ESGNN: TOWARDS EQUIVARIANT SCENE GRAPH NEURAL NETWORK FOR 3D SCENE UNDERSTANDING

Quang P.M. Pham, Khoi T.N. Nguyen, Lan C. Ngo, Truong Do, Truong Son Hy*

College of Engineering and Computer Science, VinUniversity, Vietnam
Department of Computer Science, University of Alabama at Birmingham, United States

VINUNIVERSITY
College of Engineering and Computer Science

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

Abstract

Scene graphs have been proven to be useful for various scene understanding tasks due to their compact and explicit nature. However, existing approaches often neglect the importance of maintaining the symmetry-preserving property when generating scene graphs from 3D point clouds. This oversight can diminish the accuracy and robustness of the resulting scene graphs, especially when handling noisy, multi-view 3D data. This work, to the best of our knowledge, is the first to implement an Equivariant Graph Neural Network in semantic scene graph generation from 3D point clouds for scene understanding. Our proposed method, ESGNN, outperforms existing state-of-the-art approaches, demonstrating a significant improvement in scene estimation with faster convergence. ESGNN demands low computational resources and is easy to implement from available frameworks, paving the way for real-time applications such as robotics and computer vision.

Keywords – Scene graph, Scene understanding, Point clouds, Equivariant neural network, and Semantic segmentation.

Introduction

Recent advancements in scene graph generation have transitioned from solely utilizing 2D image sequences to incorporating 3D features such as depth camera data and point clouds, with the latest approaches, like [1]–[3], leveraging both 2D and 3D information for improved representation. However, these methods overlook the symmetry-preserving property of GNNs, which potentially cause scene graphs' inconsistency, being sensitive to noisy and multi-view data such as 3D point clouds. Hence, this work adopts E(n) Equivariant Graph Neural Network [4]'s Convolution Layers with Feature-wise Attention mechanism [3] to create **E**quivariant **S**cene **G**raph **N**eural **N**etwork (ESGNN). This approach ensures that the resulting scene graph remains unaffected by rotations and translations, thereby enhancing its representation quality.

Additionally, ESGNN requires fewer layers and computing resources compared to Scene Graph Fusion (SGFN) [3], while achieving higher accuracy scores with fewer training steps. In summary, our contributions include:

- We, to the best of our knowledge, are the first to implement Equivariant GNN in generating semantic scene graphs from 3D point clouds for scene understanding.
- Our method, named ESGNN, outperforms state-of-the-art methods, achieving better accuracy scores with fewer training steps.
- We demonstrate that ESGNN is adaptive to different scene graph generation methods. Furthermore, there is significant potential to explore the integration of equivariant GNNs for scene graph representation, with considerable room for future improvement.

Overall Framework

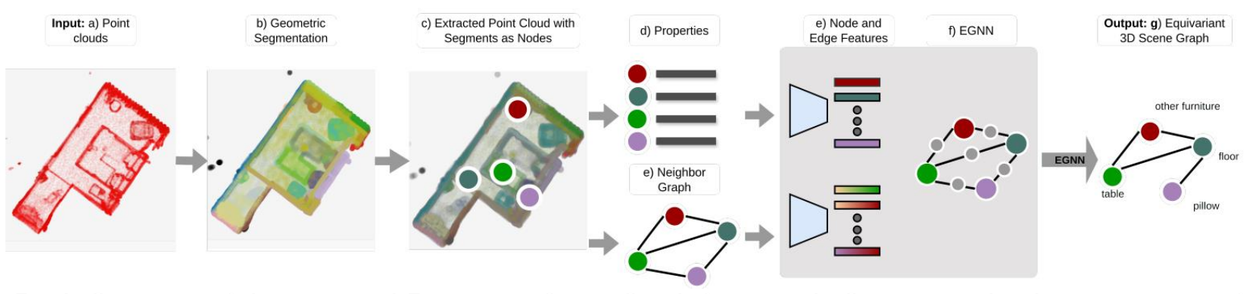


Fig. 1: Overview of the proposed Equivariant Scene Graph framework. Our approach takes a sequence of point clouds a) as input to generate a geometric segmentation b). Subsequently, the properties of each segment and a neighbor graph between segments are constructed. The properties d) and neighbor graph e) of the segments that have been updated in the current frame c) are used as the inputs to compute node and edge features f) and to predict a 3D scene graph g).

