Heterogeneous Graph Learning for Scene Graph Prediction in 3D Point Clouds

Yanni Ma[1], Hao Liu[2], Yun Pei[1], and Yulan Guo[1*]

[1] The Shenzhen Campus of Sun Yat-Sen University, Sun Yat-Sen University,
[2] Nanyang Technological University

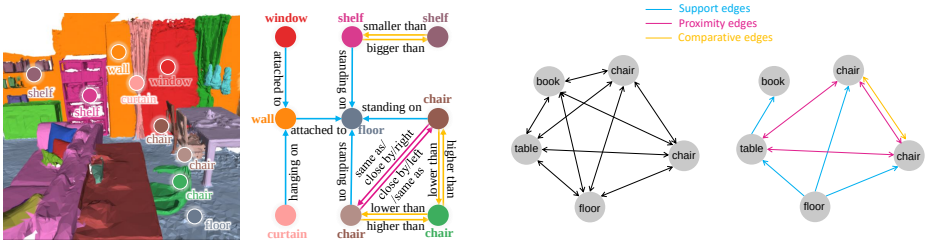
Abstract. 3D Scene Graph Prediction (SGP) aims to recognize the objects and predict their semantic and spatial relationships in a 3D scene. Existing methods either exploit context information or emphasize knowledge prior to model the scene graph in a fully-connected homogeneous graph framework. However, these methods may lead to indiscriminate message passing among graph nodes (i.e., objects), resulting in suboptimal performance. In this paper, we propose a 3D **Het**erogeneous **S**cene **G**raph **P**rediction (3D-HetSGP) framework, which performs graph reasoning on the 3D scene graph in a heterogeneous fashion. Specifically, our method consists of two stages: a heterogeneous graph structure learning (HGSL) stage and a heterogeneous graph reasoning (HGR) stage. In the HGSL stage, we learn the graph structure by predicting the types of different directed edges. In the HGR stage, message passing among nodes is performed on the learned graph structure for scene graph prediction. Extensive experiments show that our method achieves comparable or superior performance to existing methods on 3DSSG dataset.

Keywords: 3D scene understanding · 3D scene graph prediction · Heterogeneous graph

1 Introduction

3D Scene Graph Prediction (SGP) in point clouds has become an emerging research topic in 3D scene understanding, with broad applications including VR/AR [24], robotic navigation [19], and autonomous driving. Different from common tasks of 3D scene understanding such as 3D semantic segmentation [4, 9, 13, 15, 16] and object detection [10, 11, 35, 42], 3D SGP aims to recognize the semantic categories of objects and predict their semantic and spatial relationships. It typically constructs a directed scene graph whose nodes and edges represent objects and the relationships between connected objects.

Although remarkable progress has been made in recent years, 3D SGP remains highly challenging as 1) 3D point cloud data is typically sparse and irregular in spatial dimension. In particular, the appearance information (e.g., RGB) is no longer available, which makes it hard to capture the visual pattern. 2) The relationships between objects are intricate and diverse. According to the definition of 3DSSG [28], the predicates of these relationships are divided into three



(a) An example of 3DSSG [28]. **(b)** Homogeneous vs. heterogeneous graph.

Fig. 1: (a) Three different types of relationships constitute the complex 3D scene graph. The colors of edges indicate the predicate super-categories: *support*, *proximity*, and *comparative*. (b) Message passing on homogeneous and heterogeneous graphs.

distinct super-categories: *support*, *proximity*, *comparative* (as shown in Table 1). Relationship *support* implies the semantic support of an object to a subject, which is a kind of one-way relationship. Relationship *proximity* and *comparative* are both bilateral and multi-class relations. These three completely different types of relationships constitute the complex 3D scene graph. As shown in Fig. 1a, there are multiple relationships among *chairs* such as *same as*, *close by*, *left/right*, *lower/higher than*. Furthermore, these relationships also suffer from the long-tail class distribution issue. The above factors make fine-grained classification of relationships challenging.

To tackle these problems, most existing methods either exploit contextual information [28, 31, 39] or incorporate prior knowledge [3, 30, 41] to reduce prediction bias. Although these methods achieve promising performance, they still cannot obtain satisfactory results with fine-grained classification of multiple and long-tailed relationships due to the message passing process: (1) The traditional message passing mechanism treats all intricate relations equally, making it difficult to fully capture rich structural and semantic information. (2) The message passing is performed indiscriminately on fully-connected graph, which can lead to low-discriminative features after multiple iterations due to the accumulation of redundancy and bias.

To this end, we have re-analyzed the graph structure and the message passing process (see Figure 1b). We found that the 3D scene graph in existing methods is regarded as a fully-connected homogeneous graph, consisting of only single-type nodes and edges. However, there are three different types of edges (predicates) defined in a 3D scene graph. Performing message passing on a homogeneous graph without distinguishing different types of relationships is not an effective solution. Therefore, we employ a more suitable graph data structure (i.e., heterogeneous graph) to represent the 3D scene graph. Different from a homogeneous graph, a heterogeneous graph is allowed to have more than one types of nodes or edges, which is good at embedding rich semantic and structural information. First, the model can learn one type of relationship independently without being affected by irrelevant types of relationships, reducing the complexity of

Table 1: Three types of relationships defined in the 3DSSG dataset [28].

Types	Example Categories	Num
Support	supported by, attached to, hanging on, ...	14
Proximity	left, right, front, behind, close by, inside	6
Comparative	bigger than, higher than, same symmetry as,...	6

the relationship. Second, during the graph message passing stage, features are propagated discriminately and reasonably on different types of edges without incoming features from irrelevant neighbors, reducing semantic confusion and redundancy of features.

