DOI: 10.1111/cgf.14658
Pacific Graphics 2022
N. Umetani, E. Vouga, and C. Wojtan
(Guest Editors)

Volume 41 (2022), Number 7

Exploring Contextual Relationships in 3D Cloud Points by Semantic Knowledge Mining

Lianggangxu Chen,[1] Jiale Lu,[1] Yiqing Cai,[1] Changbo Wang[1,†] and Gaoqi He[2,†]

[1]School of Computer Science and Technology, East China Normal University, Shanghai, China
[2] School of Computer Science and Technology, Shanghai Key Laboratory of Mental Health and Psychological Crisis Intervention, East China Normal University, Shanghai, China.

Abstract

3D scene graph generation (SGG) aims to predict the class of objects and predicates simultaneously in one 3D point cloud scene with instance segmentation. Since the underlying semantic of 3D point clouds is spatial information, recent ideas of the 3D SGG task usually face difficulties in understanding global contextual semantic relationships and neglect the intrinsic 3D visual structures. To build the global scope of semantic relationships, we first propose two types of Semantic Clue (SC) from entity level and path level, respectively. SC can be extracted from the training set and modeled as the co-occurrence probability between entities. Then a novel Semantic Clue aware Graph Convolution Network (SC-GCN) is designed to explicitly model each SC of which the message is passed in their specific neighbor pattern. For constructing the interactions between the 3D visual and semantic modalities, a visual-language transformer (VLT) module is proposed to jointly learn the correlation between 3D visual features and class label embeddings. Systematic experiments on the 3D semantic scene graph (3DSSG) dataset show that our full method achieves state-of-the-art performance.

CCS Concepts

• ***Computing methodologies*** → *3D point cloud understanding; Graph convolution network;*

1. Introduction

A scene graph (SG) not only records the locations and classes of objects in a scene, it also represents pairwise visual relationships of objects in the triple structure of a <**subject-predicate-object**>, abbreviated as (s, p, o). The accurate understanding of SGG plays an important role in many computer vision tasks, such as image generation [JGFF18, XZH*18, MAA*19, HBX*20], visual question answering [BYCCT17, ZCX19, TZW*19, RPS21] and image captioning [LOZ*17, YPLM18, YTZC19, ZWC*20]. Therefore, SGG is treated as a potential task for bridging the huge gap between vision and natural language domains.

3D scene graph generation has recently interested researchers, benefiting from the development of Graph Convolution Network (GCN) techniques [KW17] and proposals of 3D SG datasets [WDNT20, WAN*19]. In particular, there are two groups of methods in solving this challenging task now, *i.e.*, contextual relationship learning and prior knowledge embedding. Representative works of the former group [WDNT20, ZYSC21, WWT*21] tend to tackle this task by finding contextual relationships between different objects through GCN. The latter group of studies [ZHQ*21] propose to embed the semantic information by encoding the class label. The core idea of latter group is to fuse features from visual space and label space, that can predict correct unknown predicate p from $(s, ?, o)$, by changing a probability distribution from a low score negative triple (s, p, o) into a high score positive $(s, p, o)'$.

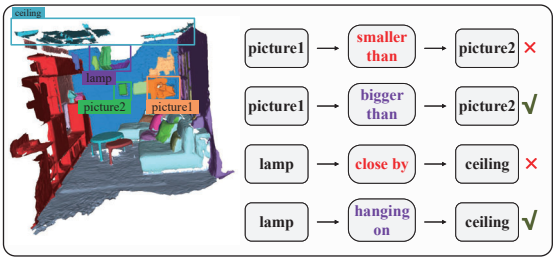

Figure 1: *An example of misclassification results by knowledge-based model [ZHQ*21]. Red represents the wrong result and purple represents the correct result.*

Different from images where RGB pixels are stored in the regular grid, for point clouds, the underlying semantic and structural information of point clouds is the spatial layout of the points. Conse-

† Changbo Wang and Gaoqi He are the corresponding authors

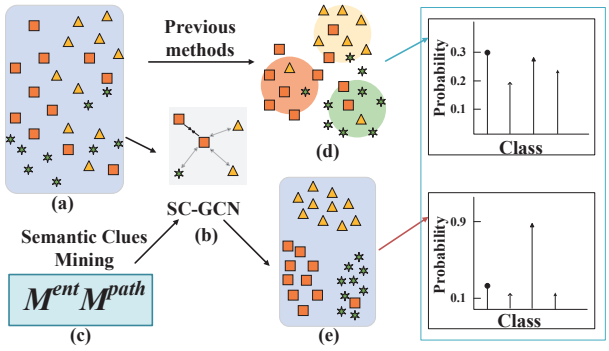

Figure 2: *Current classification models learn to map input point clouds to an embedding space (a) while ignoring intrinsic semantic relationships of labeled data. SC-GCN is introduced to foster a new training paradigm (b), by explicitly mining entity level SC and path level SC. As shown in (b), each sample embedding feature is pulled closer to the same class but pushed far from other classes. In (d), the previous methods [WDNT20] can only get a small inter-class variance. A better-structured embedding space (e) is derived by our method, eventually boosting the performance of the classification models.*

quently, existing point cloud analysis usually face difficulties in understanding the information of language modalities. Besides, compared with structural images, irregular point clouds cannot benefit from the semantic context of neighbors by CNN. Figure 1 shows a typical error scenario. Due to the inability to accurately understand the semantic relationships between entities, the model incorrectly classifies tail predicates (<**lamp-hanging on-ceiling**> to <**lamp-close by-ceiling**>). Meanwhile, the comparative predicates between the same object class are predicted to the exact opposite meanings (<**picture-bigger than-picture**> to <**picture-smaller than-picture**>).

Figure 2 provides an in-depth analysis of the impact of the lack of global contextual semantic relationship from a feature perspective. For the 3D scene graph generation task with cloud points, the deeply learned features must be not only separable but also discriminative (Figure 2(e)). Point clouds that are extracted from indoor scenes may suffer from sparseness, lack of contextual semantic information, background noise, and inaccurate segmentation of point cloud objects. As a result, there is a large intra-class variance and a small inter-class variance in the real-world point cloud dataset (Figure 2(b)). Hence, it is difficult to distinguish samples for softmax classifiers.

In this paper, we emphasize that there are two critical limitations in 3D SGG: 1) Recent works usually face difficulties understanding global contextual semantic knowledge, which leads to poor recall of tail predicates; 2) 3D visual structure is lost when propagating semantic knowledge, resulting in inaccurate comparative predicate predictions.

We propose a ternary learning framework to jointly model the 3D visual structures and semantic knowledge. As shown in Figure 3, the ternary learning framework consists of three feature inputs (3D visual-level features, semantic clues and class label embeddings). First, the normalized 3D cloud points are encoded by PointNet [QSMG17] to extract 3D visual-level features (Figure 3(a)). After that, to fully understand contextual representations of semantic knowledge, two types of semantic clues are extracted from the train set. For entity level SC, it is measured by the co-occurrence of different entities in the training set; for path level SC, it is the path connections from one entity to another entity in the training set. Each SC is modeled explicitly by a novel Semantic Clue aware Graph Convolution Network (Figure 3(b)). The multi-layer message passing mechanism of SC-GCN dynamic updates semantic clues and visual-level features, which contributes to obtaining a more powerful semantic knowledge representation. Finally, to construct a relation between the language modality and the 3D visual modality, we propose a visual-language transformer module to enhance the final prediction features by calculating the similarity of class label embeddings and 3D visual features (Figure 3(c)).

The main contributions of this paper can be summarized as follows:

- To overcome limitation 1, entity level SC and path level SC are proposed that are capable of characterizing the context information of predicates and the strength of the entity connection in finer detail, respectively. Correspondingly, a novel SC-GCN is designed to explicitly model each SC of which the message is passed in their specific neighbor pattern.
- A visual-language transformer is proposed to tackle limitation 2. It jointly models semantic and visual features by learning a similarity matrix. The cross-modal attention mechanism ensures that the visual structure information is not lost when propagating semantic knowledge.
- Comprehensive experiments are conducted on the 3DSSG dataset with the comparison of previous studies. The results demonstrate the proposed method can alleviate the problem of incorrect classification of tail predicates and also works well for the comparative predicates between the same object class.

2. Related work

2.1. 3D scene graph generation based on point clouds

3D scene graph generation based on point clouds has first been proposed in [WDNT20] and caught increasing attention in computer vision community. The applications for downstream tasks include robot task planning [AJK*22, RPH*21, ZTBZ21], mechanical search [KMMI*21], augmented reality [TSNI20], 3D scene generation [DMNT21], scene retrieval [WDNT20] and 3D panoptic/semantic segmentation [WWT*21].