Given the 3D scene data D_i and D_j that represent the same point cloud of a scene but from different views (rotation and transition), we aim to predict the probability distribution of the equivariant scene graph prediction in which the equivariance is preserved:

$$\begin{cases} P(G|D_i) = P(G|D_j)_{i \neq j} \\ D_j = R_{i \to j} D_i + T_{i \to j} \end{cases}$$

where G is the scene graph, $R_{i \to j}$ is the rotation matrix and $T_{i \to j}$ is the transition matrix.

A. Graph Initialization

- *Node features:* The node feature includes the invariant features $\boldsymbol{h_i}$ and vector coordinate $\boldsymbol{x_i} \in R^3$. $\boldsymbol{h_i}$ consists of the latent feature of the point cloud after going through the PointNet [6] $f_p(\boldsymbol{P_i})$, standard deviation σ_i, log of the bounding box size $\ln(b_i)$, log of the bounding box volume $\ln(v_i)$, and log of the maximum length of bounding box $\ln(l_i)$. The coordinate vector of the bounding box $\boldsymbol{x_i}$ is defined by the coordinate of the two furthest corners of the bound box. $\boldsymbol{h_i}$ and $\boldsymbol{x_i}$ are then fed to the MLP for predicting the label of the nodes. Mathematically:

$$\begin{aligned} \boldsymbol{v_i} &= (\boldsymbol{h}_i, \boldsymbol{x}_i) \\ \boldsymbol{h}_i &= \left[f_p(\boldsymbol{P}_i), \boldsymbol{\sigma_i}, \ln(\boldsymbol{b}_i), \ln(\boldsymbol{v}_i), \ln(l_i)\right] \\ \boldsymbol{x}_i &= \left[\boldsymbol{x}_i^{bottomright}, \boldsymbol{x}_i^{topleft}\right] \\ \boldsymbol{c}_i^{node} &= g_v(\boldsymbol{v}_i) \end{aligned}$$

- *Edge features:* For an edge between a source node i and a target node j where $j \neq i$, the edge visual feature $c_{i \to j}^{edge}$ is computed as follows:

$$\begin{aligned} \boldsymbol{r}_{ij} &= \left[\overline{\boldsymbol{p}}_{\boldsymbol{i}} - \overline{\boldsymbol{p}}_{\boldsymbol{j}}, \boldsymbol{\sigma_i} - \boldsymbol{\sigma_j}, \boldsymbol{b_i} - \boldsymbol{b_j}, \ln\left(\frac{l_i}{l_j}\right), \ln\left(\frac{v_i}{v_j}\right)\right], \\ \boldsymbol{c}_{i \to j}^{edge} &= g_e(\boldsymbol{r}_{ij}) \end{aligned}$$

where $g_v(\cdot), g_e(\cdot)$ are MLP classifiers that project the properties into a latent space.

B. Equivariant Scene Graph Neural Network (ESGNN):

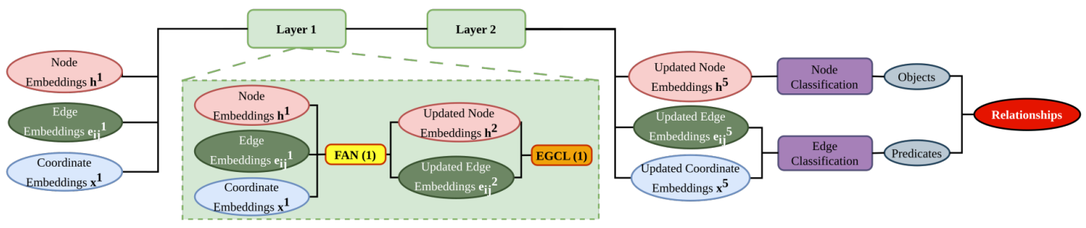


Fig. II: ESGNN Architecture.