Motivated by this, we propose a 3D heterogeneous scene graph prediction (3D-HetSGP) framework based on the heterogeneous graph neural network. We formulate the 3D scene graph as a heterogeneous graph, in which the edges are treated differently according to their types: *Support*, *Proximity*, *Comparative*. We conduct graph message passing on high-confidence edges of different types. First, we propose a Heterogeneous Graph Structure Learning (HGSL) method to predict the type confidence score as edge weights to build a heterogeneous graph. We divide the graph into three types of sub-graphs with different type edges instead of fully-connected. Then, we propose a Heterogeneous Graph Reasoning (HGR) network to perform type-weighted message passing on the heterogeneous graph, in order to avoid redundant and confusing message passing during the graph reasoning process. Finally, to reduce the difficulty of classification, we utilize hierarchical classifiers. The 26 categories of predicates are divided into 3 types. The types of predicates are classified at first and then the categories of each type are classified separately.

The main contributions of this work are summarized as follows: (1) We are the first to formulate 3D scene graph as a heterogeneous graph, which is more naturally consistent with human understanding to the real world. (2) We propose a heterogeneous graph structure learning method to construct the heterogeneous graph by learning the type edges among objects. (3) We propose a heterogeneous graph reasoning network, which reduces redundant and confusing accumulation through reasoning on heterogeneous graph. (4) Extensive experiments on the 3DSSG dataset illustrate that 3D SGP can achieve superior performance by leveraging the heterogeneity of graph.

2 Related Work

2.1 2D Image Scene Graph Generation

The concept of scene graph is first mentioned in image retrieval [6]. With the development of large-scale scene graph datasets like Visual Genome [7], plenty of research works are proposed [8, 25, 26, 32, 34, 38], achieving excellent image-based scene graph generation performance. There are two different ways of SGG: two-stage and one-stage paradigms. Traditional methods [25, 26, 32, 34, 38] follow the

two-stage paradigm that consists of object detection and relation prediction. In the object detection stage, they usually employ a detector such as Faster R-CNN [18] to detect objects. In the relation prediction stage, they focus on exploiting contextual information by a message passing neural network and collecting more external prior knowledge [37] to reduce the complexity of prediction. As for one-stage image-based SGG models, recent works [8, 22] were proposed to jointly optimize the object detection and relation prediction in an end-to-end network. Compared to 2D SGG, few works have been developed for its 3D counterpart, which provides more spatial geometric information to describe the 3D real world.

2.2 3D Point Cloud Scene Graph Prediction

Thanks to the release of the 3DSSG dataset [28], 3D SGP has attracted a lot of attention recently. Wald *et al.* [28] proposed the first baseline network for 3D scene graph prediction, SGPN, which is an end-to-end network based on Graph Neural Network (GNN). To improve the quality of prediction, existing research efforts have focused on either aggregating contextual information or incorporating prior knowledge into the network. Wu *et al.* [31] proposed SGFN, which introduces a feature-wise attention mechanism in GNN's message passing process to obtain more contextual information between objects and predicates. Zhang *et al.* [39] proposed SGGpoint, which introduces an EDGE-oriented Graph Convolution Network (GCN) and a twinning attention mechanism for graph-based reasoning. Lv *et al.* [12] proposed Semantic Graph Transformer (SGFormer), which introduces a Transformer-based framework to preserve global context. KISGP [41] introduced a graph auto-encoder network to embed prototype knowledge from class labels and integrated the learned prior knowledge embedding into 3D SGP models to improve the accuracy of object and predicate classification. KISGP is the first to evaluate 3D SGP models in PredCls and SGCls tasks [33]. Wang *et al.* [30] proposed a Visual-Linguistic Semantics Assisted Training (VL-SAT) scheme, which builds an oracle multi-modal model with CLIP [17] to alleviate the long-tailed problem and boost performance. However, existing methods lack consideration for the effective graph structure and message passing, classifying all relations equally and flatly. Such methods cannot perform directional message passing and satisfy the fine-grained classification on complex relationships.

2.3 Heterogeneous Graph Learning

Heterogeneous graph [23] is a flexible graph structure that can encompass different types of nodes and edges. There are several heterogeneous graph learning methods [5, 21, 29, 40, 43], which can encode rich semantic and structural information effectively. Since the scene graph may have multiple types of objects and relations, several works have attempted to treat the scene graph as a heterogeneous graph. SCENE [14] utilized a heterogeneous graph to reason about the dynamic agents and static infrastructure information of traffic scenes. HetSGG [36] was the first to consider the heterogeneity of a scene graph in the

SGG task. They assigned different types to objects and relations based on the prediction of objects to construct a heterogeneous graph and performed message passing on the graph to capture different contexts of each relation according to its type. However, there is no research considering the heterogeneity of the 3D scene graph in the 3D SGP task. 3D scene graph in point clouds has far more complicated relations involving not only semantic relations but also spatial relations, and geometric comparative relations. It is necessary to treat a 3D scene graph in a heterogeneous way, to reason and infer different types of relations separately.

3 Methods

3.1 Overview

A 3D scene graph is a graphical description of a given 3D scene. Given the point clouds of 3D scene data $\mathbf{D} \in \mathbb{R}^{N\times 3}$ with N 3D points and K class-agnostic instances masks $\mathcal{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_K\}$ as input, the goal of the 3D scene graph prediction task is to predict the objects $O = \{o_1, \ldots, o_K\}$ as graph nodes and relation predicates $R = \{r_{ij}\}$ between object pairs as graph edges.

In this paper, unlike previous homogeneous graph approaches, we formulate the 3D scene graph as a heterogeneous graph and solve the task as a heterogeneous graph learning problem. We use a directed heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}_\mathcal{E})$ to represent our 3D scene graph, where the nodes $\mathcal{V}$ consist of the objects O, the edges $\mathcal{E}$ consist of the predicates of relations, and the types $\mathcal{T}_\mathcal{E}$ consist of types of predicates.

