# Exploiting Edge-Oriented Reasoning for 3D Point-based Scene Graph Analysis

Chaoyi Zhang
University of Sydney
chaoyi.zhang@sydney.edu.au

Jianhui Yu
University of Sydney
jianhui.yu@sydney.edu.au

Yang Song
University of New South Wales
yang.song1@unsw.edu.au

Weidong Cai
University of Sydney
tom.cai@sydney.edu.au

## Abstract

*Scene understanding is a critical problem in computer vision. In this paper, we propose a 3D point-based scene graph generation ($\mathbf{SGG}_{\mathbf{point}}$) framework to effectively bridge perception and reasoning to achieve scene understanding via three sequential stages, namely scene graph construction, reasoning, and inference. Within the reasoning stage, an EDGE-oriented Graph Convolutional Network (EdgeGCN) is created to exploit multi-dimensional edge features for explicit relationship modeling, together with the exploration of two associated twinning interaction mechanisms between nodes and edges for the independent evolution of scene graph representations. Overall, our integrated $\mathbf{SGG}_{\mathbf{point}}$ framework is established to seek and infer scene structures of interest from both real-world and synthetic 3D point-based scenes. Our experimental results show promising edge-oriented reasoning effects on scene graph generation studies. We also demonstrate our method advantage on several traditional graph representation learning benchmark datasets, including the node-wise classification on citation networks and whole-graph recognition problems for molecular analysis.*

## 1. Introduction

Scene understanding is intrinsically close to the essence of computer vision. It simulates human visual system in recognizing the miscellaneous clues concealed in the complex visual world, succeeded by understanding what we perceive in the visual scenes surrounding us [8]. This process could be integrated and assisted with an efficient use of semantic scene graph (**SG**), which has its popularity well-demonstrated within the computer graphics community, via depicting the objects and their inner structural relationships (scene layouts) as its nodes and edges, respectively.

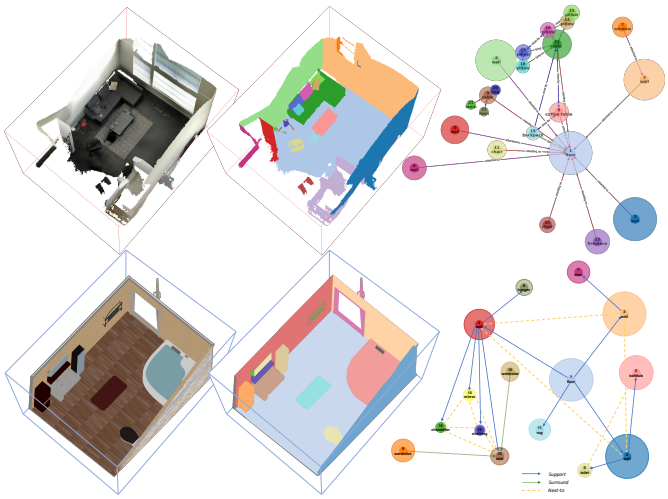

Figure 1. 3D Point-based Scene Graph Generation ($\mathbf{SGG}_{\mathbf{point}}$) takes as inputs **real-world** or **synthetic** 3D scenes $\mathcal{S}$ (left) and class-agnostic instance mask $\mathcal{M}$ (middle) to inference a scene graph $\mathcal{G}$ (right). $\mathcal{M}$ and $\mathcal{G}$ above are aligned to the same spatial layout, sharing a unified instance color encoding.

Unlike most of the successful works proposed for 2D **SG** studies [49, 46, 24, 15], this paper focuses on 3D point-based semantic **SG** analysis – an emerging 3D visual recognition task that has not been well-explored yet. Such methods could provide great aid for arising cross-domain vision tasks including 2D-3D scene retrieval [38], 3D visual grounding [4, 3], and scene captioning [5], which would subsequently benefit real-life applications such as creative interior decoration designs, self-driving autonomous vehicles, or other AI-enriched indoor/outdoor industries.

Within the rising progression of 3D point-based semantic **SG** analysis, the research interests have gradually shifted from object-centric point cloud learning tasks, such as 3D object detection [32, 48, 33], instance segmenta-

Our project page: https://$\mathbf{SGG}_{\mathbf{point}}$.github.io

tion [14, 54, 42], and semantic scene segmentation [13, 47], to the joint recognition of both objects and inter-object structural relationships, which could be further regressed to generate **SG**s describing some desired scene layouts (Fig. 1) for given point-based 3D scenes. Moreover, existing works [49, 38] have mostly treated the inter-object structural relationships as by-products derived from graph node recognition, losing sight of the visual cues lurking inside each **SG** representation and thus degrading their joint recognition performance.

In this paper, we propose a 3D $\mathbf{SGG}_{\mathbf{point}}$ framework capable of effectively bridging perception and reasoning to achieve 3D scene understanding through three sequential stages, namely scene graph construction, reasoning, and inference. The contributions of this paper are summarized as follows: 1) To endow the graph convolution networks (GCNs) with edge-assisted reasoning capability, an edge-oriented GCN (`EdgeGCN`) is proposed to exploit multi-dimensional edge features for explicit inter-node relationship modeling. 2) Two twinning interactions between **SG** nodes and edges are further explored to conduct comprehensive **SG** reasoning for each individual **SG** representation evolution, so that the node- and edge-oriented visual clues can be better perceived and utilized to assist the other ones' evolution via an attentional manner. 3) Our integrated $\mathbf{SGG}_{\mathbf{point}}$ framework is demonstrated to be handy for generating 3D scene structures from either computer-aided 3D scene synthesis or real-world 3D scans, while our edge-driven interaction scheme is also proven beneficial to conventional graph representation learning tasks.

## 2. Related Work

**2D scene graph analysis.** **SG**s were firstly introduced into computer vision to capture more semantic information of objects and their inter-relationships for image retrieval [16]. Thereafter, a string of image-based **SG** generation methods [46, 51, 22, 49, 21, 27] was substantially fostered by the release of the Visual Genome [18] dataset, which includes large-scale **SG** annotations on images. Xu *et al*. [46] adopted gated recurrent units (GRUs) [7] to propagate messages iteratively between the primal and dual graphs formed by **SG** nodes and edges, while MotifNet [51] generated **SG**s from global context parsed through bidirectional LSTM [12]. Most methods [22, 21, 27] tackled **SG** prediction problem within an object detector-cored framework for node- and edge-specific feature extraction, whereas Graph R-CNN [49] proposed an attentional variant of GCN [17] and combined it with Faster R-CNN [31] to process contextual information between objects and relationships. Unlike most of them that treat edge features as by-products derived from the 2D object recognition progress, we address this issue by handling nodes and edges equivalently and simultaneously as a pair of twining representations among 3D point-based scenes.