Our GNN network, ESGNN, has two main components: (1) Feature-wise attention Graph Convolution Layer (FAN-GCL) and (2) Equivariant Graph Convolution Layer (EGCL). ESGNN is constructed with 4 message passing layers, consisting of 2 levels of FAN-GCL followed by the EGCL. Our model architecture is illustrated in Figure 2, and the formula used to update node and edge features $(v_i^{\ell}, e_{ij}^{\ell})$ of FAN as well as the EGCL is as follows:

- Message passing FAN-GCL:

$$\begin{aligned} \boldsymbol{v}_i^{l+1} &= g_v\left(\left[\boldsymbol{v}_i^l, \max_{j \in N(i)} FAN(\boldsymbol{v}_i^l, \boldsymbol{e}_{ij}^l, \boldsymbol{v}_j^l)\right]\right) \\ \boldsymbol{e}_{ij}^{l+1} &= g_e\left(\left[\boldsymbol{v}_i^l, \boldsymbol{e}_{ij}^l, \boldsymbol{v}_j^l\right]\right) \end{aligned}$$

- Message passing EGCL:

$$\begin{aligned} \boldsymbol{h}_i^{l+1} &= \boldsymbol{h}_i^l + g_v\left(concat\left(\boldsymbol{h}_i^l, \sum_{j \in N(i)} \boldsymbol{e}_{ij}^l\right)\right) \\ \boldsymbol{e}_{ij}^{l+1} &= g_e(concat(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \left\|\boldsymbol{x}_i^l - \boldsymbol{x}_j^l\right\|_2^2, \boldsymbol{e}_{ij}^l \\ \boldsymbol{x}_i^{l+1} &= \boldsymbol{x}_i^l + \sum_{j \in N(i)} (\boldsymbol{x}_i^l - \boldsymbol{x}_j^l) \cdot \phi_{coord}(\boldsymbol{e}_{ij}^l) \end{aligned}$$

Experiments

A. Dataset and Metrics

The 3DSSG dataset, used for scene graph generation, is built upon the 3RScan dataset, a large-scale real-world dataset featuring 1482 3D reconstructions of 478 indoor environments. The 3RScan dataset [5] is processed with ScanNet [7] for geometric segmentation. For the experiment, the test set of the **I20** version is primarily used, which includes 20 objects and 8 predicates. Given the dataset's imbalance, recall of nodes and edges is used as the evaluation metric. During training, recall is calculated as the true positive over all positive predictions. Additionally, the **R@x** metric is adopted for more detailed analysis, taking the x most confident predictions and marking them as correct if at least one prediction is correct. Recall metrics are applied to *predicate* (edge classification), *object* (node classification), and relationship *(triplet <subject, predicate, object>)*.

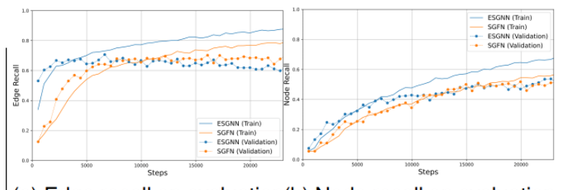
B. Results

TABLE I: Scene graph predictions for relationship triplet, object, and predicate, measured on 3DSSG-l20. The *Recall* column reports the recall scores on objects (*Obj.*) and relationships (*Rel.*)

Method	Relationship		Object		Predicate		Recall	
	R@1	R@3	R@1	R@3	R@1	R@2	Obj.	Rel.
3DSSG	32.65	50.56	55.74	83.89	**95.22**	98.29	55.74	**95.22**
SGFN	37.82	48.74	62.82	**88.08**	81.41	98.22	63.98	94.24
ESGNN	**43.54**	**53.64**	**63.94**	86.65	94.62	**98.30**	**65.45**	94.62

TABLE II: Scene graph predictions for new unseen relationship triplet, object, and predicate, measured on 3DSSG-l20 with geometric segmentation.