We propose a novel framework named **3D Het**erogeneous **S**cene **G**raph **P**rediction (**3D-HetSGP**). Our framework consists of two stages. The first stage is heterogeneous graph structure learning, which aims to learn the graph structure by predicting the different types of edges. The second stage is heterogeneous graph reasoning, which aims to learn the scene graph by performing graph message passing effectively on the predicted graph structure. Moreover, we update the graph structure by collecting the final prediction scores of the predicates output from the second stage as new type scores and retrain the framework. The training objective for mapping a 3D scene from point clouds to a scene graph is to maximize the joint probability:

$$P(\mathcal{G}, \mathcal{T}|D) = P(\mathcal{T}|D)P(\mathcal{G}|D, \mathcal{T}), \tag{1}$$

where $P(\mathcal{T}|D)$ represents the prediction of heterogeneous graph structure learning stage, and $P(\mathcal{G}|D, \mathcal{T})$ represents the prediction of heterogeneous graph reasoning stage. The overview of our framework is illustrated in Fig. 2.

3.2 Heterogeneous Graph Structure Learning

This stage aims to learn a heterogeneous graph structure of the scene graph by predicting the type scores of predicates. According to the definition in 3DSSG

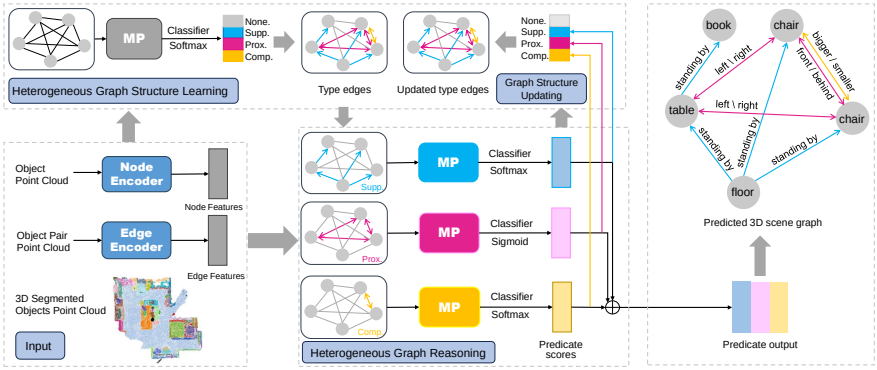


Fig. 2: Our 3D heterogeneous scene graph prediction(3D-HetSGP) framework. It consists of two stage: (a) The HGSL stage: the graph structure is learned by predicting the types of different edges. (b) The HGR stage: message passing among nodes is performed on the learned graph structure. Finally, the prediction scores are fused to generate a 3D scene graph.

[28], we divide the 26 predicate labels into 3 types of super-categories, as shown in Table 1. The pipeline at this stage is similar to recent works [41] [31]. First, we initialize a fully-connected and homogeneous scene graph with node and edge features extracted from scene point clouds. Then, we employ a graph neural network for message passing on this graph to learn node and edge features. Finally, we classify edge features into three super-category types and one 'none' category (indicating that no relationship exists on the edge). We use the type prediction scores for each edge as its weights to construct a heterogeneous graph.

3D Scene Graph Initialization We first initialize the scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a fully-connected and homogeneous graph structure, which means all nodes are connected to each other in the same way. We filter the 3D scene point clouds using class-agnostic instance masks and obtain the object point sets. Following KISGP [41], we employ the multi-scale PointNet [15] on the object point sets to obtain the object node features $\boldsymbol{f}_n$. The object-pair (predicate) edge features $\boldsymbol{f}_e$ are initialized based on the features of subjects and objects.

Message Passing Once the initial node features and edge features are obtained, we employ the graph neural network in KISGP [41] to propagate and aggregate the contextual information of nodes and edges on the initial graph through message passing. For message updating, the gated recurrent units (GRUs) are used to update the hidden state $\boldsymbol{h}_i^{n,l}$ for node $\boldsymbol{v}_i$ and $\boldsymbol{h}_{ij}^{e,l}$ for edge $(\boldsymbol{v}_i, \boldsymbol{v}_j)$ in layer l:

$$\boldsymbol{h}_i^{n,l+1} = \mathrm{GRU}(\boldsymbol{h}_i^{n,l}, \boldsymbol{m}_i^{n,l}), \boldsymbol{h}_{ij}^{e,l+1} = \mathrm{GRU}(\boldsymbol{h}_{ij}^{e,l}, \boldsymbol{m}_{ij}^{e,l}), \tag{2}$$

where $\boldsymbol{m}_i^{n,l}$ and $\boldsymbol{m}_{ij}^{e,l}$ are the incoming messages from neighbor nodes and edges for updating. The incoming messages $\boldsymbol{m}_i^{n,l}$ of nodes and $\boldsymbol{m}_{ij}^{e,l}$ of edges are:

$$\begin{aligned}\boldsymbol{m}_i^{n,l} &= \text{mean}(\sum_{j \in N_{i*}} \boldsymbol{m}_{ij}^{s,l} + \sum_{j \in N_{*i}} \boldsymbol{m}_{ji}^{o,l}) \\ &= \text{mean}(\text{LN}(\phi_n(\boldsymbol{h}_i^{n,l}) + \phi_e(\boldsymbol{h}_{ij}^{e,l})) + \text{LN}(\phi_n(\boldsymbol{h}_i^{n,l}) + \phi_e(\boldsymbol{h}_{ji}^{e,l}))), \end{aligned} \tag{3}$$

$$\boldsymbol{m}_{ij}^{e,l} = \text{LN}(\phi_n(\boldsymbol{h}_i^{n,l}) + \phi_n(\boldsymbol{h}_j^{n,l})). \tag{4}$$

where ϕ_n and ϕ_e are the non-linear transformations for nodes and edges, LN represents layer normalization, mean$(\cdot)$ is a mean operation. The graph neural network has l layers for updating node and edge features.