**3D point-based scene understanding.** Differing from voxelization-based [26] or view-based approaches [35, 29], point cloud processing techniques have been advanced to support direct point-based manipulations on 3D objects or scenes [28, 30, 43]. They transformed 3D scene understanding into several object-centric recognition tasks including semantic scene segmentation [13, 47], scene instance segmentation [14, 54, 42], and scene object detection [32, 48, 33], which ensures the deep learning advances could be inherited from 2D vision to enhance 3D object-oriented recognition performance. Additionally, other concurrent 3D scene understanding works have compiled a few augmented reality focused applications such as indoor scene synthesis and augmentation [19, 40, 39, 55], by producing object recommendation lists for given query positions within 3D class-known scenes. GRAINS [19] adopted recursive auto-encoders for semantic scene completion over the **SG**s being organized in tree structures, while SceneGraphNet [55] achieved iterative scene synthesis by passing relationship-specific messages among **SG** nodes. Apart from these investigations in object-oriented scene recognition, only a few works have spotlighted 3D scene-oriented reasoning and understanding, by encoding the scene layouts of interest or regressing the inter-object structural relationships, due to the lack of 3D **SG** datasets. Recently, the 3RScan [37] dataset, which had been initially probed for 3D object instance re-localization task, was later upgraded as a newly established benchmark [38] for learning 3D semantic **SG**s from point-based indoor environments. In this work, we selected these two datasets to evaluate our approaches on 3D real-world scans. Another synthetic dataset [34] with scene layout annotations released in [55] was also adopted for our method evaluation on 3D synthetic scenes.

**Graph-based reasoning.** Building upon GCNs [17] as their core components, graph reasoning approaches conduct graph-based information propagation to achieve global relation reasoning effects among the graph nodes. GCU [20] initiated a three-stage graph reasoning paradigm for 2D vision tasks with the graph projection, convolution, and re-projection operations, while GloRe [6] and LatenGNN [53] strengthened their global reasoning powers via flexible feature aggregations performed within their so-called interactive (or latent) space. Meanwhile, SGR [23] and GIN [44] inspected contextual reasoning over commonsense graph structures and utilized external knowledge to improve performance on several 2D segmentation benchmarks. However, most existing approaches focused on relation reasoning among graph nodes, neglecting their twinning representations, i.e., graph edges. By contrast, inspired by EGNN [10], we ameliorate GCN to make it compatible with explicit relationship modeling as desired and further exploit edge-oriented reasoning for **SG** representation learning.

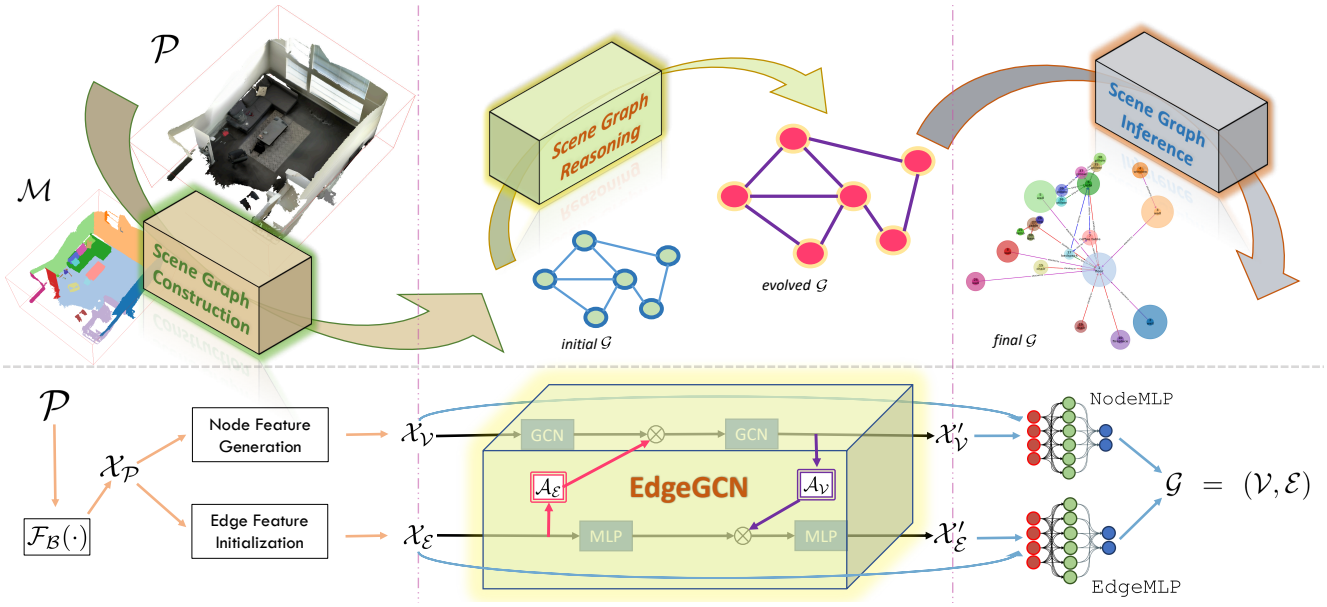


Figure 2. Our proposed 3D point-based scene graph generation ($\mathbf{SGG_{point}}$) framework consisting of three sequential stages.

## 3. Method

Suppose a 3D point cloud $\mathcal{P}$ consists of $\mathcal{N}$ points $\{\mathcal{P}_k\}_{k=1,...,\mathcal{N}}$. The ultimate goal of our point-based scene graph generation ($\mathbf{SGG_{point}}$) framework is to create a scene graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes $\mathcal{V}$ and edges $\mathcal{E}$ depict the instance objects and their inner structural relationships, respectively. This objective can be investigated through several stages: namely scene graph construction ($\text{Construction}_{\mathbf{SG}}$ in *Sec.* 3.1), reasoning ($\text{Reasoning}_{\mathbf{SG}}$ in *Sec.* 3.2), and inference ($\text{Inference}_{\mathbf{SG}}$ in *Sec.* 3.3).

### 3.1. Scene Graph Construction

Compared to the existing work [38] that employed two separate backbone networks to extract independent object- and relationship-specific features, we reduce the superfluous redundancies in scene understanding via sharing one single backbone denoted as $\mathcal{F}_{\mathcal{B}}(\cdot)$ to capture the point-wise features $\mathcal{X}_{\mathcal{P}} \in \mathcal{R}^{\mathcal{N} \times \mathcal{C}_{point}}$ from a specific $\mathcal{P} \in \mathcal{R}^{\mathcal{N} \times \mathcal{C}_{input}}$ that forms scene $S$, where $\mathcal{C}_{input}$ and $\mathcal{C}_{point}$ denote the channel numbers for inputted point clouds and their extracted point-wise features, respectively. $\mathcal{X}_{\mathcal{P}}$ is further propagated to facilitate the initial modeling of the representations of $m$ nodes and $m^2$ one-to-one edges in $\mathcal{G}$, as $\mathcal{X}_{\mathcal{V}} \in \mathcal{R}^{m \times \mathcal{C}_{node}}$ and $\mathcal{X}_{\mathcal{E}} \in \mathcal{R}^{m \times m \times \mathcal{C}_{edge}}$, respectively, where $\mathcal{C}_{node}$ and $\mathcal{C}_{edge}$ indicate the channel numbers for node and edge features constructed within $\mathcal{G}$, respectively.