To have a more comprehensive understanding of 3D scenes, Wald *et al.* [WDNT20] proposed the first learned method that generates a 3D scene graph from a 3D point cloud by a GCN. However, the methods proposed in [AHG*19] and [WDNT20] only predict the scene graph in an offline manner. Wu *et al.* [WWT*21] built up 3D scene graphs online by aggregating context features in a graph neural network. Moreover, the performance of these methods depends heavily on whether accurate inter-object structural relationships can be obtained or not, losing sight of the visual cues lurking inside each edge. Zhang *et al.* [ZYSC21] created an

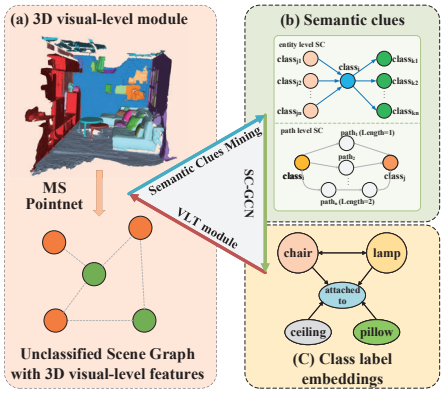

Figure 3: *A ternary learning framework is proposed in our motivation. For 3D visual modality, visual structures (a) are extracted by MS Pointnet [ZHQ*21]. For language modality, semantic clues (b) are extracted from the training set and word embeddings are generated from Bert [DCLT18] as class label embeddings (c). The SC-GCN is used to merge visual features and semantic clues. Finally, the VLT module fuses language knowledge and visual features to obtain more robust prediction features.*

EDGE-oriented Graph Convolutional Network (EdgeGCN) to exploit multi-dimensional edge features for explicit relationship modeling. Although promising results have been obtained, the long-tailed effect in the real 3D scenes is really complex. To remedy this, Zhang *et al.* [ZHQ*21] effectively enhanced the accuracy of tail classes predictions by incorporating the one-hot class label embeddings with perceptual information.

Although these methods leverage context relationships and embed semantic representations, they have the limitation of capturing the relation between the language modality and the 3D visual modality. Hence, different from [ZHQ*21], our model formulizes the co-occurrence information and explicitly incorporates them by visual-language transformer to help scene graph generation. Therefore, our model can predict more accurate tail predicates and comparative predicates.

2.2. Knowledge representation in scene graph generation

Knowledge representation in the scene graph is the ontological knowledge embedding of entity classes and co-occurrence probability, independent of scene-specific features. Embedding prior knowledge has been proven as an effective method of alleviating the long-tailed effect problem, which is an active research area. There are two main aspects in previous works making efforts to incorporate Knowledge representation based on the source of knowledge, i.e., knowledge from the datasets and knowledge from learning features.

Knowledge from the datasets: To integrate richer types of knowledge, external facts are leveraged from lexical databases such as WordNet [Mil95], knowledge bases such as ConceptNet [LS04] and the training set. Chen *et al.* [CYCL19] proposed to utilize co-occurrence statistics of triples from the training set as commonsense, which addresses the unbalanced distribution issue. To achieve a better generalization, Hou *et al.* [HWQ*19] then took knowledge graph embeddings that aggregate the representation of each component of a triplet as commonsense knowledge. Since these methods failed to exploit the graphical structure of commonsense knowledge, Zareian *et al.* [ZKC20] used a commonsense knowledge graph where each node represents an entity or predicate class and each edge states the interaction probability of two concepts as commonsense knowledge.

Knowledge from learning features: Zareian *et al.* [ZWYC20] proposed the first method to acquire visual knowledge automatically from data, and use that to improve the robustness of SGG. More recently, unlike previous works, Sharifzadeh *et al.* [SBT21] did not consider separate models for perception and prior knowledge. They entangled the perception and prior in a single model with shared parameters trained by multi-task learning.

However, these methods all have a flaw in that the knowledge input is incomplete and fail to incorporate 3D visual structure information. By contrast, we extract structured and explicit semantic knowledge from the dataset. Our method dynamically updates semantic knowledge with the 3D visual structure to build a more robust feature representation.

2.3. Vision-language transformer

Inspired by the success of transformer in natural language processing tasks, researchers attempt to use transformer architecture to capture global contextual information for computer vision tasks [UBH*22, MKK21, LYY*19, RKH*21, CZG*22].

Carion *et al.* [CMS*20] designed a object DEtection TRansformer (DETR), which shows performance on object detection. Afterwards, Zhu *et al.* [ZSL*20] introduced a attention module to solve the poor performance on small objects of DETR. In the field of text-to-image generation, Ramesh *et al.* [RPG*21] described a simple approach for this task based on a transformer that autoregressively models the text and image tokens as a single stream of data. Zheng *et al.* [ZLZ*21] employed SEgmentation TRansformer (SETR) in semantic segmentation tasks and achieve impressive performance. For image-based SGG, Zhong *et al.* [ZSY*21] designed a Transformer-based model to create "pseudo" labels for learning scene graph via a masked token prediction task.

However, there are no efforts yet to introduce transformer architecture into 3D SGG tasks. In this paper, we proposed a visual-language transformer that jointly models semantic and visual features by learning a similarity matrix. The cross-modal attention mechanism ensures that the visual structure information is not lost when propagating semantic knowledge.

3. Method

3.1. Overview

In this paper, we solve the 3D SGG task by propagating semantic clues $\mathcal{S}$ and incorporating class label embeddings $\mathcal{C}$. Our overall framework is illustrated in Figure 4. Given the point set $\mathcal{P}$ of a scene s and the class-agnostic instance segmentation $\mathcal{M}$, the task

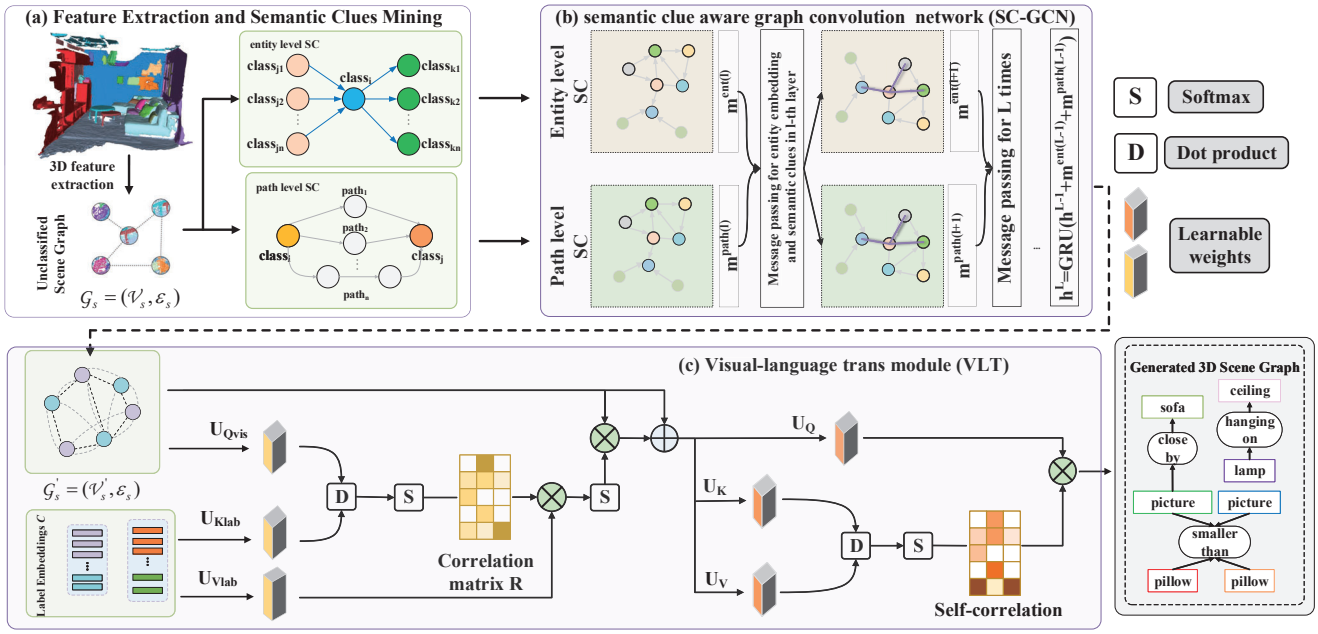

Figure 4: *An overview of our full method. (a) 3D visual features are first extracted by MS Pointnet. (b) Next, the SC-GCN module connects the visual entity node and propagates the semantic clues using an SC aware message passing method. (c) After that, the visual-language transformer module enhances the final prediction features by calculating the similarity of class label embeddings and 3D visual features. Finally, the 3D scene graph is obtained on the right side.*