Type Prediction Head The edge features $\boldsymbol{f}_e$ obtained through message passing are then classified into four classes, including three predicate types and the 'none' category. The 4-dimensional prediction scores $\boldsymbol{TS}$ of predicates type are calculated as:

$$\boldsymbol{TS} = \text{softmax}(\text{MLP}(\boldsymbol{f}_e)), \tag{5}$$

Each dimension $\boldsymbol{s}_k$ of $\boldsymbol{TS}$ is one type weight of the four categories. We then obtain the final type edges:

$$\boldsymbol{w}_t = [(\boldsymbol{s}_t) > \alpha], t \in [1, 4]. \tag{6}$$

where $\boldsymbol{w}_1$ is the edge of 'none' class, $\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4$ are learned edges of types *support*, *proximity* and *comparative*, respectively. α is a hyper-parameter. We utilize these type edges as learned heterogeneous graph structure in the next stage.

3.3 Heterogeneous Graph Reasoning

This stage aims to predict a heterogeneous 3D scene graph by reasoning on the learned heterogeneous graph structure. First, we initialize a heterogeneous scene graph with node features and edge features initialized in the first stage and updated based on 3D spatial and geometric properties. Then, we perform heterogeneous graph message passing on our heterogeneous graph, which consists of three subgraphs. The edge features of each subgraph are fed into their corresponding type of classifier to obtain the predicate prediction score for each subgraph. Finally, we concatenate the object and predicate prediction scores of each subgraph as the final prediction result of the heterogeneous 3D scene graph.

Heterogeneous Graph Initialization In this stage, we initialize the scene graph as a heterogeneous graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}_\mathcal{E})$ with type edges, where $\mathcal{T}_\mathcal{E}$ denotes the set of predicate types. According to different type edges, we construct three different relation subgraphs separately: *Support* subgraph, *Proximity* subgraph, and *Comparative* subgraph.

To initialize the heterogeneous scene graph features $\mathcal{V}$ and $\mathcal{E}$, we take the node and edge features initialized from the first stage as input. Furthermore, to propagate more precise semantic information, we update the edge features of different subgraphs according to their characteristics. The spatial and geometric features are calculated and fused into the features of *Proximity* and *Comparative* subgraphs respectively. For spatial features, we calculate the offset, distance and direction between two instances in turn:

$$\boldsymbol{f}_{ij}^{spat} = \text{fc}(\text{Concat}(\boldsymbol{c}_i - \boldsymbol{c}_j, \|\boldsymbol{c}_i - \boldsymbol{c}_j\|_2, \frac{\boldsymbol{c}_i - \boldsymbol{c}_j}{\|\boldsymbol{c}_i - \boldsymbol{c}_j\|_2})), \tag{7}$$

where $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ are the centroids of 3D bounding boxes for objects i and j, respectively. For geometric features, we calculate the difference of length l, width w, height h and volume $V = lwh$ between two instances:

$$\boldsymbol{f}_{ij}^{geom} = \text{fc}(\text{Concat}(\log\frac{l_i}{l_j}, \log\frac{w_i}{w_j}, \log\frac{h_i}{h_j}, \log\frac{V_i}{V_j})), \tag{8}$$

Then, the spatial features $\boldsymbol{f}_{ij}^{spat}$ and the geometric features $\boldsymbol{f}_{ij}^{geom}$ are fused to update the initialized edge features $\boldsymbol{e}_{ij}$ of *Proximity* and *Comparative* subgraphs.

$$\boldsymbol{f}_{e_{ij}}^{update} = \text{MLP}(\text{Concat}(\boldsymbol{f}_{e_{ij}}, \text{MLP}(\boldsymbol{f}^{update}))). \tag{9}$$

Heterogeneous Graph Message Passing After initializing the heterogeneous scene graph consisting of three subgraphs, we introduce a heterogeneous graph neural network to propagate and aggregate the contextual information according to different relation types on the initialized subgraphs. The network has the same message updating procedure as the first stage. What differs is the incoming messages. We adapt GRU to update the hidden state $\boldsymbol{h}_i^{n,l,t}$ for node $\boldsymbol{v}_i$ and $\boldsymbol{h}_{ij}^{e,l,t}$ for edge $(\boldsymbol{v}_i, \boldsymbol{v}_j)$ in every layer l for each subgraph type t in the same way as Eq. (2). We cascade multiple layers of graph convolutions to aggregate information. To learn effective and clean incoming messages for nodes and edges, we only aggregate information from the same type of edges on each subgraph.

a) Incoming messages of edge relations. The edge incoming message is generated from nodes of connected neighbors, which are determined by the type edges $\boldsymbol{w}_t$. For each type of subgraph t, given two node objects n_i and n_j on either side of edge e_{ij}, the edge incoming message $\boldsymbol{m}_{ij}^{e,l,t}$ at l-th layer is computed as:

$$\boldsymbol{m}_{ij}^{e,l,t} = \text{LN}(\boldsymbol{w}_t\phi_n(\boldsymbol{h}_i^{n,l,t}) + \boldsymbol{w}_t\phi_n(\boldsymbol{h}_j^{n,l,t})), \tag{10}$$

where $\boldsymbol{h}_i^{n,l,t}$ and $\boldsymbol{h}_j^{n,l,t}$ are hidden states of two object features, $\phi_n(\cdot)$ is the non-linear function for nodes. The type edges $\boldsymbol{w}_t$ determine the propagation path of information.

b) Incoming message of node objects. After the edge hidden state is updated by GRU, the incoming message of node is aggregated from all the neighboring edges connected to that node based on the type edge $\boldsymbol{w}_t$. For each

type subgraph t, given all the neighboring edges e_{ij} connected with node n_i, the incoming message $\boldsymbol{m}_i^{n,l,t}$ for the node at the (l)-th layer is calculated as:

$$\boldsymbol{m}_i^{n,l,t} = \sum_{j \in N_{i*}} \boldsymbol{w}_t \phi_e(\boldsymbol{h}_{ij}^{e,l,t}). \tag{11}$$

where N_{i*} denotes the neighboring nodes of node i, $\boldsymbol{h}_{ij}^{e,l,t}$ is the hidden state of edge feature, $\phi_e(\cdot)$ is the non-linear function for edges.

