Edge-Centric Relational Reasoning for 3D Scene Graph Prediction

Yanni Ma[1,2], Hao Liu[3,4], Yulan Guo[*1], Theo Gevers[2], Martin R. Oswald[2]

[1]School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen, China
[2]Computer Vision Research Group, University of Amsterdam, Netherlands
[3]Key Lab of Spatial-temporal Big Data Analysis and Application of Natural Resources in Megacities, Ministry of Natural Resources, East China Normal University (ECNU), Shanghai, China
[4] Key Laboratory of Geographic Information Science (Ministry of Education), ECNU, Shanghai, China
mayn3@mail2.sysu.edu.cn, hliu@geoai.ecnu.edu.cn, guoyulan@sysu.edu.cn, th.gevers@uva.nl, m.r.oswald@uva.nl

Abstract

3D scene graph prediction aims to abstract complex 3D environments into structured graphs consisting of objects and their pairwise relationships. Existing approaches typically adopt object-centric graph neural networks, where relation edge features are iteratively updated by aggregating messages from connected object nodes. However, this design inherently restricts relation representations to pairwise object context, making it difficult to capture high-order relational dependencies that are essential for accurate relation prediction. To address this limitation, we propose a **L**ink-guided **E**dge-centric relational reasoning framework with **O**bject-aware fusion, namely **LEO**, which enables progressive reasoning from relation-level context to object-level understanding. Specifically, LEO first predicts potential links between object pairs to suppress irrelevant edges, and then transforms the original scene graph into a line graph where each relation is treated as a node. A line graph neural network is applied to perform edge-centric relational reasoning to capture inter-relation context. The enriched relation features are subsequently integrated into the original object-centric graph to enhance object-level reasoning and improve relation prediction. Our framework is model-agnostic and can be integrated with any existing object-centric method. Experiments on the 3DSSG dataset with two competitive baselines show consistent improvements, highlighting the effectiveness of our edge-to-object reasoning paradigm.

Introduction

3D scene graph prediction (SGP) is a fundamental yet challenging task in 3D scene understanding, which aims to represent complex environments as structured graphs by recognizing objects and modeling their relationships. Unlike conventional 3D scene understanding tasks, such as 3D object detection (Liu et al. 2023a,b; Ao et al. 2022) and semantic segmentation (Ma et al. 2020; Liu et al. 2021; Dang et al. 2023), which mainly focus on identifying individual objects, 3D SGP goes a step further by capturing the rich semantic and spatial relations among them. In a 3D scene graph, nodes correspond to object instances, and edges encode pairwise relations, such as "behind", or "attached to". This representation captures both the spatial arrangement and semantic interactions of objects, providing a compact and interpretable abstraction for high-level reasoning. Consequently, 3D SGP has been increasingly adopted in downstream tasks such as 3D visual grounding (Chen, Chang, and Nießner 2020; Achlioptas et al. 2020), visual question answering (Azuma et al. 2022; Yan et al. 2021; Zhao et al. 2022) and embodied AI (Duan et al. 2022; Yang et al. 2024).

To model such structured representations, existing methods widely adopt graph neural networks (GNNs), with various architectural designs proposed for enhancing relational reasoning. For instance, SGPN (Wald et al. 2020) and KISGP (Zhang et al. 2021b) employ fully-connected GNNs to propagate messages between object nodes and their associated relations. 3DSMKA (Feng et al. 2023) introduces a

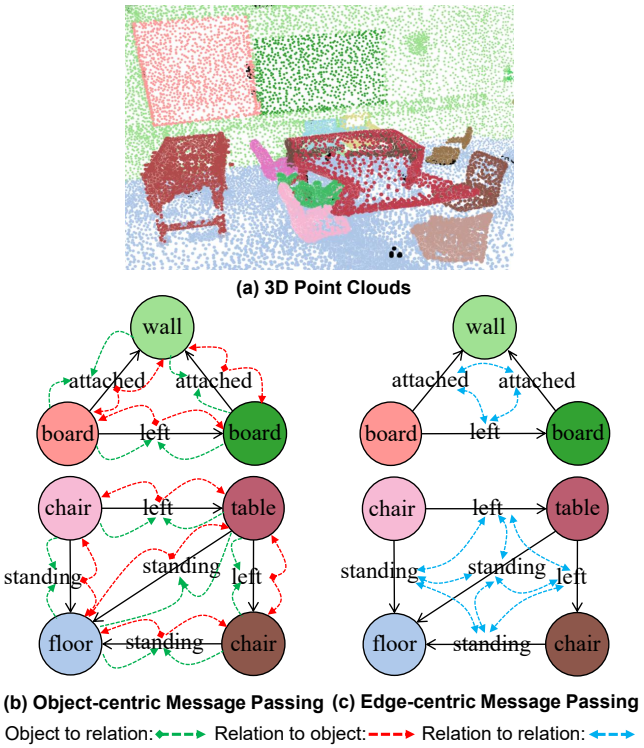

Figure 1: Object-centric message passing vs. Edge-centric message passing.

[*]Corresponding author.

hierarchical graph to capture multi-level spatial and semantic structures, while 3DHetSGP (Ma et al. 2025) constructs a heterogeneous graph to explicitly account for the diversity of relation types. These models have demonstrated strong performance on the 3DSSG benchmark, highlighting the importance of GNN-based relational reasoning. However, most existing methods follow an object-centric paradigm, where messages are iteratively propagated between objects and relations. Although this bidirectional design allows modeling of local object–relation interactions, it inherently restricts the relational reasoning to pairwise context, as each relation is updated independently based on its connected objects. As a result, these methods struggle to capture higher-order relational dependencies such as predicate co-occurrence or contextual consistency among relations.