of 3D SGG first parses the input $\mathcal{P}$ into an unclassified scene graph $\mathcal{G}_s = \{\mathcal{V}_s = (\mathcal{V}_o \cup \mathcal{V}_p), \mathcal{E}_s\}$, where $\mathcal{V}_o = \left\{\mathbf{v}_o^i\right\}_{i=1}^{N}, \mathbf{v}_o^i$ denotes the feature of object node and $\mathcal{V}_p = \{\mathbf{v}_p^i\}_{i=1}^{M}, \mathbf{v}_p^i$ denotes the feature of predicate node. Here N is the number of objects and M represents the number of predicates between objects. There are two types of undirected edges in $\mathcal{E}_s$, where Each edge connects a predicate node to its corresponding subject node or object node. To establish such a graph, a 3D visual feature extractor is first used to extract a set of objects feature vectors $\mathcal{V}_o$. Thereafter, the 3D feature extractor is adopted to generate visual features $\mathcal{V}_p$ for predicates by concatenating the object features with the center coordinates of any two object point sets [XZCFF17].

Now, we get an unclassified scene graph $\mathcal{G}_s$ defined in Section 3.1 (Figure 4(a)). We feed the $\mathcal{G}_s$ and $\mathcal{S}$ into the SC-GCN module, then output an updated 3D scene graph $\mathcal{G}_s'$ by propagating semantic clues $\mathcal{S}$ (Figure 4(b)). Finally, we feed the $\mathcal{G}_s'$ and $\mathcal{C}$ into the VLT module (Figure 4(c)), which augments the visual appearance with explicit class label embeddings to generate refined object and predicate features for final label prediction.

To this end, the 3D SGG task is formulated by the following probability function $P(\mathcal{G} \mid \mathcal{P}, \mathcal{M}, \mathcal{S}, \mathcal{C})$, which can be decomposed into four factors:

$$P(\mathcal{G} \mid \mathcal{P}, \mathcal{M}, \mathcal{S}, \mathcal{C}) = P(\mathcal{G}_s \mid \mathcal{P}, \mathcal{M}) P(\mathcal{G}_s' \mid \mathcal{G}_s, \mathcal{S}) \times P(\mathcal{V}_o \mid \mathcal{G}_s', \mathcal{C}) P(\mathcal{V}_p \mid \mathcal{G}_s', \mathcal{C}, \mathcal{V}_o), \quad (1)$$

where the factor $P(\mathcal{G}_s \mid \mathcal{P}, \mathcal{M})$ means to propose a unclassified scene graph $\mathcal{G}_s$ by a fixed feature extractor. The factor $P(\mathcal{G}_s' \mid \mathcal{G}_s, \mathcal{S})$ relies on the proposed SC-GCN module to propagate semantic clues in dataset (Section 3.2). The object factor $P(\mathcal{V}_o \mid \mathcal{G}_s', \mathcal{C})$ and the predicate factor $P(\mathcal{V}_p \mid \mathcal{G}_s', \mathcal{C}, \mathcal{V}_o)$ are realized by the proposed VLT module for incorporating class label embeddings $\mathcal{C}$ (Section 3.3) and inferring the corresponding class label (Section 3.4).

3.2. Semantic clue aware graph convolution network

In this subsection, the nodes feature in unclassified scene graph $\mathcal{G}_s$ is updated by propagating semantic clues $\mathcal{C}$. Firstly, we attempt to mine the proposed semantic clues from the training set. Next, the semantic clues are performed as a co-occurrence matrix to update the node representations in $\mathcal{G}_s$ by a novel message passing scheme in SC-GCN. After optimization through our designed update scheme, the statistical information from the dataset could be injected into the 3D scene graph. The details are described as follows:

Step 1: Semantic Clues Mining. We attempt to mine the proposed semantic clues from the training set. Noted that the object and predicate are considered to be the same entity, different from previous work [CYCL19], we directly obtain the probabilistic dependencies between entities. For a prediction query $(s, ?, o) \rightarrow p$ in the testing set, two corresponding semantic clues are proposed:

- 1. S_{ent} for entity level SC: It is the number of triples (c_i, c_j, c_k) in train set that given definite entity class c_i and c_k, where c_i, c_j and $c_k \in C_o \cup C_p$. Instantly, for triple (lamp, ?, ceiling), the

probability of predicting predicate **hanging on** should be higher than predicting **close by**.

More specifically, for two entity classes of c_i and c_k, the $c_i \to c_k$ dependence is modeled in the form of conditional probability, i.e., $p\left(c_j \mid c_i, c_k\right)$. We count the occurrence of class c_j in the presence of class c_i and c_k, and obtain the co-occurrence matrix $\mathcal{M}^{ent}_{i,j,k} \in R^{(|C_o|+|C_p|)\times(|C_o|+|C_p|)\times(|C_o|+|C_p|)}$, where $i,j,k \in \{1,2,...,(|C_o|+|C_p|)\}$. Then, the co-occurrence probabilities matrix $\mathcal{M}^{ent}$ is obtained by normalizing $\mathcal{M}^{ent}_{i,j,k} = \mathcal{M}^{ent}_{i,q,k} / \sum_{i \neq k} \mathcal{M}^{ent}_{i,q,k}, q = 1,2,...,j,...,(|C_o|+|C_p|)$.

- 2. S_{path} for path level SC: It is the number of path from entity class c_i to c_k in the training set. If there are two triples $\left(c_i, c_j, c_k\right)$ and (c_p, c_l, c_t) in the training set, then it can be regarded as a path from c_i to c_t. Triple $(c_i, \textbf{same material}, c_k)$ and triple $(c_k, \textbf{same material}, c_j)$ will bring confidence for predicting triple $(c_i, \textbf{same material}, c_j)$. Under the graph view, this can be regarded as the path from c_i to c_j.

  The path length is limited to less than or equal to 2. Similarly, the final S_{path} is obtained by normalizing $\mathcal{M}^{path}_{i,j} = \mathcal{M}^{path}_{i,j} / \sum_{j \neq i} \mathcal{M}^{path}_{i,j}$.

Step 2: Initial Node States. Given an unclassified scene graph $\mathcal{G}_s$ in Section 3.1, each node is associated with an initial node embedding, namely $\mathbf{h}_i$. We use a fully connected (FC) layer $R(\cdot)$ to map the $\mathcal{V}_s$ to the initial node embeddings in $\mathcal{G}_s$:

$$\mathbf{h}_i^{(0)} = R(\mathbf{v}_i), i \in \{1, \cdots, |\mathcal{V}_s|\}. \tag{2}$$

After that, $\mathbf{h}_i^{(0)}$ is duplicated $(|C_o|+|C_p|)$ times to obtain $(|C_o|+|C_p|)$ nodes $\left\{\mathbf{h}_{i1}^{(0)}, \mathbf{h}_{i2}^{(0)}, \ldots, \mathbf{h}_{i(|C_o|+|C_p|)}^{(0)}\right\}$, where node $\mathbf{h}_{ic}^{(0)}$ denotes the correlation of object $\mathbf{h}_i$ with class c.

Step 3: Node Updating by Semantic Clue Aware Message Passing. Knowing from previous studies on image SGG [CYCL19, ZKC20], co-occurrence probabilities are important to design models with powerful generalization ability. However, for most current works, they capture the co-occurrence probabilities mainly in an implicit and insufficient way, which limits their performance. In this section, to make better use of two novel types of co-occurrence probabilities (named semantic clues), a GCN-based model called semantic clue aware graph convolution network is proposed.