**Node feature generation.** As suggested in [38], a symmetric pooling function $g(\cdot)$ [28] is performed on an unordered set, along the class-agnostic point-to-instance indicator $\mathcal{M} \in \{1, ..., m\}^{\mathcal{N}}$ to generate the instance-wise visual signatures $\mathcal{X}_{\mathcal{V}_i} \in \mathcal{R}^{1 \times \mathcal{C}_{node}}$ for each object $i$ inside $S$, from the point-wise features $\mathcal{X}_{\mathcal{P}}$ obtained by $\mathcal{F}_{\mathcal{B}}(\cdot)$. This masking operation can be formally described as:

$$\mathcal{X}_{\mathcal{V}_i} = g\Big(\big\{\delta(\mathcal{M}_k, i) \cdot \mathcal{X}_{\mathcal{P}_k}\big\}_{k=1,...,\mathcal{N}}\Big), \tag{1}$$

where $\delta(\cdot, \cdot)$ denotes the Kronecker Delta. Our initial node features can now be modeled as $\mathcal{X}_{\mathcal{V}}$ by stacking together all $m$ instance-wise visual signatures across $S$.

**Edge feature initialization.** In contrast to some **SG** studies on 2D images [15] or 3D point clouds [38] that reformulated inter-object structural relationships as special kinds of nodes, the $\mathbf{SGG_{point}}$ framework would instead learn and encapsulate this information as multi-dimensional edge features $\mathcal{X}_{\mathcal{E}}$. Each member $\mathcal{X}_{\mathcal{E}_{(i,j)}} \in \mathcal{R}^{1 \times 1 \times \mathcal{C}_{edge}}$ records the $\mathcal{C}_{edge}$-dim status of each directional connection $\mathcal{E}_{(i,j)}$ that points from subject $\mathcal{V}_i$ toward object $\mathcal{V}_j$, which can be initialized as $\mathcal{X}_{\mathcal{E}_{(i,j)}} = \big(\mathcal{X}_{\mathcal{V}_i} +\!\!+ (\mathcal{X}_{\mathcal{V}_j} - \mathcal{X}_{\mathcal{V}_i})\big)$ using feature engineering and concatenation scheme introduced in [43].

### 3.2. Scene Graph Reasoning via EdgeGCN

Previous **SG** works mostly acquired edge predictions as by-products derived from node representation learning, which might underestimate potential impacts of the visual cues lurking inside both node and edge representations toward their joint $\mathbf{SGG_{point}}$ task. Instead, we posit that both nodes and edges are expected to be treated equally and processed simultaneously as pairs of twinning representations within a given **SG**, and we thus assign each one with an exclusive learning branch and investigate graph reasoning techniques for their feature representation enhancements.

Recall the recently proposed global relation reasoning approaches [6, 23, 44, 20] that applied GCNs to perform

node-wise message propagation to obtain their so-called *evolved node features* through graph-based reasoning [23]. We first borrow their ideas and naming rules to establish our **SG** node evolution stream, where an edge-driven interaction mechanism dubbed twinning edge attention, is proposed to enhance node-wise reasoning. Similarly, an edge evolution stream equipped with twining node attention scheme is next designed to extract node-specific cues for conducting edge-wise reasoning.

#### 3.2.1 Twinning Edge Attention for Node Evolution

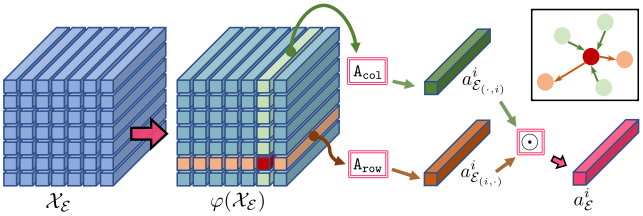


Figure 3. Twinning edge attention $\mathcal{A}_\mathcal{E}$ design within our EdgeGCN for modeling the edge-driven interactions toward node evolution.

**Twinning edge attention.** Our twinning edge attention scheme is proposed to learn a multi-dimensional attention mask $\mathcal{A}_\mathcal{E} \in \mathcal{R}^{m \times \mathcal{C}'_{node}}$ to be applied over $\mathcal{V}$ in accordance with their node-wise importance cues embedded in $\mathcal{X}_\mathcal{E}$, where $\mathcal{C}'_{node}$ is pre-defined here to match the inner channel number within node evolution stream to be described below. To make use of the directional status recorded in $\mathcal{X}_\mathcal{E}$, given a node $\mathcal{V}_i$, we compute its edge interaction vector $a^i_\mathcal{E} \in \mathcal{R}^{1 \times \mathcal{C}'_{node}}$ by considering both circumstances when it plays the roles of sources or targets in various connections. Mathematically, the outgoing interaction signals emitted by $\mathcal{V}_i$ (as sources) and the incoming interaction signals received by $\mathcal{V}_i$ (as targets) can be captured and aggregated as $a^i_{\mathcal{E}_{(i,\cdot)}}$ and $a^i_{\mathcal{E}_{(\cdot,i)}}$, respectively, through:

$$a^i_{\mathcal{E}_{(i,\cdot)}} = \mathtt{A_{row}}\left(\{W^T_\varphi \mathcal{X}_{\mathcal{E}_{(i,k)}} \mid \forall \mathcal{V}_k\}\right), \tag{2}$$

$$a^i_{\mathcal{E}_{(\cdot,i)}} = \mathtt{A_{col}}\left(\{W^T_\varphi \mathcal{X}_{\mathcal{E}_{(k,i)}} \mid \forall \mathcal{V}_k\}\right), \tag{3}$$

where $W_\varphi \in \mathcal{R}^{\mathcal{C}_{edge} \times \mathcal{C}'_{node}}$ is a trainable transformation matrix for converting each edge feature $\mathcal{X}_{\mathcal{E}_{(\cdot,\cdot)}} \in \mathcal{R}^{\mathcal{C}_{edge}}$ into the dimension $\mathcal{C}'_{node}$, while $\mathtt{A_{row}}(\cdot)$ and $\mathtt{A_{col}}(\cdot)$ represent the channel-wise aggregation functions performed along row- and column-directions, respectively. Hence, as demonstrated in Fig. 3, the overall edge-driven interaction score of node $\mathcal{V}_i$ can now be jointly learned as:

$$a^i_\mathcal{E} = \sigma(a^i_{\mathcal{E}_{(i,\cdot)}} \odot a^i_{\mathcal{E}_{(\cdot,i)}}), \tag{4}$$

where $\odot$ denotes the Hadamard Product and $\sigma$ indicates the sigmoid function to emphasize the meaningful interactions and suppress the uninformative ones.