This limitation is particularly evident in indoor scenes (Wald et al. 2020), where object semantics and geometric layouts alone are often insufficient for accurate relation prediction. For example, semantic ambiguity is a common issue: relations like "hanging on", "attached to", and "supported by" often share similar spatial cues and are easily misclassified. Moreover, spatial relations are often interdependent, *e.g.*, an object *to the left of* another may also be *behind* it, depending on global scene layout. Without modeling such relation dependencies, object-centric approaches often produce inaccurate or inconsistent scene graphs.

To address these limitations, we propose a **L**ink-guided **E**dge-centric relational reasoning framework with **O**bject-aware fusion, namely **LEO**, which enables progressive reasoning from relation-level context to object-level understanding. Specifically, LEO first predicts soft link weights for each edge in the original scene graph to modulate the message passing strength. Then, we transform the original graph into a line graph, where each node represents a relation and two nodes are connected if the associated relations share a common object. A line graph neural network (LineGNN) is applied to perform edge-centric relational reasoning, enabling each relation to aggregate contextual information from other semantically related relations. Finally, the enhanced relation features are then integrated back into the original object-centric graph to improve object-level reasoning and relation prediction. LEO is model-agnostic and can be seamlessly integrated into existing object-centric approaches. Our main contributions are summarized as:

- We propose **LEO**, a link-guide edge-to-object reasoning framework that performs progressive relational reasoning from edge-centric to object-centric representations for accurate and robust 3D scene graph prediction.
- We introduce a line graph formulation for 3D scene graphs, enabling explicit relation-level reasoning beyond object pairs. To the best of our knowledge, this is the first work to reformulate 3D scene graphs as line graphs for relation-centric reasoning.
- We design a link prediction module that assigns soft weights to object pairs, modulating relation strengths and suppressing noisy or irrelevant message passing.
- Extensive experiments on the 3DSSG dataset show that our LEO consistently improves two strong baselines, demonstrating the effectiveness and generalizability of modeling relation-level dependencies.

Related Works

3D scene graph prediction in point cloud. Recent advances in 3D scene understanding have extended the paradigm of scene graph generation from 2D images to 3D point clouds, which aims to jointly infer object categories and their semantic relationships directly from point cloud data. As the pioneering work, SGPN (Wald et al. 2020) introduces the first large-scale dataset 3DSSG for this task, and proposes a baseline framework that integrates PointNet (Qi et al. 2017) for point-wise feature encoding with an object-centric GNN for joint reasoning over nodes and edges. To better capture contextual dependencies among objects and relations, SGFN (Wu et al. 2021) introduces a feature-wise attention mechanism within the GNN message passing process, while SGGpoint (Zhang et al. 2021a) enhances relation reasoning through a twinning attention module designed for edge-oriented updates. KISGP (Zhang et al. 2021b) further incorporates category-level priors via a graph auto-encoder, enabling knowledge-guided message propagation, whereas 3DSMKA(Feng et al. 2023) leverages both external knowledge bases and spatial priors to construct a hierarchical graph that encodes multiscale semantics and geometry. VL-SAT (Wang et al. 2023) explores vision-language supervision by aligning CLIP-based features (Radford et al. 2021) with object and relation embeddings. More recently, 3DHetSGP (Ma et al. 2025) proposes a heterogeneous graph learning framework that explicitly models the diversity of relation types and mitigates the long-tail distribution of predicates in 3D scene graphs. However, these methods follow an object-centric paradigm, which inherently restricts relation features to pairwise object context, leading to less accurate and robust relation predictions. In contrast, our approach explicitly models the dependencies among relations by reformulating the scene graph into a line graph. This enables edge-centric relational reasoning that captures higher-order context beyond individual object pairs.

Line graph neural network. Line graph (Gross, Yellen, and Anderson 2018) is a classical edge-centric structure derived from an original graph, where each node in the line graph corresponds to an edge in the original graph, and two nodes are connected if their associated edges share a common object. Building upon this concept, Line graph neural network (LineGNN) treat edges as nodes and perform message passing between them, enabling relational reasoning at the edge level. Such models have been successfully applied in diverse domains, including molecular property prediction (Buterez et al. 2025), network neuroscience (Betzel, Faskowitz, and Sporns 2023), traffic forecasting (Wang et al. 2018), and community detection (Chen, Li, and Bruna 2017), where modeling interactions among edges is essential. These successes highlight the potential of edge-centric reasoning, motivating our adoption of LineGNN to enhance relation-level reasoning in 3D scene graphs.

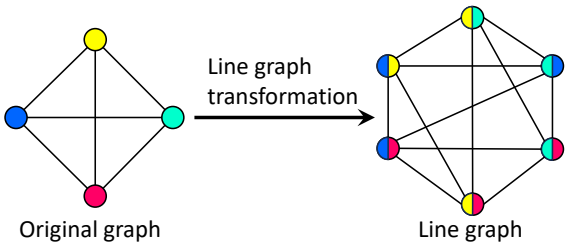

Figure 2: Line Graph Transformation.

Preliminary

Line Graph. Given an original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, its line graph $\mathcal{L}(\mathcal{G}) = (\mathcal{V}_\mathcal{L}, \mathcal{E}_\mathcal{L})$ is a derived graph where each node corresponds to an edge in $\mathcal{G}$. Two nodes in $\mathcal{L}(\mathcal{G})$ are adjacent if and only if their corresponding edges in $\mathcal{G}$ share a common node. For a fully connected original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the number of nodes in its line graph equals the number of the edges in $\mathcal{G}$, i.e., $|\mathcal{V}_\mathcal{L}| = |\mathcal{E}|$, and the number of edges in the line graph becomes $|\mathcal{V}_\mathcal{L}|(|\mathcal{V}| - 2)$. Such dense connectivity in $\mathcal{L}(\mathcal{G})$ enables comprehensive modeling of relation-level context.