Specifically, for entity level SC, it describes the triple similarity from the perspective of neighborhood structure, where both neighborhoods $\mathbf{h}_j$ and $\mathbf{h}_k$ of each node $\mathbf{h}_i$ should be considered. We compute the aggregated incoming messages $\mathbf{m}^{ent(l)}_{i(c_i)}$ of entity level SC for $\mathbf{h}_i$ by the following:

$$\mathbf{m}^{ent(l)}_{i(c_i)} = \sigma\left(\sum_{(\mathbf{h}_j,\mathbf{h}_k)\in\mathcal{N}_i} \sum_{c_j=1}^{|C_o|+|C_p|} \sum_{c_k=1}^{|C_o|+|C_p|} \alpha^{ent}_{jk} \mathbf{\Lambda}\right), \quad \text{where} \quad \mathbf{\Lambda} = W^{ent} \phi\left(\mathbf{h}^{(l-1)}_{j(c_j)}, \mathbf{h}^{(l-1)}_{k(c_k)}\right). \tag{3}$$

$\phi\left(\mathbf{h}_j, \mathbf{h}_k\right) = \sigma\left(\left[\mathbf{h}_j \| W_1 \mathbf{h}_k\right]\right)$ is the convert function to fuse the neighborhood node $\mathbf{h}_j$ and $\mathbf{h}_k$ information. The subscript $c_{\{i,j,k\}}$ represents the channel of the $c_{\{i,j,k\}}$-th class, where $c_{\{i,j,k\}} \in \{1,2,...,(|C_o|+|C_p|)\}$. W_1 is the learnable weight matrix and $\|$ denotes a concatenation. $\mathcal{N}_i$ denotes node $\mathbf{h}_i$'s neighbor nodes. W^{ent} is linear transformation matrix. In addition, σ indicates LeakyReLU [MHN*13] function and l is the layer number in GCN. α^{ent}_{jk} is aggregation attention, which is computed as:

$$\alpha^{ent}_{jk} = \frac{\exp\left(\varphi\left(\mathbf{h}_{j(c_j)}, \mathcal{M}^{ent}_{i,j,k}\right)^{\mathrm{T}} \mathbf{h}_{i(c_i)}\right)}{\sum_{(\mathbf{h}_j,\mathbf{h}_k)\in\mathcal{N}_i} \exp\left(\varphi\left(\mathbf{h}_{k(c_k)}, \mathcal{M}^{ent}_{i,j,k}\right)^{\mathrm{T}} \mathbf{h}_{i(c_i)}\right)}, \tag{4}$$

where $\mathcal{M}^{ent}_{i,j,k}$ denotes the entity level SC which is defined in Step 1 and $\varphi\left(\mathbf{h}_k, \mathcal{M}^{ent}_{i,j,k}\right) = \mathbf{h}_k * \mathcal{M}^{ent}_{i,j,k}$.

Similarly, for path level SC, it describes the overall entity-entity interactions. By aggregating all the connected pairs, we can get the incoming message representation $\mathbf{m}^{path(l)}_{i(c_i)}$ of path level SC as:

$$\mathbf{m}^{path(l)}_{i(c_i)} = \sigma\left(\sum_{\mathbf{h}_j\in\mathcal{N}_i} \alpha^{path}_{ij} W^{path} \mathbf{h}^{(l-1)}_{j(c_j)}\right), \tag{5}$$

where W^{path} is the linear transformation matrix. The attention weights α^{path}_{ij} is computed similarly as:

$$\alpha^{path}_{ij} = \frac{\exp\left(\varphi\left(\mathbf{h}_{i(c_j)}, \mathcal{M}^{path}_{i,j}\right)\right)}{\sum_{\mathbf{h}_p\in\mathcal{N}_i} \exp\left(\varphi\left(\mathbf{h}_{p(p_j)}, \mathcal{M}^{path}_{i,p}\right)\right)}. \tag{6}$$

After obtaining the incoming messages $\mathbf{m}^{ent(l)}_{i(c_i))}$ and $\mathbf{m}^{path(l)}_{i(c_i)}$ by message passing, we merge them with original node features and adopt the gated recurrent unit (GRU) [CVMG*14] to update the state of node i. Then the output $\mathbf{h}^{l}_{i(c_i)}$ after updating is took as the next layer's input:

$$\mathbf{h}^{l+1}_{i(c_i)} = GRU\left(\mathbf{h}^{l}_{i(c_i)} + \mathbf{m}^{ent(l)}_{i(c_i)} + \mathbf{m}^{path(l)}_{i(c_i)}\right). \tag{7}$$

Step 4: Feature Readout. After L message aggregation iterations, we take $\mathbf{h}^L_i$ as the output embedding of node i. After that, all of its class channel $(|C_o|+|C_p|)$ are merged by a transform matrix W^{out}:

$$\mathbf{v}^{\text{out}}_i = W^{\text{out}} \ \text{Concat}\left(\left\{\mathbf{h}^L_{ij} \mid j = 1,\ldots,(|C_o|+|C_p|)\right\}\right). \tag{8}$$

The output of SC-GCN is formulated as $\mathcal{V}'_s$, where $\mathcal{V}'_s = \{\mathbf{v}^{out}_i\}^{N+M}_{i=1}$. Now, we can obtain the an newly unclassified 3D scene graph $\mathcal{G}'_s = \left(\mathcal{V}'_s, \mathcal{E}_s\right)$.

3.3. Visual-language transformer module

So far, we have an isolated graph $\mathcal{G}'_s$ with visual appearance and semantic context information. In order to augment the visual appearance with explicit class label embeddings, a visual-language transformer module is proposed. Our VLT model establishes the relationship between 3D visual features and class label embeddings by computing a similarity matrix. In addition, the self-attention mechanism in the Transformer ensures that the two types of cross-modal features are decoupled from each other without causing perceptual confusion. The details are described as follows:

Step 1: Class Label Embeddings Mining. The input class label embeddings can be formulated as $\mathcal{C} = \{\mathbf{c}_i\}^{|C_o|+|C_p|}_{i=1}$, where $\mathbf{c}_i$

denotes the word embeddings of class label i, hence each class label exactly appears once. Entity-wise word embeddings are constructed by pre-trained Bert model [DCLT18]. In ablation experiments, we also compare the impact of different embedding extraction methods on the final results, e.g., Glove [PSM14] and KISGP [ZHQ*21].

Step 2: Visual-Language Cross-Attention. As a representation of language modalities, the word embeddings of labels contain specific semantic knowledge in natural language and can be used as prior support. In order to embed the class label embeddings $\mathcal{C}$ into the unclassified 3D scene graph node features $\mathcal{V}_s'$, class label embeddings are encoded into pairs of key and value maps through shared convolution layers. The class keys and class values are transformed into the 3D visual space of the query. The query and keys are utilized to compute the correlation matrix R, denoting correlation scores across visual and language modalities:

$$\mathbf{R} = \text{softmax}\left(\left(\mathcal{V}_s'\mathbf{U}_{Qvis}\right) \otimes \left(\mathcal{C}\mathbf{U}_{Klab}\right)\right), \tag{9}$$

where $\otimes$ denotes the scaled dot product operation, $\mathbf{U}_{Qvis}$ denotes the learnable weights of projection on $\mathcal{V}_s'$ for modal alignment.

With the correlation matrix $\mathbf{R}$, the value is re-weighted to reinforce the classes responded in both two modalities. To further enhance the responses of these classes in the visual modality, we adaptively fuse the re-weighted values with query values together to enhance the visual features:

$$\mathcal{V}_s^{LCA} = \text{softmax}\left(\mathbf{R}\mathbf{U}_{Vlab}\mathcal{C}\right)\mathcal{V}_s' + \mathcal{V}_s', \tag{10}$$

where $\mathcal{V}_s^{LCA}$ denotes the visual features with fusing class label embeddings. $\mathbf{U}_{Klab}$ and $\mathbf{U}_{Vlab}$ denote the learnable weights of projection on $\mathcal{C}$.

Step 3: Visual-Language Self-Attention. Besides, we exploit the self-attention mechanism on the enhanced visual features $\mathcal{V}_s^{LCA}$ across modalities to capture the discriminative and link each node to the class embeddings for better 3D visual representation:

$$\mathcal{V}_s^{P} = \text{softmax}\left(\frac{\mathcal{V}_s^{LCA}\mathbf{U}_Q\left(\mathcal{V}_s^{LCA}\mathbf{U}_K\right)^\top}{\sqrt{D_P}}\right)\mathcal{V}_s^{LCA}\mathbf{U}_V, \tag{11}$$

where $\mathbf{U}_{\{Q,K,V\}}$ denote the learnable weights of different projections on the enhanced query, key, and value across modalities. D_P denotes the hidden dimension of the Transformer. The visual feature distribution is related to the semantic information D_P, so after each element of the visual feature is divided by D_P, the variance becomes 1 again. It decouples the steepness of the distribution of visual features from D_P, so that gradient values remain stable during training. $\mathcal{V}_s^P$ is the output of the VLT module. The final 3D SGG results are predicted on the evolved node features $\mathcal{V}_s^P$. The network structure of two fully-connected layers is used to get the final classification score.