Prediction Head To generate the final prediction 3D scene graph, the final node features $\boldsymbol{f}_n^t$ and edge features $\boldsymbol{f}_e^t$ obtained through heterogeneous message passing are classified into objects and predicates categories by predictors. The network finally outputs three node and edge features, one for each predicate type. The predicate confidence score vectors $\boldsymbol{s}_e^t$ of each type t are computed as:

$$\boldsymbol{s}_e^t = \sigma(\text{MLP}(\boldsymbol{f}_e^t)), t \in \{supp, prox, comp\}, \tag{12}$$

$\sigma(\cdot)$ is an activation function, we use sigmoid$(\cdot)$ for type *proximity*. The final predicate confidence score $\boldsymbol{s}_e$ is computed as:

$$\boldsymbol{s}_e = \max(\text{Concat}(\boldsymbol{s}^{supp}, \boldsymbol{s}^{prox}, \boldsymbol{s}^{comp})), \tag{13}$$

The object confidence score $\boldsymbol{s}_n$ is computed as:

$$\boldsymbol{s}_n = \text{softmax}(\text{MLP}(\text{Concat}(\boldsymbol{f}_n^{supp}, \boldsymbol{f}_n^{prox}, \boldsymbol{f}_n^{comp}))). \tag{14}$$

3.4 Heterogeneous Graph Structure Updating

To improve the accuracy between the predicted graph structure with the actual graph structure, we propose a graph structure updating operation to update the type edges $\boldsymbol{w}_t$ in Eq. (10) and (11). Specifically, $\boldsymbol{w}_t$ is updated by summing the predicate confidence scores produced by HGR separately as: $\boldsymbol{w}_t = [(\sum_i^{d^t} \boldsymbol{s}_{ei}^t) > \alpha], t \in \{supp, prox, comp\}$, where $\boldsymbol{s}_{ei}^t$ is the confidence score of the i-th sub-category, d^t is the number of sub-categories. After updating the type edges $\boldsymbol{w}_t$, we retrain the HGR to improve performance. An ablation study is provided in Sec. 4.4.

3.5 Training Objective

Our scene graph prediction consists of object classification, type classification and predicate classification. Therefore the total loss $\mathcal{L}_{total}$ is formulated as:

$$\begin{aligned} \mathcal{L}_{total} &= \lambda \mathcal{L}_{obj} + \mathcal{L}_{type} + \mathcal{L}_{pred} \\ &= \lambda \mathcal{L}_{obj} + \mathcal{L}_{type} + \mathcal{L}_{supp} + \mathcal{L}_{prox} + \mathcal{L}_{comp}. \end{aligned} \tag{15}$$

where λ is a weight to balance the loss of objects and predicates. $\mathcal{L}_{supp}$, $\mathcal{L}_{prox}$ and $\mathcal{L}_{comp}$ are the classification losses for the types *support*, *proximity* and *comparative* respectively. Same as KISGP [41], all these losses are defined as focal loss, except for the *proximity* type loss. Since the relationships of *proximity* have multiple classes, we utilize the binary cross-entropy loss for $\mathcal{L}_{prox}$.

4 Experiments

4.1 Experimental Settings

3DSSG Dataset The 3DSSG dataset [28] is an extension of 3RScan [27], providing annotations for 3D semantic scene graphs within the 3RScan dataset. It includes 1,482 3D reconstructed models of 478 indoor environments. The 1,482 scene graphs have a total of 48k object nodes and 544k edges. For a fair comparison, we split the 1,482 scenes into 3852 sub-scenes for the training set and 548 for the test set in the same way as KISGP [41]. Each scene has 9 object nodes on average. Following RIO27 annotation [2], we utilize 160 object categories and 27 predicate classes, including 'none' relation, in our experiments.

Metrics and Task Description We evaluate our model in two standard tasks of scene graph prediction [33]. (1) The predicate classification (PredCls) aims to evaluate the performance of predicate prediction between every object pair, given the ground truth labels of objects. (2) The scene graph classification (SGCls) aims to evaluate the overall performance of scene graph prediction, including predictions of both objects and predicates. We use top-k (R@k) recall, no graph constraint top-k (ngcR@k) recall, and mean recall (mR@k) as the metrics for these two tasks. The top-k(R@K) recall measures the proportion of correct relationships among the top k highest confidence scores. The ngcR@k allows each edge to have multiple predicates without graph constraint to only one relationship with the highest score. Due to the long-tail distribution in the dataset, we also compute the mean recalls (mR@k) for each predicate.

Implementation Details We follow the same experiment configuration and procedure as in KISGP [41]. Our model is implemented in PyTorch. We train our model on an NVIDIA GTX TITAN GPU for 40 epochs using the ADAM optimizer. We set the initial learning rate to 0.0001, and the weight decay is set to 0.7 every ten epochs. Following KISGP, we also pretrain the multi-scale PointNet [15] on the 3DSSG dataset and utilize the pretrained PointNets to encode the point cloud into initial object and predicate features for the training of scene graph prediction model. The GNN modules are cascaded for 3 layers in the heterogeneous graph structure learning stage and 5 layers in the heterogeneous graph reasoning stage. For graph structure updating, we collect predicate score results and compute type weights after the first 40 epochs. Subsequently, we replace the type edges in the heterogeneous graph with the updated edges and train it for another 40 epochs. We set $\lambda = 0.1$, and the focal loss is the same as SGPN [28].