**Node evolution stream.** Suppose $A_\mathcal{G}$ as the adjacency matrix defined over $\mathcal{G}$. Based on the definition of edge-driven twinning attention, our evolved **SG** node representation $\mathcal{X}'_\mathcal{V}$ could now be learnt as:

$$\mathcal{X}'_\mathcal{V} = f\Big(\widehat{A}_\mathcal{G}\Big(f(\widehat{A}_\mathcal{G}\mathcal{X}_\mathcal{V}W_{\mathtt{G1}}) \odot \mathcal{A}_\mathcal{E}\Big)W_{\mathtt{G2}}\Big), \tag{5}$$

where $\mathcal{A}_\mathcal{E}$ and $f$ denote the edge-driven interactive score and non-linear activation function, respectively, and $\widehat{A}_\mathcal{G}$ is a symmetric Laplacian matrix normalized from $A_\mathcal{G} + I$ such that all rows sum to one [17], while $W_{\mathtt{G1}} \in \mathcal{R}^{\mathcal{C}_{node} \times \mathcal{C}'_{node}}$ and $W_{\mathtt{G2}} \in \mathcal{R}^{\mathcal{C}'_{node} \times \mathcal{C}_{node}}$ are learnable weights for two consecutive graph convolution layers to squeeze and expend the channels of node features through $\mathcal{C}_{node} \rightarrow \mathcal{C}'_{node} \rightarrow \mathcal{C}_{node}$. **Note:** The inner layer outputs in Eq. 5 and 7 are indicated in *dark brown* for convenience. Unlike [55] which provides detailed relationship categorical information to guide their node evolution and employs several independent GRUs to conduct the message passing for each kind of relationship, we instead only reveal the class-agnostic relationship existences (i.e., $A_\mathcal{G}$) for our approach designs, leading to a flexible scalability upon the various dataset-dependent inter-object relationship annotations across datasets.

#### 3.2.2 Twinning Node Attention for Edge Evolution

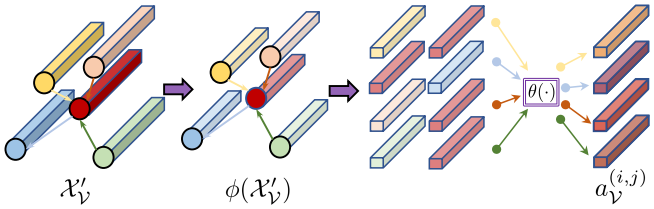


Figure 4. Twinning node attention $\mathcal{A}_\mathcal{V}$ design inside our EdgeGCN for modeling the node-driven interactions toward edge evolution.

**Twinning node attention.** Similarly, the interactions made by the source-nodes and target-nodes, toward their reaching edges, are modeled by another multi-dimensional attention mask $\mathcal{A}_\mathcal{V} \in \mathcal{R}^{m \times m \times \mathcal{C}'_{edge}}$ to be assigned upon $\mathcal{E}_{(\cdot,\cdot)}$, where $\mathcal{C}'_{edge}$ is pre-set here as the inner channel number of edge evolution stream to be demonstrated below. Specifically, given nodes $i$ and $j$, together with the directional edge connecting them $\mathcal{E}_{(i,j)}$, their resulting edge-wise node-driven interaction score $a^{(i,j)}_\mathcal{V} \in \mathcal{R}^{1 \times 1 \times \mathcal{C}'_{edge}}$ can be learnt from the concatenation of evolved node features that belongs to the sources and targets, as:

$$a^{(i,j)}_\mathcal{V} = \sigma\left(W^T_\theta f(W^T_\phi \mathcal{X}'_{\mathcal{V}_i} + W^T_\phi \mathcal{X}'_{\mathcal{V}_j})\right), \tag{6}$$

where $W_\theta \in \mathcal{R}^{2\mathcal{C}'_{edge} \times \mathcal{C}'_{edge}}$ and $W_\phi \in \mathcal{R}^{\mathcal{C}_{node} \times \mathcal{C}'_{edge}}$ represent the learnable weight matrices of a two-layer structure

for transforming channel number from $2\mathcal{C}'_{edge}$ to $\mathcal{C}'_{edge}$, and from $\mathcal{C}_{node}$ to $\mathcal{C}'_{edge}$, respectively.

**Edge evolution stream.** With node-driven twinning attention defined, the other evolved edge feature $\mathcal{X}'_{\mathcal{V}}$ could now be obtained through:

$$\mathcal{X}'_{\mathcal{E}} = f\Big(W^T_{\texttt{FC2}}\Big(f(W^T_{\texttt{FC1}}\mathcal{X}_{\mathcal{E}}) \odot \mathcal{A}_{\mathcal{V}}\Big)\Big), \tag{7}$$

where $\mathcal{A}_{\mathcal{V}}$ is the node-driven interactive score, while $W_{\texttt{FC1}} \in \mathcal{R}^{\mathcal{C}_{edge} \times \mathcal{C}'_{edge}}$ and $W_{\texttt{FC2}} \in \mathcal{R}^{\mathcal{C}'_{edge} \times \mathcal{C}_{edge}}$ are trainable parameters for two fully-connected layers to transform edge features through $\mathcal{C}_{edge} \rightarrow \mathcal{C}'_{edge} \rightarrow \mathcal{C}_{edge}$.

#### 3.2.3 EdgeGCN

As illustrated in Fig. 2, a joint reasoning module dubbed EdgeGCN capable of exploiting edge features for more comprehensive graph reasoning performed over $\mathcal{G}$ is designed to take as inputs $\mathcal{X}_{\mathcal{V}}$ and $\mathcal{X}_{\mathcal{E}}$ that are initially formed in the Construction$_{\mathbf{SG}}$ stage, conduct collaborative message propagation between two twinning **SG** representations enriched by their corresponding edge- and node-driven interactions, and produce the evolved ones to be used by the Inference$_{\mathbf{SG}}$ stage. More specifically, EdgeGCN contains two feature evolution streams for the nodes ($\mathcal{X}_{\mathcal{V}} \rightarrow \mathcal{X}'_{\mathcal{V}}$) and edges ($\mathcal{X}_{\mathcal{E}} \rightarrow \mathcal{X}'_{\mathcal{E}}$), and the evolution stream of each representation is endowed with an attentional interaction mechanism ($\mathcal{A}_{\mathcal{E}}$ or $\mathcal{A}_{\mathcal{V}}$) to escort the interdependence between itself and its twinning representation. The detailed architecture designs can be viewed in *Sec.* 3.4.

As its name suggests, the most distinctive attribute of our EdgeGCN is the explicit modeling of multi-dimensional edge features and their effective interactions with node features for **SG** reasoning, compared to other node-wise graph reasoning approaches [6, 44]. Noticeably, a Vanilla EdgeGCN without any interactive designs, i.e., $\mathcal{A}_{\mathcal{E}} = \mathcal{A}_{\mathcal{V}} = \mathbb{1}$, could be built as two isolated representation learning branches consisting of a two-layer GCN and a two-layer MLP for the independent node and edge evolution.

### 3.3. Scene Graph Inference

The final **SG** recognition results are predicted on the evolved node features $\mathcal{X}'_{\mathcal{V}}$ and edge features $\mathcal{X}'_{\mathcal{E}}$. Two Multilayer Perceptron based inference streams are established as NodeMLP and EdgeMLP to perform the recognition of objects and their inner structural relationships, respectively. Moreover, NodeMLP and EdgeMLP share the same network structure of two fully-connected layers, but with individual learnable parameters to convert channel numbers through $\mathcal{C}_{in} \rightarrow \frac{\mathcal{C}_{in}}{2} \rightarrow \mathcal{C}_{out}$, where $\mathcal{C}_{in}$ indicates their corresponding input channels and $\mathcal{C}_{out}$ equals to the number of object classes or relationship classes.