3.4. Learning objective:

Existing work on SGG tends to utilize cross-entropy loss as the objective function for entity classification [XZCFF17, ZYTC18], which considers the priority of entities are all equal. The focal loss is used for entity classification to handle this problem [LGG*17, WDNT20, ZHQ*21]. The focal loss is formulated as:

$$\mathcal{L} = \alpha(1-p)^\gamma \log(p), \tag{12}$$

where p denotes the classification score of objects and predicates on the ground-truth class for the i-th node.

4. Experiments

In this section, the performance of the ternary learning framework is evaluated on the 3D scene graph generation task with the 3DSSG dataset. We describe the detail of task description and experiment settings, compare our networks with state-of-art methods, and perform ablation studies to demonstrate the effectiveness of SC-GCN and VLT.

4.1. Task description

The task of 3D scene graph generation aims to predict the object class and predicate class based on 3D cloud points and corresponding instance segmentation input. Our model is trained and validated on the 3DSSG dataset [WDNT20] which provides 3D level scene graph labels (support, proximity, and comparative predicates) and is built upon the 3RScan dataset [WAN*19]. The training set of 3DSSG includes 3582 scenes and the testing set is composed of the remaining 548 scenes. There are 160 object classes and 27 predicate classes. Two standard tasks are followed in [ZHQ*21] for evaluation : (1) predicate classification (PREDCLS): given ground truth labels and bounding boxes of objects, predict predicate labels of object pairs. (2) scene graph classification (SGCLS): classify the ground truth bounding boxes and predict predicate labels.

4.2. Implementation details

For a fair comparison, the pre-trained MS PointNet [ZHQ*21] is adopted as our backbone. For the VLT module, the same attention settings are used in [VSP*17]. The hidden dimension is set as $D_p = 512$. For class label embeddings, we adopt 768-dim Bert [DCLT18] trained on the Wikipedia dataset. The Adam [KB14] optimizer is used with batch size 16 for 40 epochs. The initial learning rate is 0.0001 for the backbone and 0.0003 for other parts, which decays by a factor of 10 for every 5 epochs. The weight decay is set as 0.0001. All experiments are conducted on an Nvidia RTX 2080Ti GPU. We implement our approach based on PyTorch [PGC*17]. The same sub-scene split is followed in [WDNT20]. We reproduce the methods compared in this paper on the 3DSSG dataset.

4.3. Comparisons with the state-of-the-art methods

Quantitative results and comparison: Table 1 summarizes the overall 3D scene graph generation results of different methods on the 3DSSG dataset. The Recall (R) evaluation metric is used, which measures the fraction of ground truth visual triplets appearing in top-20, top-50 and top-100 confident predictions.

Our proposed approach significantly outperforms several state-of-the-art methods on the 3DSSG dataset based on the same evaluation metrics. Co-Occurrence [ZYTC18] has the worst performance

Table 1: *Comparisons with state-of-the-arts on the 3DSSG dataset. GC denotes the graph constrained results. Because the 3D scene graph generation task inputs the class-agnostic instance segmentation, we only compute the mean of the two tasks of SGCLS and PREDCLS.*

	Model	SGCLS			PREDCLS			
		R@20	R@50	R@100	R@20	R@50	R@100	Mean
w/ GC	Co-Occurrence [ZYTC18] (2018)	0.148	0.197	0.199	0.347	0.474	0.479	0.307
	KERN [CYCL19] (2019)	0.203	0.224	0.227	0.468	0.557	0.565	0.374
	SGPN [WDNT20] (2020)	0.270	0.288	0.290	0.519	0.580	0.585	0.422
	Schemata [SBT21] (2021)	0.274	0.292	0.294	0.487	0.582	0.591	0.420
	SGF [WWT*21] (2021)	0.275	0.292	0.292	0.526	0.589	0.594	0.428
	EdgeGCN [ZYSC21] (2021)	0.280	0.298	0.298	0.547	0.609	0.615	0.441
	KISGP [ZHQ*21] (2022)	0.285	0.300	0.301	0.593	0.650	0.653	0.464
	Our method	**0.335**	**0.360**	**0.362**	**0.601**	**0.662**	**0.728**	**0.508**
w/o GC	Co-Occurrence [ZYTC18]	0.141	0.202	0.258	0.351	0.556	0.706	0.369
	KERN [CYCL19]	0.208	0.247	0.276	0.483	0.648	0.772	0.439
	SGPN [WDNT20]	0.282	0.326	0.353	0.545	0.701	0.824	0.505
	Schemata [SBT21]	0.288	0.335	0.363	0.496	0.671	0.802	0.493
	SGF [WWT*21]	0.290	0.332	0.357	0.557	0.728	0.834	0.516
	EdgeGCN [ZYSC21]	0.296	0.338	0.359	0.569	0.779	0.859	0.533
	KISGP [ZHQ*21]	0.298	0.343	0.370	0.622	0.784	0.883	0.550
	Our method	**0.342**	**0.377**	**0.398**	**0.634**	**0.793**	**0.890**	**0.572**

Table 2: *Comparison of mean recall on the two tasks of the 3DSSG dataset.*

Method	SGCLS			PREDCLS			
	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	Mean
Co-Occurrence [ZYTC18]	0.088	0.127	0.129	0.338	0.474	0.479	0.273
KERN [CYCL19]	0.095	0.115	0.119	0.188	0.256	0.265	0.173
SGPN [WDNT20]	0.195	0.226	0.231	0.321	0.384	0.389	0.291
Schemata [SBT21]	0.238	0.270	0.272	0.352	0.426	0.433	0.332
SGF [WWT*21]	0.242	0.281	0.282	0.453	0.531	0.532	0.387
EdgeGCN [ZYSC21]	0.245	0.291	0.292	0.543	0.621	0.622	0.436
KISGP [ZHQ*21]	0.244	0.286	0.288	0.566	0.635	0.638	0.443
Our method	**0.254**	**0.297**	**0.298**	**0.577**	**0.640**	**0.643**	**0.452**

among these methods. On this basis, KERN network [CYCL19] and Schemata [SBT21] improve the accuracy of 3D scene graph generation by 0.067 and 0.113 on average, respectively. These results indicate the capability of prior knowledge. SGF [WWT*21] and EdgeGCN [ZYSC21] have improved the GCN in the model at varying degrees on the basis of SGPN [WDNT20], and the results have been improved to a certain extent (0.428 and 0.441), indicating that the task of SGG is sensitive to the features of propagation in GCN. KISGP [ZHQ*21] achieves better results through pre-trained class label embedding. Our method significantly improves the baselines in two standard tasks. In particular, the accuracy can achieve 0.508 which exceeds KISGP by 0.044. The results demonstrate the robust representation capability of the ternary learning strategy. We also present the Recall on the two tasks without constraint in Table 1. Still, our method achieves the best results on these metrics.

Owing to the scarcity of the tail class annotation in 3DSSG, previous studies usually achieve poor performance on less frequent classes. Hence, the Mean Recall (mR) is also utilized as an evaluation metric [CYCL19, TZW*19].

As shown in Table 2, these are fairer measures for an unbiased SGG [TNH*20]. Co-Occurrence [ZYTC18] reflects the importance weight of each class, which can achieve a better effect on the Mean Recall, for which our method shows a large absolute gain. The mean of the mR over all two evaluation metrics is 0.452 for our method. This value outperforms the two recently effective methods EdgeGCN [ZYSC21] (0.07 for SGCLS, 0.247 for PREDCLS) and KISGP [ZHQ*21] (0.103 for SGCLS, 0.07 for PREDCLS) on average. All results show that our method is especially capable of dealing with the class imbalance problem.