Methods

Overview

Given a segmented point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ with N 3D points and a set of K class-agnostic instances masks $\mathcal{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_K\}$, the objective of 3D scene graph prediction is to construct a structured graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of the 3D scene by identifying object instances $\mathcal{V}$ and predicting relations predicates $\mathcal{E}$ between them.

As illustrated in Fig. 3, we propose Link-guided Edge-to-Object Reasoning network (LEO), a unified framework that performs sequential edge-centric and object-centric relational reasoning. Specifically, it consists of three key components: (1) Link Prediction assigns soft link weights to object pairs in the original scene graph, modulating relation strengths for subsequent reasoning. (2) Edge-centric Relational Reasoning transforms a line graph from original graph and applies a Line Graph Neural Network (LineGNN) to model inter-relation context. (3) Object-centric Relational Reasoning integrates the enhanced relation features into the original scene graph for final object and predicate classification. This design allows LEO to effectively capture both relation-level dependencies and object-level context, leading to more accurate and consistent 3D scene graph predictions.

3D Scene Graph Initialization

We first build a fully connected primitive graph $G = (V, E)$ to represent all possible pairwise relations among object instances. Following our baseline models KISGP (Zhang et al. 2021b) and 3DHetSGP (Ma et al. 2025), the initial object and edge features are encoded and pretrained separately. The object features f_n are extracted from segmented point sets using a multi-scale PointNet encoder (Qi et al. 2017). For edge features f_e, we compute the difference between subject and object feature pairs, followed by an MLP projection. If object class labels are available (e.g., in the PredCls setting), the object features are substituted with the object label embeddings.

Link Prediction

To mitigate the redundancy from dense connections in the line graph, we introduce a link prediction module to predict the likelihood of correlation between object pairs. The predicted scores are subsequently assigned as soft weights to the edges in the original graph, thereby modulating the relation strengths and suppressing irrelevant message passing during relational reasoning.

Given a set of objects with their initialized features f_i and spatial attributes of bounding boxes b_i, we first compute the geometric embedding:

$$g_i = \phi_b([b_i, c_i, l_i, w_i, h_i, V_i]), \tag{1}$$

where $\phi_b(\cdot)$ is the non-linear transformation. b_i and c_i denote the box corners and centroid, while l_i, w_i, h_i,V_i represent the box length, width, height, volume, respectively. For each pair (i, j), we construct the link features f_{ij}^{link} by concatenating the differences of the initialized object features f_i, f_j and geometric embeddings g_i, g_j:

$$f_{ij}^{link} = \phi_p([(f_i - f_j) \| (g_i - g_j)]), \tag{2}$$

where $\phi_p(\cdot)$ is the non-linear transformation for concatenated features. The link features f_{ij}^{link} are then classified into two link categories (link and non-link) using a linear classifier $\phi_l(\cdot)$, followed by a 2-way softmax that yields the confidence scores for both categories:

$$s_{ij}^{link} = \text{softmax}(\phi_l f_{ij}^{link}) \tag{3}$$

We take the probability of the positive link as the final link confidence score, which serves as the soft link weight s_{ij}^{link} between objects i and j to guide relational reasoning.

Edge-centric Relational Reasoning

Weighted Original Graph. Based on the soft link weights predicted by the link prediction module, we convert the original graph $\mathcal{G}$ to a weighted graph by multiplying the link weights with all edge features.

$$\tilde{f}_{ij} = s_{ij}^{link} \cdot f_{ij} \tag{4}$$

This weighted graph encodes the importance of each relation for modulating the message passing in LineGNN.

Line Graph Transformation. To enable edge-centric relational reasoning, we transform the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ into a line graph $\mathcal{L}(\mathcal{G}) = (\mathcal{V}', \mathcal{E}')$. In this line graph, each node $e_{ij} \in \mathcal{V}'$ represents a directed edge between object nodes (o_i, o_j) in the original graph. There exists an edge between two line nodes e_{ij} and e_{ik} if they share the same object node o_i in $\mathcal{G}$, indicating a contextual relation dependency. Formally:

$$\mathcal{V}' = \{e_{ij} | (o_i, o_j) \in \mathcal{E}\}, \tag{5}$$

$$\mathcal{E}' = \{(e_{ij}, e_{ik}) | e_{ij} \cap e_{ik} = o_i \in V\}. \tag{6}$$