### 3.4. Implementation Details

For the specific instances of $\mathcal{F}_{\mathcal{B}}(\cdot)$ adopted in Construction$_{\mathbf{SG}}$ stage, we chose the pioneering PointNet [28] and its promising follower DGCNN [43] for their concise but effective architecture design philosophy, as well as the dynamic and powerful context modeling of each local neighborhood in semantic spaces, respectively. Concretely, we set $\mathcal{C}_{input}$ to 9 including 3-dim coordinates, 3-dim RGB colors and 3-dim normal vectors, while $\mathcal{C}_{point}$ set to 256 for unified point-wise feature extraction. Within the Reasoning$_{\mathbf{SG}}$ stage, we set $\mathcal{C}_{node} = 2 \times \mathcal{C}'_{node} = 256$, and $\mathcal{C}_{edge} = 2 \times \mathcal{C}'_{edge} = 512$ for **SG** node and edge evolution streams, respectively, while we used ReLU for $f(\cdot)$ and adopted the Synchronized BatchNorm [52] for multi-GPU training. Regarding the Inference$_{\mathbf{SG}}$ stage, two multi-class cross entropy losses $\mathcal{L}_{node}$ and $\mathcal{L}_{edge}$ were applied for their corresponding **SG** representation learning, and hence, the integrated **SGG$_{\text{point}}$** framework could be supervised via a joint loss $\mathcal{L}_{\mathbf{SG}} = \mathcal{L}_{node} + \mathcal{L}_{edge}$. Please refer to the supplementary materials[1] for the training details for each dataset.

Before being fed into the Inference$_{\mathbf{SG}}$ stage, the evolved **SG** representations obtained from EdgeGCN are combined with the initial ones via a residual connection [11], to further enhance the discriminative power of graph-based reasoning approaches for **SGG$_{\text{point}}$** studies, which is unnecessary for conventional graph representation learning tasks and thus omitted to match up with other GNN setups.

## 4. Experiments

We evaluated the **SGG$_{\text{point}}$** framework on both real-world (*Sec.* 4.1) and synthetic 3D scenes (*Sec.* 4.3), with extensive ablation studies conducted on real-world ones (*Sec.* 4.2) to demonstrate the individual contribution of each proposed component toward the overall quality of generated **SG**s. Despite the studies of **SG** representation learning, we also verify the proposed EdgeGCN on five conventional graph representation learning tasks (*Sec.* 4.4), including three node-wise classification problems and two whole-graph recognition problems.

### 4.1. 3D SGG$_{\text{point}}$ on Real-World 3D Scans

**Dataset and evaluation details.** We first validated the effectiveness of our proposed methods on real-world 3D scans using the 3RScan [37] dataset. Extending [37] with [38] results in over one thousand 3D indoor point cloud reconstructions, as well as their corresponding semantic 3D **SG** annotations including 27 object classes and 16 relationship categories (details in supplementary materials). For evaluation, we applied the same scene-level split specified in [38] on the point cloud representations, which were densely sampled from their released surface reconstructions using

[1] https://**SGG$_{\text{point}}$**.github.io/supplementary.pdf

| Graph Reasoning Approach | Object Class Prediction | | Predicate Class Prediction | | Relationship Triplet Prediction | |
|---|---|---|---|---|---|---|
| | R@5 | R@10 | F1@3 | F1@5 | R@50 | R@100 |
| $\mathcal{F}_{\mathcal{B}}(\cdot)$ *alone* | *87.40* | *96.26* | *68.55* | *82.79* | *34.97* | *45.86* |
| + GCN (SGPN) [38] * | 89.61$^{\uparrow}$ | 96.98$^{\uparrow}$ | 63.58$^{\downarrow}$ | 77.79$^{\downarrow}$ | 32.45$^{\downarrow}$ | 41.65$^{\downarrow}$ |
| + GloRe$_{\text{PC}}$ [25] | 84.06$^{\downarrow}$ | 95.17$^{\downarrow}$ | 69.23$^{\uparrow}$ | 80.01$^{\downarrow}$ | 31.87$^{\downarrow}$ | 42.21$^{\downarrow}$ |
| + GloRe$_{\text{SG}}$ [6] | 85.27$^{\downarrow}$ | 96.62$^{\uparrow}$ | 72.57$^{\uparrow}$ | 83.42$^{\uparrow}$ | 29.58$^{\downarrow}$ | 38.64$^{\downarrow}$ |
| + **EdgeGCN** (our $\mathbf{SGG}_{\mathbf{point}}$) | **90.70**$^{\uparrow}$ | **97.58**$^{\uparrow}$ | **78.88**$^{\uparrow}$ | **90.86**$^{\uparrow}$ | **39.91**$^{\uparrow}$ | **48.68**$^{\uparrow}$ |

Table 1. Results on real-world 3D scans. Note: $*$ denotes the usage of two separate $\mathcal{F}_{\mathcal{B}}(\cdot)$ within Construction$_{\mathbf{SG}}$ stage, for independent feature extractions of initial node and edge representations in $\mathcal{G}$.

CloudCompare [1], with all mesh information discarded and surface density set as 10k points per square unit.

Following [46, 49, 38], the scene graph prediction performance of the $\mathbf{SGG}_{\mathbf{point}}$ framework was evaluated upon the three perspectives, namely object class prediction, predicate class prediction, and relationship triplet prediction. More specifically, we adopted the top-k recall metric used in [24] for object class prediction and computed the macro-F1 score for predicate class prediction, due to the imbalanced predicate class distribution in [38]. The relationship triplet prediction was jointly generated as an ordered list of (`subject`, `predicate`, `object`) triplets, whose triplet-level confidence scores were obtained by multiplying each respective score [49], and the most confident ones are separated for evaluation against the ground truth annotations [38].

**Experimental results.** We first set the $\mathcal{F}_{\mathcal{B}}(\cdot)$ *alone* baseline without invoking any graph reasoning modules. Note: $\mathcal{F}_{\mathcal{B}}(\cdot)$ was implemented as PointNet [28] here to make fair comparisons with the current benchmark [38] and justify various graph reasoning effects, while we also demonstrated that our method could be further enhanced consistently by changing the backbone in the coming ablation studies.

We then reproduced SGPN [38], which was similar to [15] that treated both objects and their interrelationships as graph nodes to conduct message propagation with GCN [17], to generate the acquired triplets. As shown in Table 1, employing GCN for scene graph reasoning in an intuitive way as presented in [38] could improve the object class recognition but may harm the performance in other two **SG** tasks, which confirms the empirical findings reported in [38] on over-smoothing issue caused by multi-layer GCNs. We next verified GloRe modules to perform global relation reasoning on scene graph representation learning using official implementations in their released repository [6]. The GloRe module could be applied at various positions to reach reasoning effects at two different levels, namely the point cloud level (GloRe$_{\text{PC}}$) [25] and scene graph level (GloRe$_{\text{SG}}$) [6]. In contrast to SGPN, both GloRe$_{\text{PC}}$ and GloRe$_{\text{SG}}$ tend to benefit the predicate class prediction and may damage the recognition under two other metrics as a trade-off. Our `EdgeGCN` achieved the best results under all evaluation metrics, which demonstrated the superiority of edge-oriented relationship modeling, as well as its associated twinning attentions between graph nodes and edges, for scene graph reasoning. The qualitative visualization can be viewed in Fig. 5.