4.4. Qualitative results

Figure 5 shows three challenging scenes. Our model solves the incorrect identification of tail classes and incorrect classification

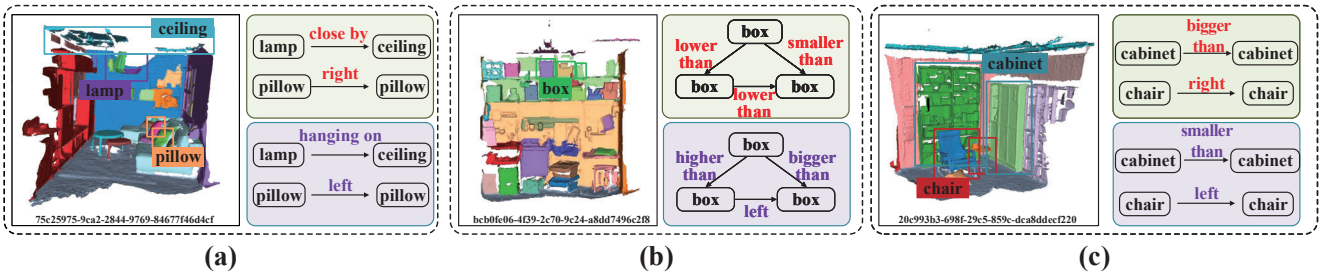

Figure 5: *Qualitative examples of the improvement in 3D scene graph generation. On the right side of each scene, the result of the KISGP [ZHQ*21] is at the top, and our result is at the bottom. The purple predicates are those correctly classified relationships (in ground truth), and the red predicates are those incorrectly classified relationships. For better viewing, we only show failure cases. The scene ID of the test set is below the picture.*

of predicates between the same objects by introducing semantic knowledge. For example, considering the 3D scene on the Figure 5(a), we succeeded in correcting the bias in the class label embeddings based on the semantic clues (from **close by** to **hanging on**, **lamp-hanging on-ceiling** has a high co-occurrence probability in the train set, we use entity level SC to propagate this information). Considering the Figure 5(b), the visual appearance of the box is deceptive, which easily makes the model randomly "guess" a result in the label embedding space (**lower than** or **higher than**, **smaller than** or **bigger than**). With our novel VLT method, we correctly identify the comparative predicates between two boxes. The prediction is precise because our model propagates the 3D visual appearance of the important neighbors. Another interesting example is the Figure 5(c). Our model can correct not only comparative predicates, but also proximity predicates (**right** and **left**).

4.5. Ablation study

In our ternary learning framework, two modules are proposed, an SC-GCN and VLT. In this section, we investigate the contributions of the proposed SC-GCN and VLT to the performance of 3D scene graph generation. We show the ablation performance on the 3DSSG dataset in Table 3.

Effectiveness of semantic clues in SC-GCN: We conduct experiments by removing two types of semantic clues from our SC-GCN, leading to a network with only GCN layers. When only the 3D visual features are enabled, the model works the same as the GCN-based method and also has a similar performance as KISGP [ZHQ*21]. The isolated semantic SC (EXP 2 and EXP 3) boosts the performance slightly with the additional information from train set. PREDCLS and SGCLS are improved significantly when the entity level SC and path level SC both work (EXP 4). The learned semantic features help GCN fully understand the entity dependencies in the dataset and have a strong, positive effect both on SGCLS and PREDCLS.

Message passing iterations L in SC-GCN: The role of message passing parameter L is also evaluated in Table 3 (see EXP 6-8). According to our experiments, the model, by learning feature representations with only one GCN layer for message passing ($L = 1$), achieves 0.489 on average, which is lower than the model using 2 GCN layers ($L = 2$). This results indicate that increasing the layer number is beneficial to propagate the context information of adjacent nodes. Interestingly, when using more GCN layers ($L = 3$), we do not observe any benefit (0.488 on average). The reason is that the redundant information in semantic knowledge embeddings causes the output features of nodes to be over-smoothed, and the representation vectors of nodes tend to be consistent, which makes it impossible to distinguish nodes of different classes. Therefore, this suggests that $L = 2$ is the optimal choice.

The impact of VLT: In this paper, we design one insightful module to build intrinsic interactions across 3D visual and language modalities, i.e. visual-language transformer module. To be specific, to exploit the effectiveness of the VLT module, Experiment 5 in Table 3 directly takes the output results of SC-GCN for prediction. The results show that our VLT module could effectively improve the performance, e.g., 0.02 for SGCLS and 0.055 for PREDCLS on average.

4.6. Further analysis

Effects of different class label embeddings: To verify the effects of different class label embeddings, we utilize Bert [DCLT18], Glove [PSM14] and KISGP [ZHQ*21] to respectively extract class label embeddings as the inputs of our proposed framework. Compared with the obvious performance gain in Table 3 (EXP 9-11), it can be found that adopting different label embeddings from different extraction methods has little effect on the final classification performance. That is, as representations of different modalities, label embeddings are important inputs for our proposed framework, but the choice of specific extraction methods, e.g., Bert or Glove is not the key factor in our framework.

Why does our global scope of semantic relationships help more on recall? To intuitively illustrate the performance gains brought by the ternary learning framework, we compare the performance of each class of KISGP [ZHQ*21] and our proposed method as shown in Figure 6. In general, Recall@5 of the tail predicates (e.g., **lying on**, and **cover**) has been improved by different programs at the cost of a massive decrease in the results of the head predicates (e.g., **supported by**, **left**, and **right**). This shows that our

Table 3: *Ablation studies on the proposed semantic clue aware graph convolution network and visual-language transformer module.*

Exp	Module	SGCLS			PREDCLS			
		R@20	R@50	R@100	R@20	R@50	R@100	Mean
1	Our full method (SC-GCN+VLT)	**0.335**	**0.360**	**0.362**	**0.601**	**0.662**	**0.728**	**0.508**
2	w/o entity SC	0.331	0.353	0.353	0.595	0.655	0.665	0.492
3	w/o path SC	0.290	0.320	0.333	0.598	0.640	0.651	0.472
4	w/o entity & path SC	0.284	0.310	0.315	0.594	0.630	0.631	0.461
5	w/o VLT	0.305	0.338	0.355	0.561	0.631	0.634	0.471
6	SC-GCN (L=1)	0.326	0.359	0.361	0.583	0.632	0.673	0.489
7	SC-GCN (L=2)	**0.335**	**0.360**	**0.362**	**0.601**	**0.662**	**0.728**	**0.508**
8	SC-GCN (L=3)	0.324	0.354	0.357	0.591	0.648	0.659	0.488
9	label embeddings by Bert	**0.335**	**0.360**	**0.362**	**0.601**	**0.662**	**0.728**	**0.508**
10	label embeddings by Glove	0.324	0.354	0.357	0.591	0.648	0.718	0.499
11	label embeddings by KISGP	0.334	0.364	0.368	0.593	0.651	0.722	0.505

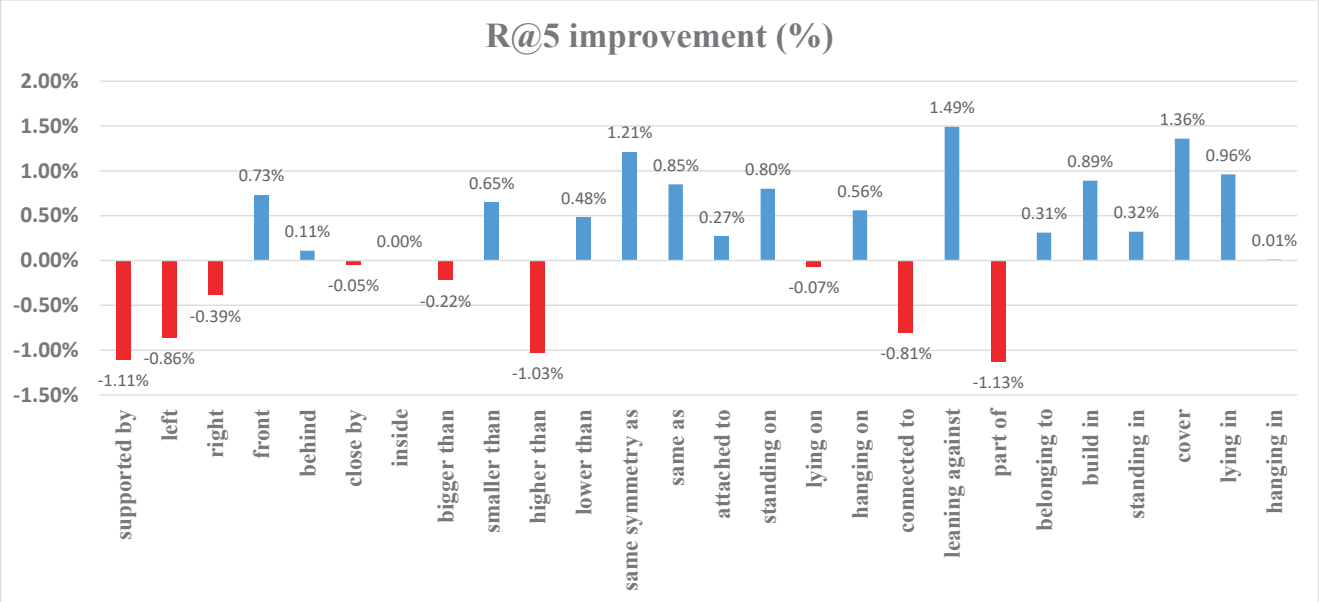

Figure 6: *Improvement in PREDCLS of our full model for R@5 in comparison with KISGP [ZHQ*21] under a graph constraint.*

semantic knowledge still has the ability to alleviate the long-tailed effect. Specifically, our method achieves a 0.65% improvement on **smaller than** and 0.22% decrease on **bigger than**, which indicates that our model solves the problem of comparative predicates recognition error between the same objects after combining 3D visual structure features. Relying on explicit labels and well-defined semantic clues, rather than embedded in latent space, our model has broader explanatory and predictive potential. This means that our results are beyond the simple reflection of the statistical bias of a semantic space which achieved a more generalized performance.