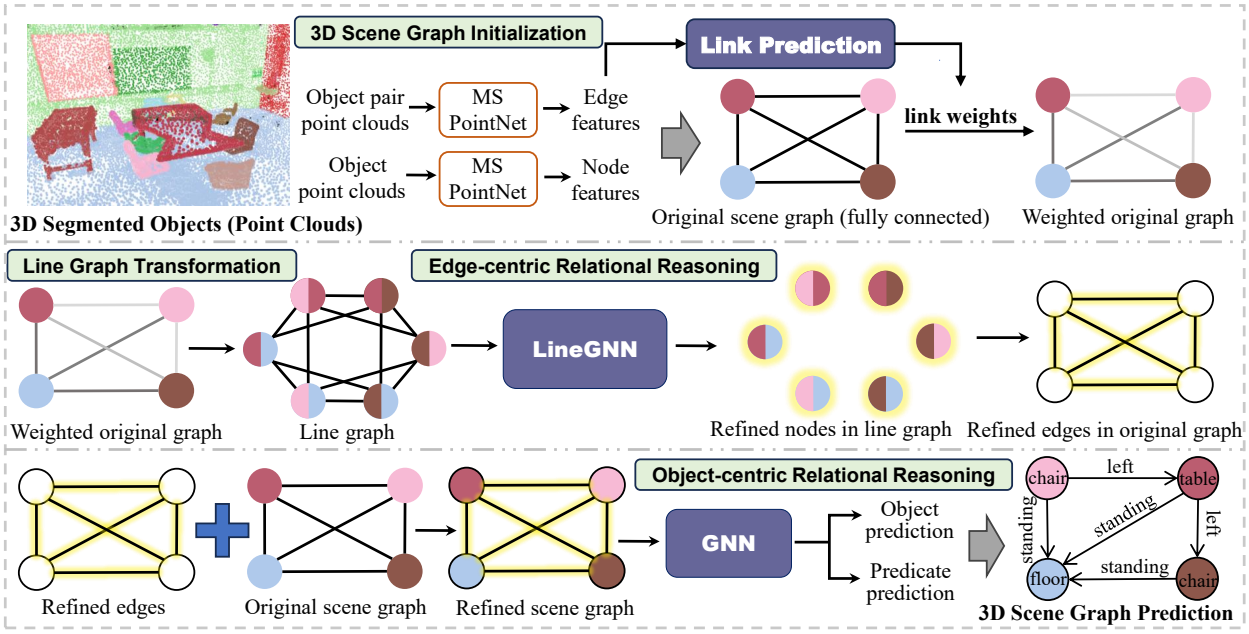

Figure 3: The overview of our LEO framework. It consists of three stages: (a) Link Prediction assigns soft link weights to object pairs in the original scene graph to modulate relation strengths for subsequent reasoning; (b) Edge-centric Relational Reasoning transforms the weighted original graph into a line graph and applies LineGNN to capture relation-level context and refine relation features; (c) Object-centric Relational Reasoning integrates the refined relations into the original graph for final object and predicate prediction.

Line Graph Neural Network. To model interactions among relation edges, we apply a Line Graph Neural Network (LineGNN) over our line graph $\mathcal{L}(\mathcal{G}) = (\mathcal{V}', \mathcal{E}')$ to propagate and aggregate relational context features. In LineGNN, message passing is performed between adjacent relation nodes based on the connectivity defined in $\mathcal{E}'$. Specifically, let $h_{ij}^{(l)}$ denote the hidden state of relation node e_{ij} at layer l. At each layer, the feature of e_{ij} is updated by aggregating messages from its neighboring nodes in $\mathcal{V}'$.

a) Incoming Message. Given a relation node e_{ij}, its incoming message $\boldsymbol{m}_{ij}^{(l)}$ is computed by aggregating the features of its neighboring relation nodes $e_{ik} \in \mathcal{N}(i)$ as:

$$\boldsymbol{m}_{ij}^{(l)} = \mathrm{LN}\left(\sum_{e_{ik} \in \mathcal{N}_{e_{ij}}} \alpha_{ij \to ik}^{(l)} \phi_e(\boldsymbol{h}_{ik}^{(l)}) \right), \tag{7}$$

where $\phi_e(\cdot)$ is a non-linear transformation, $\mathcal{N}_{e_{ij}}$ is the neighboring relations set of relation node e_{ij}, and $\mathrm{LN}(\cdot)$ denotes layer normalization. The attention score $\alpha_{ij \to ik}^{(l)}$ ensures that more relevant neighbors contribute more to the message, computed as:

$$\alpha_{ij \to ik}^{(l)} = \mathrm{softmax}(\phi_{\mathrm{att}}([h_{ij}^{(l)} \| h_{ik}^{(l)}])), \tag{8}$$

where $[\cdot\|\cdot]$ denotes the concatenation operation, $\phi_{\mathrm{att}}(\cdot)$ is a non-linear transformation, the $\mathrm{softmax}(\cdot)$ is computed over all neighbor relations $e_{ik} \in \mathcal{N}_{e_{ij}}$.

b) Edge-centric Updating. Each relation node then updates its hidden state via a gated recurrent unit (GRU):

$$\boldsymbol{h}_{ij}^{(l+1)} = \mathrm{GRU}(\boldsymbol{h}_{ij}^{(l)}, \boldsymbol{m}_{ij}^{(l)}). \tag{9}$$

The relation features $\tilde{h}_{ij}^{(l+1)}$ output from the last LineGNN layer encode rich contextual dependencies among relations and are subsequently utilized as the starting edge features in the first layer of primitive message passing for object-centric reasoning.

Object-centric Relational Reasoning

After enriching the relation features via the LineGNN module, we propagate the updated relation representations to the primitive graph to perform object-centric reasoning. The initial hidden state in this primitive graph is defined as:

$$\boldsymbol{h}_i^{(0)} = f_i, \boldsymbol{h}_{ij}^{(0)} = \tilde{h}_{ij}^{(l+1)}, \tag{10}$$

where $\tilde{h}_{ij}^{(l+1)}$ denotes the output from the lineGNN module.

Each object node receives messages from its connected edges to update its hidden state accordingly. Meanwhile, edge features are also updated based on the hidden states of their associated object nodes.

$$\begin{aligned} \boldsymbol{h}_i^{(l+1)} &= \mathrm{GRU}(\boldsymbol{h}_i^{(l)}, \boldsymbol{m}_i^{(l)}) \\ &= \mathrm{GRU}\Big(\boldsymbol{h}_i^{(l)}, \mathrm{LN}(\sum_{j \in N_{i*}} \phi_e(\boldsymbol{h}_{ij}^{l}))\Big), \end{aligned} \tag{11}$$

$$\begin{aligned} \boldsymbol{h}_{ij}^{(l+1)} &= \mathrm{GRU}(\boldsymbol{h}_{ij}^{(l)}, \boldsymbol{m}_{ij}^{(l)}) \\ &= \mathrm{GRU}\Big(\boldsymbol{h}_{ij}^{(l)}, \mathrm{LN}(\phi_n(\boldsymbol{h}_i^{l}) + \phi_n(\boldsymbol{h}_j^{l}))\Big). \end{aligned} \tag{12}$$

Scene Graph Prediction and Training Objective

To predict the 3D scene graph, we classify the object features f_n and edge features f_e obtained from primitive graph message passing into object and predicate categories.

$$\boldsymbol{s}_n = \mathrm{softmax}\big(\phi_{obj}(\boldsymbol{f}_n)\big), \tag{13}$$

$$\boldsymbol{s}_e = \mathrm{softmax}\big(\phi_{pred}(\boldsymbol{f}_e)\big). \tag{14}$$

Our scene graph prediction involves object classification, predicate classification and link classification. Therefore the overall training objective of is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{obj} + \mathcal{L}_{pred} + \mathcal{L}_{link}, \tag{15}$$

where the $\mathcal{L}_{link}$ refers to link loss of link prediction module. We use focal loss for all loss components.