Method	New Relationship		New Object		New Predicate	
	R@1	R@3	R@1	R@3	R@1	R@2
3DSSG	39.74	49.79	55.89	84.42	**70.87**	83.29
SGFN	**47.01**	55.30	64.50	**88.92**	68.71	**83.76**
ESGNN (Ours)	46.85	**56.95**	**65.47**	87.52	66.90	82.88



(a) Edge recall on evaluation set 3DSSG-**I20** (b) Node recall on evaluation set 3DSSG-**I20**

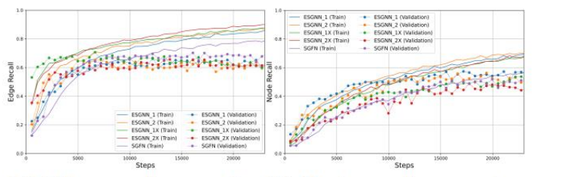
Fig. 3: Comparison of ESGNN with SGFN through the training steps.

ESGNN is shown to converge more quickly in the early training epochs and achieve competitive performance throughout further epochs.

C. Ablation Study

TABLE III: Evaluation of different ESGNN architectures on scene graph generation task on 3DSSG-l20 dataset. ② is our best performer and is used for the evaluation in Section IV-B

Method	Relationship		Object		Predicate	
	R@1	R@3	R@1	R@3	R@1	R@2
① SGFN	37.82	48.74	62.82	**88.08**	81.41	98.22
② ESGNN_1	**42.30**	**53.30**	**63.21**	86.70	94.34	**98.30**
③ ESGNN_2	35.63	44.63	57.55	84.41	93.93	97.94
④ ESGNN_1X	34.96	42.59	57.55	86.18	92.68	98.08
⑤ ESGNN_2X	37.94	50.58	59.97	85.23	**94.53**	98.01

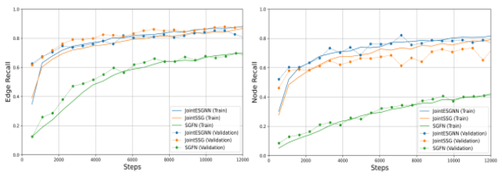


(a) Edge recall on evaluation set 3DSSG-**I20** (b) Node recall on evaluation set 3DSSG-**I20**

Fig. 4: Comparison of multiple ESGNN models with SGFN through the training steps.

(1) is the SGFN, run as the baseline model for comparison. (2) is the ESGNN with a single FAN layer and an EGCL layer. (3) is with 2 FAN layers and 2 layers EGCL. (4) is similar to (1) but concatenating coordinate embedding to the output edge embedding. We expect this modification to improve the performance of edge prediction.(5) is similar with 2 layers of FAN and EGCL.

D.ESGNN with Image Encoder



(a) Edge recall on evaluation set 3DSSG-**I20** (b) Node recall on evaluation set 3DSSG-**I20**

Fig. 5: Comparison of Joint-ESGNN, SGFN, JointSSG through the training steps.

Conclusion

We introduced the Equivariant Scene Graph Neural Network (ESGNN), which improves the robustness and accuracy of generating semantic scene graphs by leveraging the E(n) Equivariant Graph Neural Network (EGNN). ESGNN outperforms state-of-the-art methods with fewer layers and reduced computational resources.

thy@uab.edu

2024 IEEE International Symposium on Robot and Human Interactive Communication, Los Angeles, US, 26-30 August 2024