How semantic clues affect feature distribution? We record initialized features of the unclassified scene graph $\mathcal{G}_s$ (Figure 7(a)), and the refined graph $\mathcal{G}_s'$ (Figure 7(b)) from the SC-GCN module, and their corresponding labels on the 3DSSG dataset. Then we take the average for the features according to the labels and use the t-SNE [VdMH08] method to visualize them as shown in Figure 7. Note that if features of some classes are closed to each other, the edges between those close classes are more likely to be activated.

From the enlarged regions in Figure 7 we can find that by introducing our semantic clues on the cluttered initialized features, the entity features which share high co-occurrence probabilities are likely to be closed to each other, such as **hanging on** and **lamp**, **ceiling** and **shower**. From these results we can find that our semantic knowledge can be well incorporated in the 3D scene graph generation process to guide the feature refinement, therefore producing better recall results.

More qualitative results: Additional qualitative results for 3D scene graph generation are shown in Figure 8. The 3D scene graphs are generated in the PREDCLS task. For the Figure 8(a), we cor-

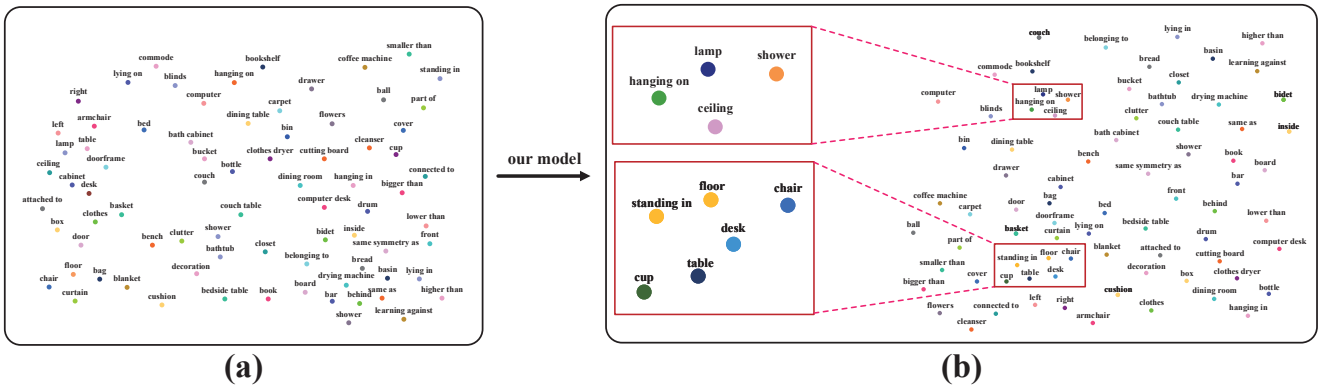

Figure 7: *Visualization of initialized and refined entity features by t-SNE [VdMH08].*

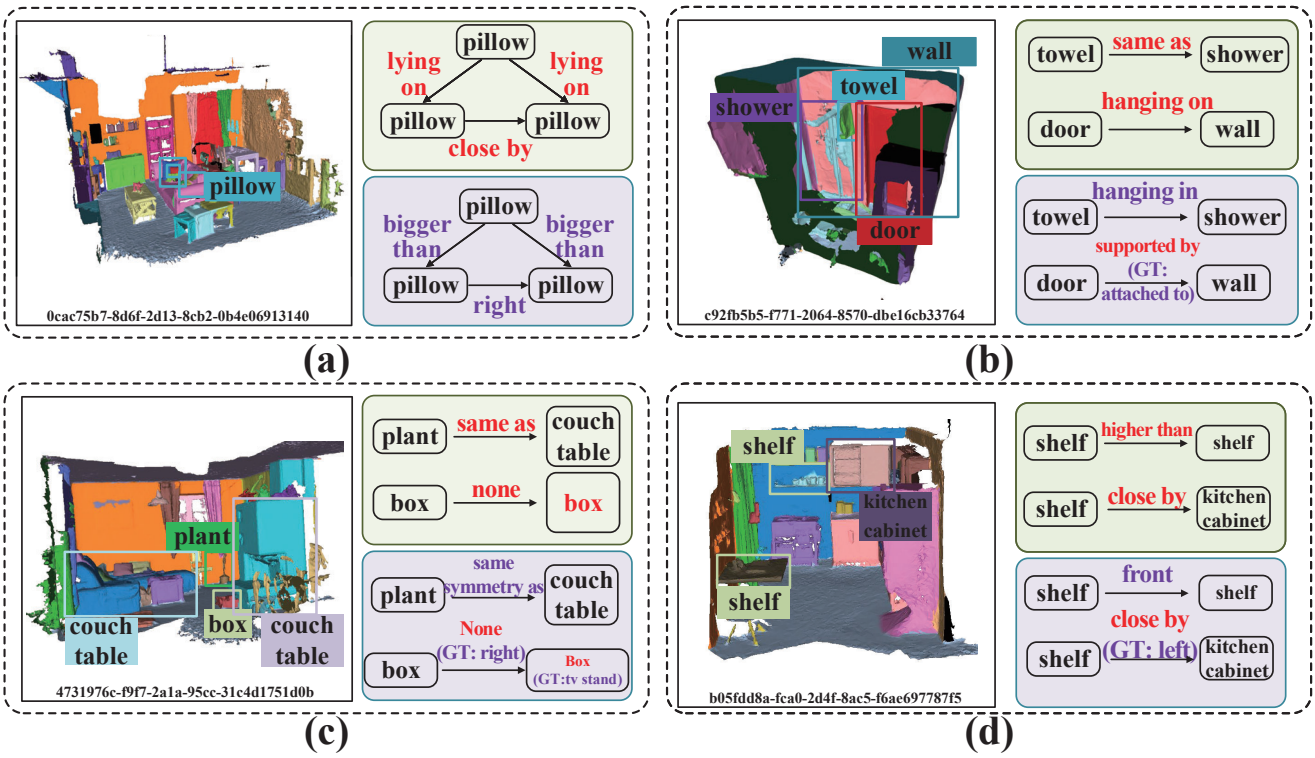

Figure 8: *Additional qualitative examples of the improvement in 3D scene graph generation.*

rectly identify the comparative predicates between three pillows. Another interesting phenomenon is for the Figure 8(c), we classified a predicate **same symmetry as** that was difficult to identify as a result of our introduction of visual features, and SOTA [ZHQ*21] recognizes it as **same as**.

Failure Cases: we analyze the qualitative results and summarize the following most common failure cases to clarify the limitation of the model: 1. The 3D object is incorrectly detected. 2. The predicates have similar meanings. 3. The model predicts the wrong tail predicate instead of the correct head predicate. The failure cases are shown in Figure 8. For the Figure 8(c), neither SOTA [ZHQ*21] nor our model detected **tv stand**. We conjecture that Failure 1 can be improved by a better object detector or a more robust feature extractor such as DGCNN [WSL*19]. Failure 2 caused by the human-labeled annotations (For the Figure 8(d), both **left** and **close by** can correctly describe the relationship between **shelf** and **kitchen cabinet**). Failure 3 (For the Figure 8(b), **attached to** is falsely detected

as **supported by**) is caused by the imbalanced predicate distribution both in the 3D scene graph dataset and in the real world.