4.2 Comparison with Related Methods

Since our model is developed based on KISGP [41], we evaluate our model in PredCls/SGCls tasks against both KISGP and several reference methods, including Co-Occurrence [41], KERN [1], Schemata [20], SGPN [28]. For a fair

Table 2: Quantitative results of the evaluated methods in PredCls tasks. The model with * represents the results we have reproduced.

Model	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
Co-Occurrence [41]	0.347	0.474	0.479	0.351	0.556	0.706	0.338	0.474	0.479
KERN [1]	0.468	0.557	0.565	0.483	0.648	0.772	0.188	0.256	0.265
Schemata [20]	0.487	0.582	0.591	0.496	0.671	0.802	0.352	0.426	0.433
SGPN [28]	0.519	0.580	0.585	0.545	0.701	0.824	0.321	0.384	0.389
SGFN* [31]	0.545	0.610	0.615	0.614	0.801	0.900	0.305	0.364	0.366
KISGP [41]	0.593	0.650	0.653	0.622	0.784	0.883	0.566	0.635	0.638
KISGP* [41]	0.599	0.651	0.654	0.631	0.791	0.886	0.571	0.619	0.620
Ours	**0.618**	**0.659**	**0.659**	**0.703**	**0.889**	**0.946**	**0.637**	**0.681**	**0.682**

comparison, we reproduce the prediction results of SGFN and KISGP models under the same data and experimental environment. The evaluation code of KISGP is utilized to reproduce the top-k recall of the PredCls and SGCls tasks. We only reproduce and compare with the results of models with open-sourced code.

Table 3: Quantitative results of the evaluated methods in SGCls tasks. The model with * represents the results we have reproduced.

Model	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
Co-Occurrence [41]	0.148	0.197	0.199	0.141	0.202	0.258	0.088	0.127	0.129
KERN [1]	0.203	0.224	0.227	0.208	0.247	0.276	0.095	0.115	0.119
Schemata [20]	0.274	0.292	0.294	0.288	0.335	0.363	0.238	0.270	0.272
SGPN [28]	0.270	0.288	0.290	0.282	0.326	0.353	0.197	0.226	0.231
SGFN* [31]	0.272	0.289	0.289	0.300	0.346	0.369	0.182	0.208	0.209
KISGP [41]	0.285	0.300	0.301	0.298	0.343	0.370	0.244	0.286	0.288
KISGP* [41]	**0.288**	**0.305**	**0.306**	0.303	0.347	0.376	0.251	0.281	0.283
Ours	0.285	0.298	0.299	**0.312**	**0.364**	**0.387**	**0.273**	**0.294**	**0.295**

Quantitative Results Tables 2, 3 report the quantitative results for the PredCls and SGCls tasks. Compared to the baseline model KISGP [41], our method achieves a significant performance improvement. For example, in the PredCls task on ngcR@K, our model improves around 7%, 10% and 6% at ngcR@20, ngcR@50 and ngcR@100 than KISGP, respectively. This suggests that leveraging a heterogeneous graph can enhance the classification of multiple relationships. Moreover, in the SGCls task, we achieve around 1.7% improvement over KISGP at ngcR@50, indicating that reasoning within a heterogeneous graph is capable of enhancing the message passing among nodes and edges in an entire scene graph. As shown in the mean recall part of the table, our model outperforms KISGP by 1.3% and 6.2% at mR@50 in SGCls and PredCls tasks, respectively. This demonstrates that our model alleviates the long-tail distribution issue without fusing prior knowledge into the models.

Table 4: Comparison results of no graph constraint Recall@20/50/100 for *Support, Proximity, Comparative* types.

Type	*Support*			*Proximity*			*Comparative*			Mean		
Methods	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100
SGFN	71.82	82.29	89.72	51.44	74.14	84.28	10.34	16.75	21.91	10.34	16.75	21.91
KISGP	**89.00**	**94.54**	**96.43**	41.88	61.85	73.94	20.12	24.14	26.32	50.33	60.18	65.56
Ours	88.88	93.22	94.93	**51.84**	**79.07**	**86.31**	**25.63**	**27.98**	**29.07**	**55.45**	**66.76**	**70.10**

Moreover, Table 4 shows the comparison results of our 3DHetSGP with KISGP [41] on per predicate type: *support*, *proximity*, and *comparative*, and their mean values. To evaluate the predicates including multi-relationship predicates, we report the results in the ngcR@k for the PredCls task. Compared to KISGP, our 3DHetSGP achieves significant improvements on the *proximity* and *comparative* types of predicates. Specifically, we obtain 9.96%, 17.22%, and 12.37% improvements for the *proximity* type on R@20, R@50, and R@100, respectively, although our 3DHetSGP drops slightly on the *support* type (which is already predicted quite accurately). The mean results across all types show that our method increases the overall performance by a significant margin. Therefore, it is effective to treat the 3D scene graph as a heterogeneous graph and predict predicates separately according to their belonging types.

Qualitative Results Figure 3 shows a predicted scene graph on 3DSSG [28]. In the top 20 predicted relationships, our predictions match all the ground truth relationships, especially for the multi-relationship scenarios. For the prediction of relation pair *chair-chair*, we capture not only the type *proximity* predicates 'right/left', 'close by', but also the type *comparative* predicate 'same as', which are not predicted by KISGP. These results demonstrate that the heterogeneous graph structure is indeed more suitable for learning of multiple relationships.