### 4.2. Ablation Studies

To reveal the precise performance gain of each proposed component, we instead reported the respective recognition results of objects and relationships in top-1 manner within our ablation studies. `Models A-C` were raised to establish the baselines of the $\mathbf{SGG}_{\mathbf{point}}$ framework, especially for the Construction$_{\mathbf{SG}}$ and Inference$_{\mathbf{SG}}$ stages, while `models D-G` were built to demonstrate the effectiveness of adopting multi-dimensional edge features $\mathcal{X}_{\mathcal{E}}$ for scene graph reasoning, together with two associated twinning interactions between nodes and edges.

**Model designs.** More specifically, `model A` indicates the baseline performance of utilizing backbone networks for object classification in the **SG** context, leaving the initialization of $\mathcal{X}_{\mathcal{E}}$, edge evolution stream, and

| ID | Task | Reasoning$_{\mathbf{SG}}$ | | | **SG** Recognition | |
|---|---|---|---|---|---|---|
| | | GNNs | $\mathcal{A}_{\mathcal{E}}$ | $\mathcal{A}_{\mathcal{V}}$ | node $_{\text{R@1}}$ | edge $_{\text{F1@1}}$ |
| A w/ $\diamond$ | N | - | - | - | 48.6 | - |
| B w/ $\diamond$ | N | GCN | - | - | 54.4 | - |
| C w/ $\diamond$ | N+E | - | - | - | 48.4 | 38.7 |
| D w/ $\diamond$ | N+E | EdgeGCN | $\times$ | $\times$ | 54.8 ④ | 41.9 ③ |
| E w/ $\diamond$ | N+E | EdgeGCN | $\surd$ | $\times$ | 56.9 ② | 41.1 ④ |
| F w/ $\diamond$ | N+E | EdgeGCN | $\times$ | $\surd$ | 56.4 ③ | **50.0** ① |
| **G w/ $\diamond$** | **N+E** | **EdgeGCN** | $\surd$ | $\surd$ | **57.1** ① | 48.7 ② |
| A w/ $\star$ | N | - | - | - | 58.0 | - |
| B w/ $\star$ | N | GCN | - | - | 61.3 | - |
| C w/ $\star$ | N+E | - | - | - | 57.1 | 39.6 |
| D w/ $\star$ | N+E | EdgeGCN | $\times$ | $\times$ | 60.8 ④ | 43.9 ③ |
| E w/ $\star$ | N+E | EdgeGCN | $\surd$ | $\times$ | 61.7 ② | 41.1 ④ |
| F w/ $\star$ | N+E | EdgeGCN | $\times$ | $\surd$ | 61.0 ③ | 47.4 ② |
| **G w/ $\star$** | **N+E** | **EdgeGCN** | $\surd$ | $\surd$ | **62.5** ① | **49.7** ① |

Table 2. Ablation studies of $\mathbf{SGG}_{\mathbf{point}}$ framework. Task (N) and Task (E) represent the node and edge recognition tasks for $\mathbf{SGG}_{\mathbf{point}}$ studies, respectively, while dash lines indicate 'not applicable'. Two specific $\mathcal{F}_{\mathcal{B}}(\cdot)$ implementations include PointNet ($\diamond$) and DGCNN ($\star$). ⓚ denotes rankings within each sub-block.

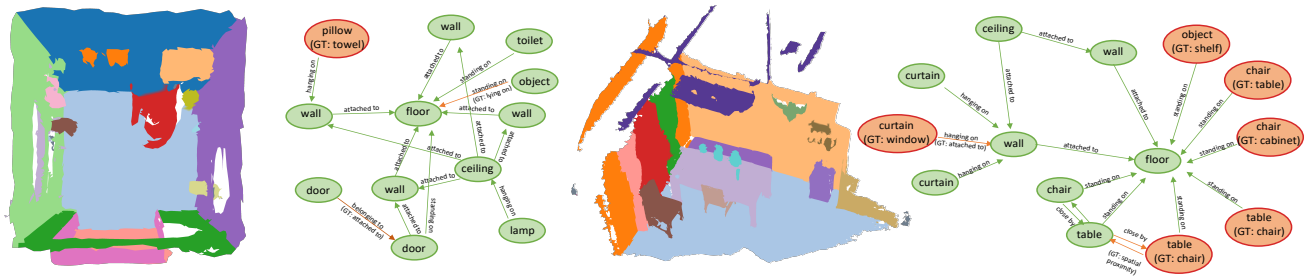


Figure 5. Qualitative analysis of the $\mathbf{SGG}_{\mathbf{point}}$ framework. For visualization purpose, misclassified object or structural relationship predictions are indicated with ground truth (GT) values in red, while the correct ones are shown in green with GT values omitted.

`EdgeMLP` untouched in $\text{Construction}_{\mathbf{SG}}$, $\text{Reasoning}_{\mathbf{SG}}$, and $\text{Inference}_{\mathbf{SG}}$ stages, respectively. `Model B` enhances `model A` via the same two-layer GCN adopted in our `EdgeGCN` to perform a $\mathcal{X}_{\mathcal{E}}$-uninvolved graph reasoning, which relies on the node information alone, where as `model C` enriches `model A` to be compatible with the joint recognition tasks for **SG** objects and relationships via adding edge-wise supervisions into the model training. Furthermore, a `Vanilla EdgeGCN` is trained, as `model D`, to fulfill `model B` with the necessary components for relationship recognition. Then, two twinning interactive attention schemes are equipped to `model D`, investigating the independent impacts of edge-driven and node-driven interactions from `model E` and `F`, respectively. At the end, with the aim of reaching a comprehensive $\mathcal{X}_{\mathcal{E}}$-involved **SG** reasoning effect, `model G` is constructed with collaborative improvements invoked from both node and edge sides for **SG** representation evolution.

**Discussions.** As details revealed in Table 2, graph reasoning could benefit the object recognition process ($\texttt{A}^{\diamond/\star} \rightarrow \texttt{B}^{\diamond/\star}$), while roughly adding edge-wise supervision may harm the outcomes by contrast ($\texttt{A}^{\diamond/\star} \rightarrow \texttt{C}^{\diamond/\star}$) and we attribute it as the potential distractions on the unified extraction of point-wise features via $\mathcal{F}_{\mathcal{B}}(\cdot)$. These distractions could be partially alleviated by adding extra trainable parameters such that more degree of freedom might be positively introduced into the edge representation learning ($\texttt{B}^{\diamond} \rightarrow \texttt{D}^{\diamond}$; $\texttt{C}^{\diamond/\star} \rightarrow \texttt{D}^{\diamond/\star}$), which thus forms the edge evolution stream in our `Vanilla EdgeGCN`. The effectiveness of $\mathcal{A}_{\mathcal{E}}$ could be somewhat controversial, as it brings improvements on node recognition results yet reduces the edge ones as a trade-off ($\texttt{D}^{\diamond/\star} \rightarrow \texttt{E}^{\diamond/\star}$), and we blame it as similar distractions discussed above. The other twinning interaction module $\mathcal{A}_{\mathcal{E}}$ could accelerate the edge evolution as expected, without distressing the edge evolution ($\texttt{D}^{\diamond/\star} \rightarrow \texttt{F}^{\diamond/\star}$). Stunningly, combining $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{A}_{\mathcal{V}}$ not only achieves the best performance in terms of object classifications ($\{\texttt{E}, \texttt{F}\}^{\diamond/\star} \rightarrow \texttt{G}^{\diamond/\star}$), but also practically curbs the distractions made by $\mathcal{A}_{\mathcal{E}}$ ($\texttt{E}^{\diamond} \rightarrow \texttt{G}^{\diamond}$) and even produces the best outcomes ($\texttt{E}^{\star} \rightarrow \texttt{G}^{\star}$) in terms of relationship predictions.