Experiments

Experimental Settings

Dataset. The 3DSSG dataset provides annotated 3D semantic scene graphs built upon the 3RScan dataset, encompassing 1,482 scans from 478 indoor environments. It contains approximately 48k object nodes and 544k relation edges, capturing rich spatial and semantic relationships among objects. Following prior work, we split the dataset into 3,852 sub-scenes for training and 548 for testing, where each sub-scene contains 4 to 9 objects. 160 object categories and 26 predicate categories are adopted for training and evaluation, consistent with the RIO27 annotation scheme.

Metrics. We evaluate our model on two standard tasks of 3D scene graph prediction: (1) Predicate Classification (PredCls), where the model only predicts the predicate category for each object pair given the ground truth object labels; and (2) Scene Graph Classification (SGCls), which requires the model to predict both object categories and the relationships between objects. For both tasks, we adopt the following metrics: top-k recall (R@k), no-graph-constraint top-k recall (ngcR@k), and mean recall (mR@k). Specifically , R@k measures the propotion of ground truth triplets recalled among the top-k highest-scoring predictions, with constraint that each subject-object pair is assigned with a single predicate. In contrast, ngcR@k allows multiple predicates per object pair without this constraint. The mR@k metric computes the average recall across all predicate categories, providing a more balanced evaluation and better captures performance under long-tail distributions.

Implementation Details. Our model is implemented in PyTorch and trained on a single NVIDIA RTX TITAN GPU. We follow the same training configuration as the baseline models KISGP (Zhang et al. 2021b) and 3DHetSGP (Ma et al. 2025) for fair comparison. Specifically, we use the ADAM optimizer with an initial learning rate of 0.0001 and weight decay of 0.0001. The learning rate decays by a factor of 0.7 every 10 epochs (minimum 1e-8). Following KISGP and 3DHetSGP, we pretrain the object and predicate encoders based on multi-scale PointNet. The LineGNN is inserted before the original object-centric GNN. In particular in 3DHetSGP, which employs three GNN branches for type subgraphs, we also insert LineGNN before GNN independently in each branch. Our experiment run 40 epochs for link prediction, and 50 epochs for relational reasoning stage to predict the final 3D scene graph.

Quantitative Results

Tables 1 and 2 report the results on the PredCls and SGCls tasks, respectively. To evaluate the effectiveness and generalizability of the proposed Le framework, we integrate it into two representative baselines: KISGP and 3DHetSGP, which adopt object-centric graph reasoning architectures and have achieved leading performance on the 3DSSG benchmark. We further compare our method with a set of advanced 3DSGP models in point clouds, including SGPN (Wald et al. 2020), (Wu et al. 2021), SGFormer (Lv et al. 2024), VL-SAT (Wang et al. 2023), and 3DSMKA (Feng et al. 2023). Methods marked with $\star$ are reproduced by us using their open-source code, and all results are evaluated using the KISGP evaluation protocol to ensure consistency.

LEO on KISGP. When integrated into KISGP, LEO brings consistent improvements. On the PredCls task (Table 1), we observe gains of +2.2%, +2.0%, and +0.6% at ngcR@20/50/100, respectively. The improvement at ngcR@20 (from 62.2% to 64.4%) indicates that more correct relations are ranked among the top 20 predictions, suggesting enhanced accuracy in high-confidence relation predictions. The mean recall also increases from 63.5% to 64.7% at mR@50, and from 63.8% to 64.8% at mR@100, indicating more balanced coverage of predicate categories. On the SGCls task (Table 2), LEO also increases ngcR@50 from 34.7% to 37.4%, and mR@50 from 28.1% to 30.4%. This suggests that LEO enhances 3DSGP performance under both settings where object information is provided as ground-truth labels or inferred from visual features.

LEO on 3DHetSGP. We further integrate LEO to a stronger baseline, 3DHetSGP, and observe consistent improvements. On the PredCls task, LEO improves R@20 from 61.8% to 62.9%, ngcR@20 from 70.3% to 73.3%, and mR@20 from 63.7% to 65.8%. The concurrent improvements indicate that the proposed edge-centric relational reasoning improves the accuracy and ranking of predicted relations, particularly among high-confidence predictions. On the SGCls task, we also observe modest gains, particularly in mean Recall@20/50/100, indicating that the model achieves more balanced predictions acro ss predicate categories, including those with lower frequency.