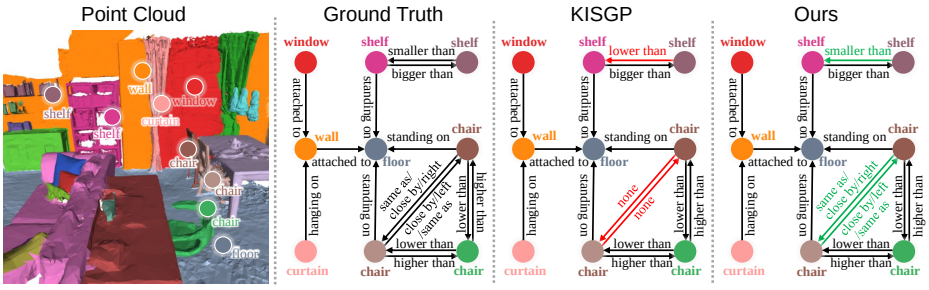


Fig. 3: Qualitative results between ours and KISGP [41] on no graph constraint Recall@20 for the PredCls task. Red arrows: incorrect predictions by KISGP, Green arrows: correct predictions by our model.

4.3 Analysis on Long-Tail Distribution

We investigate the impact of our model on the long-tail distribution problem of 3DSSG [28]. We sort all predicate classes in descending order based on their occurrence frequency and then separate them into three groups (i.e., head, body, tail) according to frequency. We report the R@k and mR@k metrics for each group in Table 5. Compared to KISGP, although we drop 0-1.5% on head classes, we obtain 10.82% on mR@20 for body and 8.23% on mR@50 for tail classes.

Table 5: Comparison results of Recall@20/50 and mean Recall@20/50 for the head, body, tail predicate classes in 3DSSG [28].

	Head		**Body**		**Tail**	
Methods	R@20/50	mR@20/50	R@20/50	mR@20/50	R@20 / 50	mR@20/50
SGFN	58.50/**65.17**	50.21/58.67	23.23/27.37	33.63/42.49	8.58/9.85	17.36/19.58
KISGP	**59.84**/65.00	**51.43/58.84**	49.13/52.61	69.66/76.46	20.50/20.74	61.87/62.52
Ours	59.53/64.01	50.39/57.40	**54.56/56.51**	**80.48/82.76**	**22.3/23.66**	**66.30/70.75**

We additionally visualize the comparative performance with KISGP for per-class predicates on mR@50. As shown in Fig. 4, our method outperforms KISGP on many predicates, especially on body and tail predicates, including *same as*, *same symmetry as*, *lying in*, and *cover*. The line chart indicates the occurrence frequency ratio for each predicate in the test set.

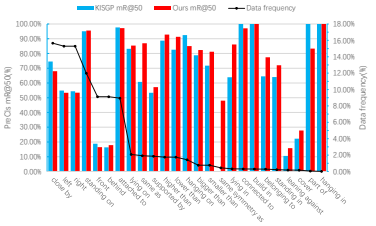


Fig. 4: The bar chart shows the results on mR@50 for each predicate category of KISGP [41] and our 3D-HetSGP. The line chart shows the data frequency for predicate categories in 3DSSG [28].

4.4 Ablation Study

Heterogeneous Graph Reasoning To investigate the effectiveness of our heterogeneous graph reasoning, we report the ablation results of different graph structures and connection methods in Table 6. Note that, HomoGraph (KISGP) denotes the fully-connected homogeneous graph structure of KISGP [41]. HeterGraph denotes heterogeneous graph structure with different connection methods: **FC** (Fully-connected graph, i.e., without type edges), **Learned** (Learned type edges from HGSL for subsequent graph reasoning), **Updated** (Updated type edges after HGR), **GT** (Ground truth type edges). We find that reasoning on the fully-connected homogeneous graph drops the performance by a large margin. Furthermore, the HeterGraph (Learned) structure leads to sub-optimal performance compared to

HeterGraph (Updated), indicating that using a sufficiently accurate graph structure is crucial for the graph reasoning stage.

Table 6: Ablation results on heterogeneous graph reasoning.

Graph Structure	R@20	R@50	R@100	ngcR@20	ngcR@50	ngcR@100	mR@20	mR@50	mR@100
HomoGraph(KISGP)	0.599	0.651	0.654	0.631	0.791	0.886	0.571	0.619	0.620
HeterGraph(FC)	0.607	0.654	0.655	0.641	0.799	0.889	0.574	0.665	0.670
HeterGraph(Learned)	0.604	0.654	0.655	0.660	0.811	0.892	0.623	0.667	0.668
HeterGraph(Updated)	**0.618**	**0.659**	**0.659**	**0.703**	**0.889**	**0.946**	**0.637**	**0.681**	**0.682**
HeterGraph(GT)	0.665	0.684	0.684	0.810	0.970	0.995	0.673	0.720	0.721

Heterogeneous Graph Structure Learning We further investigate the accuracy of the learned type edges and the final updated type edges. As shown in Table 7, the type edges updated from final predicate scores are more precise than the prediction of HGSL. However, it does not mean that we have to abandon HGSL. As shown in Table 6, our model is iteratively updated to achieve optimal scene graph predictions step-by-step.

Table 7: Ablation study on heterogeneous graph structure learning. Note that, type-acc denotes the accuracy of predicted type edges among existing type edges, edge-acc denotes the accuracy of edges among all objects in a scene.

	Support		*Proximity*		*Comparative*	
Type edge	type-acc	edge-acc	type-acc	edge-acc	type-acc	edge-acc
Learned	0.93	0.87	0.83	0.80	0.91	0.87
Updated (Ours)	**0.97**	**0.96**	**0.91**	**0.89**	**0.96**	**0.96**

5 Conclusion

In this paper, we proposed a novel framework termed 3D-Heterogeneous Scene Graph Prediction (3D-HetSGP) to learn the complex 3D scene graph in a heterogeneous manner. Our framework first learns the edge weights of different types to construct a heterogeneous graph, and then performs heterogeneous message passing on the graph to avoid semantic confusion and bias accumulation. Extensive experimentation on the 3DSSG dataset demonstrates that our method achieves comparable or superior performance to KISGP. In the future, we plan to investigate the heterogeneity and hierarchy of 3D objects to minimize the gap between machine and human brain comprehension of the real world as much as possible.