### 4.3. 3D $\mathbf{SGG}_{\mathbf{point}}$ on Synthetic 3D Scenes

**Dataset and evaluation details.** We further analyzed our methods on the SUNCG [34] dataset, to verify its generalization ability on synthetic 3D scenes. The SUNCG dataset is comprised of over 45k 3D virtual scenes, which were manually created with the Planner5d platform [2] in four room categories, i.e., office, bedroom, bathroom, and living room. As suggested in [55], we filtered out the non-rectangular scenes to maintain fair comparisons with previous studies [19, 40, 55], then repeated the whole experimental procedure independently for each synthetic room category [40, 39], as each category owns unique objects.

For evaluation, we followed the same dataset splitting and preprocessing policy in [55] and adopted three-class inter-object relationship annotations (support, surround, and next-to) [55, 19] as the predicate ground truth to train $\mathbf{SGG}_{\mathbf{point}}$ on synthetic 3D scenes, with point cloud sampling settings unchanged to previous studies. The **SG** object classification accuracy was reported to compare $\mathbf{SGG}_{\mathbf{point}}$ with other existing SOTAs on SUNCG dataset.

**Results and discussions.** Some early methods [19, 40] on this dataset were merely designed for scene synthesis studies, where they firstly removed the target objects from the scenes and then utilized their surrounding contexts to predict the missing labels for scene synthesis evaluations. We thus recognized their approaches as *missing* object prediction studies and included their results as reference benchmarks (in Table 3) for comparisons with traditional object recognition methods such as [35] and [55]. Unlike these missing object predictions who treat target objects

| Method | Bed | Living | Bath | Office |
|---|---|---|---|---|
| GRAINs [19] * | 45.1 | 43.7 | 42.4 | 45.6 |
| Wang et al. [40] * | 48.9 | 46.6 | 61.4 | 46.6 |
| MVCNN [35] | 69.6 | 55.8 | 43.4 | 67.8 |
| SceneGraphNet [55] * | 66.8 | 67.6 | **69.8** | 64.8 |
| SceneGraphNet [55] | **79.9** | 74.7 | 56.4 | 73.0 |
| $\mathbf{SGG}_{\mathbf{point}}$ **w/** $\diamond$ | 79.5 | **76.2** | 61.6 | **74.1** |

Table 3. Results on synthetic 3D scenes, where $*$ denotes missing object predictions achieved by scene synthesis based approaches.

as empty nodes and take as inputs the contextual scene formed by other available nodes in $\mathcal{G}$, the traditional ones including ours would instead extract visual features from target objects themselves for further usage (e.g., `EdgeGCN`). As shown in Table 3, our $\mathbf{SGG_{point}}$ outperformed existing SOTAs on *Living* and *Office* datasets, and we achieved on-par result on *Bed* category. In contrast to [55], our `EdgeGCN` employed multi-dimensional edge features and it is thus insensitive to specific types of inter-object relationships. Besides, compared to the multi-view based approaches [35, 55], $\mathbf{SGG_{point}}$ supports direct point-wise manipulations on 3D scenes via a more efficient manner, in terms of the time and space complexity [28].

### 4.4. Graph Representation Learning

The effectiveness of our `EdgeGCN` could also be verified on graph representation learning studies, such as node-wise classification and whole-graph recognition problems. More specifically, our method was evaluated on three popular citation network datasets (Cora, CiteSeer, and Pubmed) [50] and two molecular datasets (Tox21 and BBBP) [45].

Since these conventional graph representation learning tasks do not provide edge annotation, we thus omitted our edge evolution stream, together with its associated twinning node attention mechanism, and compared the resulting `EdgeGCN` ($\mathcal{A}_{\mathcal{E}}$) with its counterparts including GCN [17], GAT [36], and EGNNs, i.e., EGNN(A) and EGNN(C), which were reproduced in accordance to their reported settings [10]. We applied a Pytorch Geometric [9] script and a DGL [41] script, for evaluations conducted on citation network datasets and molecular datasets, respectively. We kept all specific training settings unchanged for all method evaluations, except for repeating their procedure 50 times for each approach and reporting the averaged accuracy (Accu.), or area under the ROC curve (AUC), with standard deviation to reach reliable comparisons.

**Node-wise classification for citation analysis.** The GCN, GAT, and our `EdgeGCN` ($\mathcal{A}_{\mathcal{E}}$) were constructed as two-layer networks. Since [9] does not provide a universal GAT implementation, we reproduced GAT with various settings

| GNNs | Node Accu. | | | Graph AUC | |
|---|---|---|---|---|---|
| | Cora | CiteSeer | Pubmed | Tox21 | BBBP |
| GCN | $80.3_{\pm 0.7}$ | $67.7_{\pm 0.8}$ | $78.5_{\pm 0.5}$ | $73.1_{\pm 1.1}$ | $64.3_{\pm 3.5}$ |
| GAT (8, 4) | $79.8_{\pm 0.8}$ | $68.1^{+0.4\uparrow}_{\pm 0.9}$ | $76.7_{\pm 0.7}$ | $68.3_{\pm 1.8}$ | $\mathbf{65.1}^{+0.8\uparrow}_{\pm 1.9}$ |
| GAT (8, 8) | $79.5_{\pm 0.7}$ | $68.0_{\pm 1.1}$ | $76.4_{\pm 0.8}$ | | |
| GAT (16, 4) | $79.7_{\pm 1.0}$ | $67.9_{\pm 1.0}$ | $76.4_{\pm 0.9}$ | | |
| GAT (16, 8) | $79.8_{\pm 1.0}$ | $67.5_{\pm 1.6}$ | $76.1_{\pm 1.1}$ | | |
| EGNN(A) | $81.1^{+0.8\uparrow}_{\pm 0.7}$ | $68.5^{+0.8\uparrow}_{\pm 0.8}$ | $79.4_{\pm 0.5}$ | $73.3^{+0.2\uparrow}_{\pm 1.2}$ | $64.5^{+0.2\uparrow}_{\pm 3.1}$ |
| EGNN(C) | $80.9_{\pm 0.7}$ | $67.9_{\pm 0.6}$ | $\mathbf{79.5}^{+1.0\uparrow}_{\pm 0.4}$ | $73.2^{+0.1\uparrow}_{\pm 1.2}$ | $63.9_{\pm 2.9}$ |
| **EdgeGCN** | $\mathbf{81.6}^{+1.3\uparrow}_{\pm 0.7}$ | $\mathbf{69.4}^{+1.7\uparrow}_{\pm 0.9}$ | $78.7^{+0.2\uparrow}_{\pm 0.4}$ | $\mathbf{73.7}^{+0.6\uparrow}_{\pm 0.9}$ | $64.6^{+0.3\uparrow}_{\pm 3.8}$ |

Table 4. Conventional graph representation learning tasks, including node classification on citation network datasets and graph recognition for molecular analysis.