Overall Comparison. Compared to both KISGP and 3DHetSGP, integrating the edge-centric relational reasoning consistently improves performance across various metrics. The gains are more evident on KISGP, which uses a relatively simple GNN-GRU architecture without explicit modeling of relation-level interactions. On 3DHetSGP, which adopts a heterogeneous graph structure with multiple type edges, the improvements are smaller but consistent. These results indicate that edge-centric relational reasoning provides consistent benefits across scene graph models with varying GNN architectures, including both standard and het-

Model	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
Co-Occurrence (Zhang et al. 2021b)	34.7	47.4	47.9	35.1	55.6	70.6	33.8	47.4	47.9
KERN (Chen et al. 2019)	46.8	55.7	56.5	48.3	64.8	77.2	18.8	25.6	26.5
Schemata (Sharifzadeh, Baharlou, and Tresp 2021)	48.7	58.2	59.1	49.6	67.1	80.2	35.2	42.6	43.3
SGPN (Wald et al. 2020)	51.9	58.0	58.5	54.5	70.1	82.4	32.1	38.4	38.9
SGFN* (Wu et al. 2021)	54.5	61.0	61.5	61.4	80.1	90.0	30.5	36.4	36.6
VL-SAT* (Wang et al. 2023)	58.3	65.2	65.8	66.2	85.9	93.9	40.4	47.4	47.7
3DSMKA (Feng et al. 2023)	-	68.3	69.5	-	79.8	89.6	-	66.5	66.9
SGFormer* (Lv et al. 2024)	53.6	59.8	60.2	56.2	71.7	83.5	37.2	43.1	43.4
KISGP (Zhang et al. 2021b)	59.3	65.0	65.3	62.2	78.4	88.3	56.6	63.5	63.8
3DHetSGP* (Ma et al. 2025)	61.8	65.9	65.9	70.3	88.9	94.6	63.7	68.1	68.2
KISGP* (Zhang et al. 2021b)	59.9	65.1	65.4	63.1	79.1	88.6	57.1	61.9	62.0
KISGP+LEO (Ours)	**61.5**	**66.8**	**66.9**	**64.4**	**80.3**	**89.6**	**59.0**	**64.7**	**64.8**
3DHetSGP*	61.8	65.9	65.9	70.3	88.9	94.6	63.7	68.1	68.2
3DHetSGP+LEO (Ours)	**62.9**	65.8	65.8	**73.3**	**90.1**	**95.1**	**65.8**	**68.5**	**68.9**

Table 1: Quantitative results of the evaluated methods in PredCls tasks. Models marked with * indicate reproduced results.

Model	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
Co-Occurrence (Zhang et al. 2021b)	14.8	19.7	19.9	14.1	20.2	25.8	8.8	12.7	12.9
KERN (Chen et al. 2019)	20.3	22.4	22.7	20.8	24.7	27.6	9.5	11.5	11.9
Schemata (Sharifzadeh, Baharlou, and Tresp 2021)	27.4	29.2	29.4	28.8	33.5	36.3	23.8	27.0	27.2
SGPN (Wald et al. 2020)	27.0	28.8	29.0	28.2	32.6	35.3	19.7	22.6	23.1
SGFN* (Wu et al. 2021)	27.2	28.9	28.9	30.0	34.6	36.9	18.2	20.8	20.9
VL-SAT* (Wang et al. 2023)	29.5	31.0	31.2	32.8	37.7	39.8	27.5	29.9	30.0
3DSMKA (Feng et al. 2023)	-	31.5	31.6	-	35.4	37.7	-	30.3	30.6
SGFormer* (Lv et al. 2024)	28.9	30.7	30.8	30.2	34.7	37.2	24.1	26.2	26.2
KISGP (Zhang et al. 2021b)	28.5	30.0	30.1	29.8	34.3	37.0	24.4	28.6	28.8
3DHetSGP* (Ma et al. 2025)	28.5	29.8	29.9	31.2	36.4	38.7	27.3	29.4	29.5
KISGP* (Zhang et al. 2021b)	28.8	30.5	30.6	30.3	34.7	37.6	25.1	28.1	28.3
KISGP+LEO (Ours)	**29.5**	**30.9**	**31.0**	**32.5**	**37.4**	**39.5**	**27.5**	**30.4**	**30.4**
3DHetSGP* (Ma et al. 2025)	**28.5**	29.8	29.9	31.2	36.4	**38.7**	27.3	29.4	29.5
3DHetSGP+LEO (Ours)	28.4	**30.1**	**30.2**	**31.9**	**36.5**	38.5	**29.6**	**32.3**	**32.3**

Table 2: Quantitative results of the evaluated methods in SGCls tasks. Models marked with * indicate reproduced results.

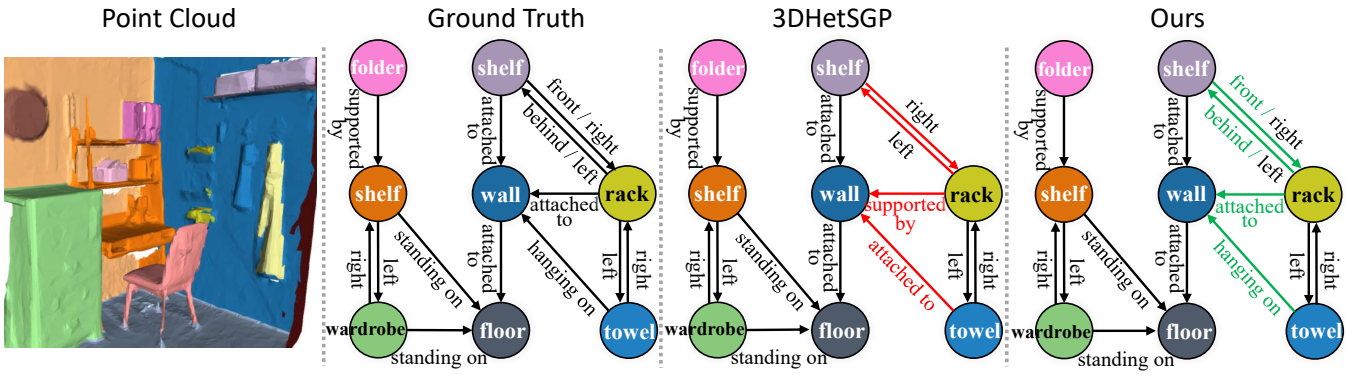

Figure 4: Qualitative results of our model and baseline 3DHetSGP (Ma et al. 2025) on ngcR@20. Red arrows indicate incorrectly predicted relationships. Green arrows indicate relationships that are missed or misclassified by 3DHetSGP but correctly predicted by ours.

erogeneous GNN-based frameworks, suggesting its general applicability to 3D scene graph prediction.