Acknowledgement.
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).

References

1. Chen, T., Yu, W., Chen, R., Lin, L.: Knowledge-embedded routing network for scene graph generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). p. 6163–6171 (2019)
2. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS). vol. 27 (2014)
3. Feng, M., Hou, H., Zhang, L., Wu, Z., Guo, Y., Mian, A.: 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 (CVPR). pp. 9182–9191 (2023)
4. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11108–11117 (2020)
5. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: Proceedings of the web conference 2020. pp. 2704–2710 (2020)
6. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3668–3678 (2015)
7. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision (IJCV) **123**, 32–73 (2017)
8. Li, R., Zhang, S., He, X.: Sgtr: End-to-end scene graph generation with transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19486–19496 (2022)
9. Liu, H., Guo, Y., Ma, Y., Lei, Y., Wen, G.: Semantic context encoding for accurate 3D point cloud segmentation. IEEE Transactions on Multimedia (TMM) **23**, 2045–2055 (2021)
10. Liu, H., Ma, Y., Hu, Q., Guo, Y.: Centertube: Tracking multiple 3D objects with 4D tubelets in dynamic point clouds. IEEE Transactions on Multimedia (TMM) **25**, 8793–8804 (2023)
11. Liu, H., Ma, Y., Wang, H., Guo, Y.: Anchorpoint: Query design for transformer-based 3D object detection and tracking. IEEE Transactions on Intelligent Transportation Systems (TITS) **24**(10), 10988–11000 (2023)
12. Lv, C., Qi, M., Li, X., Yang, Z., Ma, H.: SGFormer: Semantic graph transformer for point cloud-based 3d scene graph generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 4035–4043 (2024)
13. Ma, Y., Guo, Y., Liu, H., Lei, Y., Wen, G.: Global context reasoning for semantic segmentation of 3D point clouds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2931–2940 (2020)

14. Monninger, T., Schmidt, J., Rupprecht, J., Raba, D., Jordan, J., Frank, D., Staab, S., Dietmayer, K.: Scene: Reasoning about traffic scenes using heterogeneous graph neural networks. IEEE Robotics and Automation Letters (RAL) **8**(3), 1531–1538 (2023)
15. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 652–660 (2017)
16. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS). pp. 5099–5108 (2017)
17. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 8748–8763. PMLR (2021)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) **28** (2015)
19. Rosinol, A., Gupta, A., Abate, M., Shi, J., Carlone, L.: 3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans. arXiv preprint arXiv:2002.06289 (2020)
20. Sahand, S., Sina, M.B., Volker, T.: Classification by attention: Scene graph classification with prior knowledge. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 35, pp. 5025–5033 (2021)
21. Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering (TKDE) **31**(2), 357–370 (2018)
22. Shit, S., Koner, R., Wittmann, B., Paetzold, J., Ezhov, I., Li, H., Pan, J., Sharifzadeh, S., Kaissis, G., Tresp, V., et al.: Relationformer: A unified framework for image-to-graph generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 422–439. Springer (2022)
23. Sun, Y., Han, J.: Mining heterogeneous information networks: a structural analysis approach. ACM Sigkdd Explorations Newsletter **14**(2), 20–28 (2013)
24. Tahara, T., Seno, T., Narita, G., Ishikawa, T.: Retargetable ar: Context-aware augmented reality in indoor scenes based on 3d scene graph. In: 2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). pp. 249–255 (2020)
25. Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H.: Unbiased scene graph generation from biased training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3716–3725 (2020)
26. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6619–6628 (2019)
27. Wald, J., Avetisyan, A., Navab, N., Tombari, F., Nießner, M.: Rio: 3d object instance relocalization in changing indoor environments. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7658–7667 (2019)
28. Wald, J., Dhamo, H., Navab, N., Tombari, F.: Learning 3D semantic scene graphs from 3D indoor reconstructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3961–3970 (2020)

29. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: Proceedings of the World Wide Web conference. pp. 2022–2032 (2019)
30. Wang, Z., Cheng, B., Zhao, L., Xu, D., Tang, Y., Sheng, L.: 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 (CVPR). pp. 21560–21569 (2023)
31. Wu, S.C., Wald, J., Tateno, K., Navab, N., Tombari, F.: Scenegraphfusion: Incremental 3D scene graph prediction from RGB-D sequences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7515–7525 (2021)
32. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5410–5419 (2017)
33. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5410–5419 (2017)
34. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph r-cnn for scene graph generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 670–685 (2018)
35. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3D object detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11784–11793 (2021)
36. Yoon, K., Kim, K., Moon, J., Park, C.: Unbiased heterogeneous scene graph generation with relation-aware message passing neural network. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 37, pp. 3285–3294 (2023)
37. Zareian, A., Karaman, S., Chang, S.F.: Bridging knowledge graphs to generate scene graphs. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 606–623. Springer (2020)
38. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5831–5840 (2018)
39. Zhang, C., Yu, J., Song, Y., Cai, W.: 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). pp. 9705–9715 (2021)
40. Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 793–803 (2019)
41. Zhang, S., Hao, A., Qin, H., et al.: Knowledge-inspired 3D scene graph prediction in point cloud. Proceedings of the Advances in Neural Information Processing Systems (NeruIPS) **34**, 18620–18632 (2021)
42. Zhang, Y., Hu, Q., Xu, G., Ma, Y., Wan, J., Guo, Y.: Not all points are equal: Learning highly efficient point-based detectors for 3D lidar point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
43. Zhao, J., Wang, X., Shi, C., Hu, B., Song, G., Ye, Y.: Heterogeneous graph structure learning for graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 35, pp. 4697–4705 (2021)