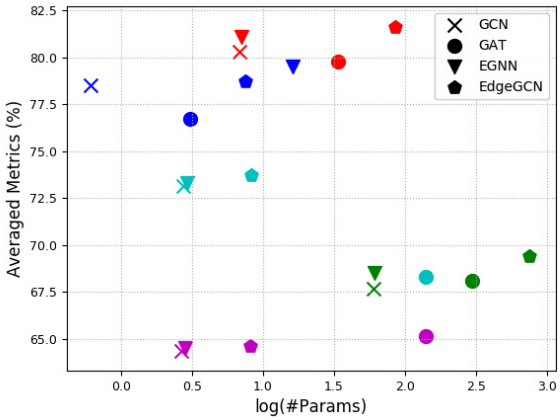


Figure 6. GNN comparisons on various graph representation learning tasks on *Cora*, *CiteSeer*, *Pubmed*, *Tox21*, and *BBBP* datasets. `#Params` denotes the number of trainable parameters (K).

according to [36], where GAT ($C_{inner}$, $K$) denotes its network settings, i.e., $C_{inner}$ inner channels and $K$ attention heads. $C_{inner}$ was set by default to 16 for all other graph networks. As shown in Table 4, our `EdgeGCN` ($\mathcal{A}_{\mathcal{E}}$) has shown its superior performance over all competitors under the same training settings on Cora and CiteSeer, and achieved on-par result on Pubmed. Unlike other designs such as GAT and EGNN, our method does not rely on the hyper-parameter settings or different types of layer instance.

**Whole-graph recognition for molecular analysis.** We adopted the universal three-layer instances of GCN and GAT provided by [41] and extended our `EdgeGCN` to three layers as well, with $\mathcal{A}_{\mathcal{E}}$ inserted to the second layer. As shown in Table 4, the significance of our `EdgeGCN`($\mathcal{A}_{\mathcal{E}}$) design could be verified on both Tox21 and BBBP datasets under the same evaluation protocol applied. Fig. 6 demonstrates the trade-off between effectiveness and efficiency.

## 5. Conclusion

To endow GCNs with edge-assisted reasoning capability, we introduced an edge-oriented GCN dubbed `EdgeGCN` to learn a pair of twinning interactions between nodes and edges, so that comprehensive **SG** reasoning could thus be conducted to enhance each individual evolution. Taking `EdgeGCN` as the core component, we proposed an integrated $\mathbf{SGG_{point}}$ framework to tackle 3D point-based scene graph generation problems through three sequential stages. Overall, our integrated $\mathbf{SGG_{point}}$ framework was established to seek and infer scene structures of interest from both real-world and synthetic 3D point-based scenes. Moreover, we also validated our edge-driven reasoning scheme on conventional graph representation learning benchmark datasets for citation network and molecular analysis.

## References


[1] CloudCompare, version 2.11.1, gpl software. http://www.cloudcompare.org/. Accessed: 2020-11-15. 6

[2] Home design software and interior design tool online for home and floor plans in 2D and 3D. https://planner5d.com. Accessed: 2020-11-15. 7

[3] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 422–440, 2020. 1

[4] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 202–221, 2020. 1

[5] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. Scan2Cap: Context-aware dense captioning in RGB-D scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[6] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 433–442, 2019. 2, 3, 5, 6

[7] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of the Syntax, Semantics and Structure in Statistical Translation (SSST)*, pages 103–111, 2014. 2

[8] Li Fei-Fei, Christof Koch, Asha Iyer, and Pietro Perona. What do we see when we glance at a scene? *Journal of Vision (JOV)*, 4(8):863–863, 2004. 1

[9] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *Proceedings of ICLR Workshop on Representation Learning on Graphs and Manifolds (ICLRW)*, 2019. 8

[10] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 8

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 2

[13] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[14] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3D instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[15] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3, 6

[16] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015. 2

[17] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 2, 4, 6, 8

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 123(1):32–73, 2017. 2

[19] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. GRAINS: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)*, 37, 2018. 2, 7

[20] Yin Li and Abhinav Gupta. Beyond Grids: Learning graph representations for visual recognition. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 9225–9235, 2018. 2, 3

[21] Yikang Li, Wanli Ouyang, Zhou Bolei, Shi Jianping, Zhang Chao, and Xiaogang Wang. Factorizable Net: An efficient subgraph-based framework for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[22] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[23] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 1853–1863, 2018. 2, 3, 4

[24] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 1, 6

[25] Yanni Ma, Yulan Guo, Hao Liu, Yinjie Lei, and Gongjian Wen. Global context reasoning for semantic segmentation of 3D point clouds. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020. 6

[26] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 2

[27] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Proceedings of the Advances in Neu-*

*ral Information Processing Systems (NeurIPS)*, pages 2171–2180, 2017. 2

[28] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 2, 3, 5, 6, 8

[29] Charles R. Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5648–5656, 2016. 2

[30] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017. 2

[31] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(6):1137–1149, 2017. 2

[32] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[33] Weijing Shi and Raj Rajkumar. Point-GNN: Graph neural network for 3D object detection in a point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[34] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 7

[35] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, 2015. 2, 7, 8

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 8

[37] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D object instance relocalization in changing indoor environments. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 5

[38] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 5, 6

[39] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X. Chang, and Daniel Ritchie. PlanIT: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)*, 38(4), 2019. 2, 7

[40] Kai Wang, Manolis Savva, Angel X. Chang, and Daniel Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)*, 37(4), 2018. 2, 7

[41] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep Graph Library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019. 8

[42] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[43] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. 2, 3, 5

[44] Tianyi Wu, Yu Lu, Yu Zhu, Chuang Zhang, Ming Wu, Zhanyu Ma, and Guodong Guo. GINet: Graph interaction network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 5

[45] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.*, 9:513–530, 2018. 8

[46] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 6

[47] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[48] Bin Yang, Wenjie Luo, and Raquel Urtasun. PIXOR: Real-time 3D object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7652–7660, 2018. 1, 2

[49] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018. 1, 2, 6

[50] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 48, pages 40–48, New York, New York, USA, 2016. PMLR. 8

[51] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural Motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[52] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7151–7160, 2018. 5

[53] Songyang Zhang, Xuming He, and Shipeng Yan. Latent-GNN: Learning efficient non-local relations for visual recognition. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97, pages 7374–7383, Long Beach, California, USA, 2019. PMLR. 2

[54] Lin Zhao and Wenbing Tao. JSNet: Joint instance and semantic segmentation of 3D point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2

[55] Yang Zhou, Zachary While, and Evangelos Kalogerakis. SceneGraphNet: Neural message passing for 3D indoor scene augmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 4, 7, 8