Qualitative Results

Fig 4 presents a qualitative comparison between our method and the baseline 3DHetSGP on ngcR@20. For the object pair *towel* and *wall*, the ground truth relation is *hanging on*. 3DHetSGP identifies the relation predicate as *attached to*, likely due to its limited capacity to distinguish among visually similar predicates. In contrast, our method successfully predicts *hanging on*, demonstrating edge-centric relational reasoning can handle semantic confusion by leveraging relational context. For the pair *shelf* and *rack*, the ground truth includes two directional spatial relations: *behind*, *left* and *front*, *right*. 3DHetSGP fails to predict *behind* and *front* among the top 20 predicted relations. Our method accurately predicts both relations, suggesting that it more effectively captures the inter-relation between spatial predicates through edge-centric relational reasoning. More qualitative results are provided in the supplementary material.

PredCls	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
3DHetSGP (Baseline)	61.8	65.9	65.9	70.3	88.9	94.6	63.7	68.1	68.2
LineGNN (FC)	60.4	64.3	64.4	68.6	88.5	94.6	60.0	66.0	66.1
LineGNN (LP)	62.9	65.8	65.8	73.3	90.1	95.1	65.8	68.5	68.9
LineGNN (GT)	63.2	65.1	65.1	77.8	94.3	97.7	66.6	69.0	69.3
SGCls	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
3DHetSGP (Baseline)	28.5	29.8	29.9	31.2	36.4	38.7	27.3	29.4	29.5
LineGNN (FC)	29.5	30.9	31.0	32.5	37.4	39.5	27.5	30.4	30.4
LineGNN (LP)	28.4	30.1	30.2	31.9	36.5	38.5	29.6	32.3	32.3
LineGNN (GT)	32.2	32.7	32.8	37.2	41.5	42.4	31.6	33.1	33.1

Table 3: Ablation study on the effects of different link weights for LineGNN. "FC" denote the fully connected original graph, "LP" denotes the edges in original graph processed with predicted soft link weights, "GT" denotes the edges in original graph filtered by link ground truth.

	Ngc Recall			*mean Recall*		
l	R@20	R@50	R@100	mR@20	mR@50	mR@100
1	72.30	89.91	95.11	64.17	66.82	67.14
2	73.11	**90.59**	**95.62**	64.94	68.41	68.78
3	73.40	90.33	95.20	62.98	66.89	67.03
4	73.27	90.22	95.54	63.79	67.03	67.39
5	73.30	90.15	95.14	**65.80**	**68.49**	**68.92**
6	**73.48**	**90.59**	95.50	65.22	68.09	68.56
7	73.28	90.57	95.40	63.96	67.30	67.68

Table 4: Ablation on layer numbers l of LineGNN.

	Ngc Recall			*mean Recall*		
-	R@20	R@50	R@100	mR@20	mR@50	mR@100
Pre	**73.30**	90.15	95.14	**65.80**	**68.49**	**68.92**
Post	72.58	90.07	94.91	61.60	66.52	66.53
Mix	72.57	89.98	95.22	62.22	67.06	67.18
Para	72.34	**90.35**	**95.69**	62.43	67.11	67.17

Table 5: Ablation on LineGNN integration strategies.

Ablation Study

Effect of Link Prediction for LineGNN. Table 3 reports the effects of different link weights in the original graph on LineGNN. We consider three settings: using a fully connected link weights (FC), applying a link prediction module to obtain confident link weights (LP), and using ground-truth as link weights (GT). Compared to FC, LP consistently improves performance across both PredCls and SGCls, especially on ngcR@k and mR@k, showing that suppressing noisy edges before the line graph transformation benefits edge-centric relational reasoning. The GT variant yields the best results, serving as a reference to illustrate the performance upper bound under ideal link guidance. Note that the GT setting is included solely for analysis and is not considered part of our proposed method. These results confirm that the effectiveness of LineGNN relies significantly on the quality of links in the original graph, and that link prediction provides a practical solution to enhance relational context without requiring additional annotations.

Ablation Study on LineGNN Depth. We evaluate the impact of LineGNN depth by varying the number of layers from 1 to 7. As shown in Table 4. The results show that the best performance is achieved with 5 layers in terms of mean Recall (mR@20: 65.80, mR@50: 68.49, mR@100: 68.92). Based on these results, 5 layers is adopted as the default setting for LineGNN.

Ablation on LineGNN Integration Strategies. We evaluate four strategies for integrating LineGNN into the object-centric framework: inserting it before the object-centric GNNs (**Pre**), after (**Post**), mixing the object-centric GNN with LineGNN (**Mix**), and concatenating outputs from parallel branches (**Para**). As shown in Table 5, the **Pre** strategy consistently outperforms the others, achieving the highest ngcR@20 (73.30) and mR@100 (68.92), indicating that injecting relational context early benefits subsequent object-centric reasoning. In contrast, **Post** performs the worst, especially in mR@20 (61.60), suggesting that reasoning over objects first may limit the ability to capture relational dependencies. **Mix** and **Para** yield moderate performance, but still fall behind **Pre**. These results demonstrate that early integration of LineGNN is the most effective strategy for enhancing 3D scene graph prediction.

Conclusion

In this paper, we presented LEO, a unified framework for 3D scene graph prediction that performs sequential reasoning from edge-centric to object-centric paradigms. To address the limitations of conventional object-centric GNNs in capturing rich inter-relation context, we first construct a line graph where each relation is represented as a node, and apply edge-centric relational reasoning using a Line Graph Neural Network. The enhanced relation features are then integrated back into the original scene graph to facilitate object-aware reasoning. Extensive experiments on the 3DSSG dataset demonstrate that LEO consistently improves the performance of two strong baselines, validating the effectiveness of modeling relation-level dependencies through our edge-to-object reasoning paradigm.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (No. U20A20185, 62372491), the Guangdong Basic and Applied Basic Research Foundation (2022B1515020103, 2023B1515120087), the Shenzhen Science and Technology Program (No. RCYX20200714114641140). This work was also supported by the Oversea Study Program of Guangzhou Elite Project. Part of this work was conducted at the Computer Vision Group, University of Amsterdam, whose support is gratefully acknowledged.

