standard deviation of 619.9 and a minimum of 0, indicates cases where entity matching retains no edges, explaining the poor accuracy of entity-only selection in Table 5.

Budget Compliance. Entity grounding remains subject to the same hard budget check used in the main algorithm: a missing query entity is restored only if adding its best adjacent edge still satisfies $\mathrm{tokens}(E' \cup \{e\}) \leq B$. Across 405 audited runs spanning synthetic scenes, ScanQA, and SQA3D, we observed zero budget violations.

A.3 Query Type Classification Details

GraphSelect classifies queries into five types with a priority-based keyword heuristic. The rules are applied sequentially, so lower-priority rules are skipped once a higher-priority rule is matched.

Classification Rules. Classification rules are applied in the following priority order: counting queries are detected by "how many", "count", or "number of"; existence queries by prefixes such as "is there", "are there", "does", or "do"; spatial queries by spatial terms such as "left", "right", "behind", "near", "above", "below", "on", and "where"; functional queries by action or affordance cues such as "used", "purpose", "function", "store", "open", and "hold"; all remaining queries are assigned to the mixed type.

Relation Type Classification. Relations are mapped to four coarse structural types before weighting: **spatial** relations include *on*, *near*, *next-to*, *between*, *left-of*, *right-of*, *above*, *below*,

under, *in-front-of*, and *behind*; **functional** relations include *supports*, *contains*, *enables*, *used-for*, *provides-light-for*, *illuminates*, and *blocks*; **instance** relations include *is-a*, *part-of*, *has*, and *type-of*; all remaining relations are assigned to **other**.

Structural Weight Matrix. Table 7 lists the full weight matrix $w(r, \mathrm{type}(Q))$ used in Equation (3).

Table 7: Structural weight matrix. Only *spatial* and *functional* queries receive high weights when paired with their corresponding relation types.

Query type	spatial	functional	instance	other
spatial	1.0	0.3	0.3	0.3
functional	0.3	1.0	0.3	0.3
counting	0.3	0.3	0.3	0.3
existence	0.3	0.3	0.3	0.3
mixed	0.5	0.5	0.5	0.5

Classification Distribution on ScanQA. In a 500-sample audit of the ScanQA validation set, the query-type distribution is 90.4% spatial, 9.4% counting, and 0.2% existence, with no functional or mixed queries under the current heuristic. This skew reflects the spatial focus of ScanQA and the coarse nature of keyword-based typing. Despite this imbalance, GraphSelect maintains strong performance because the semantic relevance score can compensate for noisy or coarse structural weights.

Limitations. The keyword-based classifier is intentionally lightweight and may assign coarse query types for ambiguous or underspecified questions. For example, short spatial prepositions can be difficult to disambiguate without syntactic parsing, and generic functional cues may overlap with ordinary question wording. Despite this coarse typing, GraphSelect remains robust because the final ranking also depends on query-edge semantic relevance. In practice, the dual-signal design provides graceful degradation: when structural typing is weak or noisy, the semantic signal can still dominate the final ranking. A learned query classifier or phrase-level parser would be a natural extension for improving structural weighting in future work.