References

Achlioptas, P.; Abdelreheem, A.; Xia, F.; Elhoseiny, M.; and Guibas, L. 2020. Referit3d: Neural listeners for fine-grained 3D object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 422–440. Springer.

Ao, S.; Guo, Y.; Hu, Q.; Yang, B.; Markham, A.; and Chen, Z. 2022. You only train once: Learning general and distinctive 3D local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3949–3967.

Azuma, D.; Miyanishi, T.; Kurita, S.; and Kawanabe, M. 2022. Scanqa: 3D question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19129–19139.

Betzel, R. F.; Faskowitz, J.; and Sporns, O. 2023. Living on the edge: network neuroscience beyond nodes. *Trends in cognitive sciences*, 27(11): 1068–1084.

Buterez, D.; Janet, J. P.; Oglic, D.; and Liò, P. 2025. An end-to-end attention-based approach for learning on graphs. *Nature Communications*, 16(1): 5244.

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

Chen, T.; Yu, W.; Chen, R.; and Lin, L. 2019. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6163–6171.

Chen, Z.; Li, X.; and Bruna, J. 2017. Supervised community detection with line graph neural networks. *arXiv preprint arXiv:1705.08415*.

Dang, J.; Zheng, H.; Lai, J.; Yan, X.; and Guo, Y. 2023. Efficient and robust video object segmentation through isogenous memory sampling and frame relation mining. *IEEE Transactions on Image Processing*, 32: 3924–3938.

Duan, J.; Yu, S.; Tan, H. L.; Zhu, H.; and Tan, C. 2022. A survey of embodied AI: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2): 230–244.

Feng, M.; Hou, H.; Zhang, L.; Wu, Z.; Guo, Y.; and Mian, A. 2023. 3D spatial multimodal knowledge accumulation for scene graph prediction in point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9182–9191.

Gross, J. L.; Yellen, J.; and Anderson, M. 2018. *Graph theory and its applications*. Chapman and Hall/CRC.

Liu, H.; Guo, Y.; Ma, Y.; Lei, Y.; and Wen, G. 2021. Semantic context encoding for accurate 3D point cloud segmentation. *IEEE Transactions on Multimedia (TMM)*, 23: 2045–2055.

Liu, H.; Ma, Y.; Hu, Q.; and Guo, Y. 2023a. CenterTube: Tracking multiple 3D objects with 4D tubelets in dynamic point clouds. *IEEE Transactions on Multimedia*, 25: 8793–8804.

Liu, H.; Ma, Y.; Wang, H.; Zhang, C.; and Guo, Y. 2023b. AnchorPoint: Query design for transformer-based 3D object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 24(10): 10988–11000.

Lv, C.; Qi, M.; Li, X.; Yang, Z.; and Ma, H. 2024. Sgformer: Semantic graph transformer for point cloud-based 3D scene graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4035–4043.

Ma, Y.; Guo, Y.; Liu, H.; Lei, Y.; and Wen, G. 2020. Global context reasoning for semantic segmentation of 3D point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2931–2940.

Ma, Y.; Liu, H.; Pei, Y.; and Guo, Y. 2025. Heterogeneous graph learning for scene graph prediction in 3D point clouds. In *European Conference on Computer Vision*, 274–291. Springer.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 652–660.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 8748–8763. PMLR.

Sharifzadeh, S.; Baharlou, S. M.; and Tresp, V. 2021. Classification by attention: Scene graph classification with prior knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 5025–5033.

Wald, J.; Dhamo, H.; Navab, N.; and Tombari, F. 2020. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3961–3970.

Wang, X.; Chen, C.; Min, Y.; He, J.; Yang, B.; and Zhang, Y. 2018. Efficient metropolitan traffic prediction based on graph recurrent neural network. *arXiv preprint arXiv:1811.00740*.

Wang, Z.; Cheng, B.; Zhao, L.; Xu, D.; Tang, Y.; and Sheng, L. 2023. Vl-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*, 21560–21569.

Wu, S.-C.; Wald, J.; Tateno, K.; Navab, N.; and Tombari, F. 2021. Scenegraphfusion: Incremental 3D scene graph

prediction from RGB-D sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7515–7525.

Yan, X.; Yuan, Z.; Du, Y.; Liao, Y.; Guo, Y.; Li, Z.; and Cui, S. 2021. CLEVR3D: Compositional language and elementary visual reasoning for question answering in 3D real-world scenes. *arXiv preprint arXiv:2112.11691*, 2(3).

Yang, Y.; Jia, B.; Zhi, P.; and Huang, S. 2024. Physcene: Physically interactable 3D scene synthesis for embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16262–16272.

Zhang, C.; Yu, J.; Song, Y.; and Cai, W. 2021a. Exploiting edge-oriented reasoning for 3D point-based scene graph analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9705–9715.

Zhang, S.; Hao, A.; Qin, H.; et al. 2021b. Knowledge-inspired 3D scene graph prediction in point cloud. *Proceedings of the Advances in Neural Information Processing Systems (NeruIPS)*, 34: 18620–18632.

Zhao, L.; Cai, D.; Zhang, J.; Sheng, L.; Xu, D.; Zheng, R.; Zhao, Y.; Wang, L.; and Fan, X. 2022. Toward explainable 3D grounded visual question answering: A new benchmark and strong baseline. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6): 2935–2949.