5. Conclusions

In this article, we introduce a novel ternary learning framework to build the global scope of semantic context as well as interactions between 3D visual modality and language modality. Specifically, for building the global scope of semantic context, we distill two types of semantic clues from the training set and design a novel SC-GCN model to obtain more powerful semantic representation. For constructing the interactions between the 3D visual and language modalities, we propose a visual-language transformer module to embed the class label embeddings into the 3D visual structure learning. According to our experiments, with the aid of SC-GCN and VLT, our full method can better understand the given 3D scene and produce more precise 3D scene graph results. Our results achieve new state-of-the-arts on the 3DSSG dataset.

However, due to the imbalance of annotations, the ability of our model to predict infrequent predicates is still limited. In future work, we consider using a more robust point cloud feature extractor and the global features of all points, rather than inputting the point cloud of each object in isolation. This idea can be extended to end-to-end 3D scene graph generation without inputting segmentation results. Moreover, since this paper has explored the use of structured semantic knowledge of dataset to improve 3D scene graph generation, our work also opens a promising direction for few-shot learning or zero-shot learning in 3D SGG.

Acknowledgment

This work was supported in part by Natural Science Foundation Project of CQ (No. CSTC2021JCYJ-MAXMX0062), National Natural Science Foundation of China (No. 62002121 and 62072183), the National Key Research and Development Program of China (No. 2021ZD0111000), Shanghai Science and Technology Commission (No. 21511100700), the Research Project of Shanghai Science and Technology Commission (No. 20DZ2260300), the Open Project Program of the State Key Lab of CAD&CG (No. A2203), Zhejiang University.

References

[AHG*19] Armeni I., He Z.-Y., Gwak J., Zamir A. R., Fischer M., Malik J., Savarese S.: 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 5664–5673. 2

[AJK*22] Agia C., Jatavallabhula K. M., Khodeir M., Miksik O., Vineet V., Mukadam M., Paull L., Shkurti F.: Taskography: Evaluating robot task planning over large 3d scene graphs. In *Conference on Robot Learning* (2022), PMLR, pp. 46–58. 2

[BYCCT17] Ben-Younes H., Cadene R., Cord M., Thome N.: Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2612–2620. 1

[CMS*20] Carion N., Massa F., Synnaeve G., Usunier N., Kirillov A., Zagoruyko S.: End-to-end object detection with transformers. In *European conference on computer vision* (2020), Springer, pp. 213–229. 3

[CVMG*14] Cho K., Van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014). 5

[CYCL19] Chen T., Yu W., Chen R., Lin L.: Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 6163–6171. 3, 4, 5, 7

[CZG*22] Chandran P., Zoss G., Gross M., Gotardo P., Bradley D.: Shape transformers: Topology-independent 3d shape models using transformers. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 195–207. 3

[DCLT18] Devlin J., Chang M.-W., Lee K., Toutanova K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018). 3, 6, 8

[DMNT21] Dhamo H., Manhardt F., Navab N., Tombari F.: Graph-to-3d: End-to-end generation and manipulation of 3d scenes using scene graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 16352–16361. 2

[HBX*20] Herzig R., Bar A., Xu H., Chechik G., Darrell T., Globerson A.: Learning canonical representations for scene graph to image generation. In *European Conference on Computer Vision* (2020), Springer, pp. 210–227. 1

[HWQ*19] Hou J., Wu X., Qi Y., Zhao W., Luo J., Jia Y.: Relational reasoning using prior knowledge for visual captioning. *arXiv preprint arXiv:1906.01290* (2019). 3

[JGFF18] Johnson J., Gupta A., Fei-Fei L.: Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 1219–1228. 1

[KB14] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 6

[KMMI*21] Kurenkov A., Martín-Martín R., Ichnowski J., Goldberg K., Savarese S.: Semantic and geometric modeling with neural message passing in 3d scene graphs for hierarchical mechanical search. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021), IEEE, pp. 11227–11233. 2

[KW17] Kipf T. N., Welling M.: Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)* (2017). 1

[LGG*17] Lin T.-Y., Goyal P., Girshick R., He K., Dollár P.: Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2980–2988. 6

[LOZ*17] Li Y., Ouyang W., Zhou B., Wang K., Wang X.: Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 1261–1270. 1

[LS04] Liu H., Singh P.: Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal 22*, 4 (2004), 211–226. 3

[LYY*19] Li L. H., Yatskar M., Yin D., Hsieh C.-J., Chang K.-W.: Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557* (2019). 3

[MAA*19] Mittal G., Agrawal S., Agarwal A., Mehta S., Marwah T.: Interactive image generation using scene graphs. *arXiv preprint arXiv:1905.03743* (2019). 1

[MHN*13] Maas A. L., Hannun A. Y., Ng A. Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (2013), vol. 30, Citeseer, p. 3. 5

[Mil95] Miller G. A.: Wordnet: a lexical database for english. *Communications of the ACM 38*, 11 (1995), 39–41. 3

[MKK21] Mogadala A., Kalimuthu M., Klakow D.: Trends in integration of vision and language research: A survey of tasks, datasets, and methods. *Journal of Artificial Intelligence Research 71* (2021), 1183–1317. 3

[PGC*17] PASZKE A., GROSS S., CHINTALA S., CHANAN G., YANG E., DEVITO Z., LIN Z., DESMAISON A., ANTIGA L., LERER A.: Automatic differentiation in pytorch. 6

[PSM14] PENNINGTON J., SOCHER R., MANNING C. D.: Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543. 6, 8

[QSMG17] 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* (2017), pp. 652–660. 2

[RKH*21] 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 *International Conference on Machine Learning* (2021), PMLR, pp. 8748–8763. 3

[RPG*21] RAMESH A., PAVLOV M., GOH G., GRAY S., VOSS C., RADFORD A., CHEN M., SUTSKEVER I.: Zero-shot text-to-image generation. In *International Conference on Machine Learning* (2021), PMLR, pp. 8821–8831. 3

[RPH*21] RAVICHANDRAN Z., PENG L., HUGHES N., GRIFFITH J. D., CARLONE L.: Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks. *arXiv preprint arXiv:2108.01176* (2021). 2

[RPS21] REICH D., PUTZE F., SCHULTZ T.: Adventurer's treasure hunt: A transparent system for visually grounded compositional visual question answering based on scene graphs. *arXiv preprint arXiv:2106.14476* (2021). 1

[SBT21] SHARIFZADEH S., BAHARLOU S. M., TRESP V.: Classification by attention: Scene graph classification with prior knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 5025–5033. 3, 7

[TNH*20] 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* (2020), pp. 3716–3725. 7

[TSNI20] 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)* (2020), IEEE, pp. 249–255. 2

[TZW*19] 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* (2019), pp. 6619–6628. 1, 7

[UBH*22] UPPAL S., BHAGAT S., HAZARIKA D., MAJUMDER N., PORIA S., ZIMMERMANN R., ZADEH A.: Multimodal research in vision and language: A review of current and emerging trends. *Information Fusion 77* (2022), 149–171. 3

[VdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, 11 (2008). 9, 10

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L., POLOSUKHIN I.: Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017). 6

[WAN*19] WALD J., AVETISYAN A., NAVAB N., TOMBARI F., NIESSNER M.: Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 7658–7667. 1, 6

[WDNT20] 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* (2020), pp. 3961–3970. 1, 2, 6, 7

[WSL*19] WANG Y., SUN Y., LIU Z., SARMA S. E., BRONSTEIN M. M., SOLOMON J. M.: Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog) 38*, 5 (2019), 1–12. 10

[WWT*21] 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* (2021), pp. 7515–7525. 1, 2, 7

[XZCFF17] 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* (2017), pp. 5410–5419. 4, 6

[XZH*18] XU T., ZHANG P., HUANG Q., ZHANG H., GAN Z., HUANG X., HE X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 1316–1324. 1

[YPLM18] YAO T., PAN Y., LI Y., MEI T.: Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 684–699. 1

[YTZC19] YANG X., TANG K., ZHANG H., CAI J.: Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 10685–10694. 1

[ZCX19] ZHANG C., CHAO W. L., XUAN D.: An empirical study on leveraging scene graphs for visual question answering. In *British Machine Vision Conference (BMVC)* (2019). 1

[ZHQ*21] ZHANG S., HAO A., QIN H., ET AL.: Knowledge-inspired 3d scene graph prediction in point cloud. *Advances in Neural Information Processing Systems 34* (2021). 1, 3, 6, 7, 8, 9, 10

[ZKC20] ZAREIAN A., KARAMAN S., CHANG S.-F.: Bridging knowledge graphs to generate scene graphs. In *European Conference on Computer Vision* (2020), Springer, pp. 606–623. 3, 5

[ZLZ*21] ZHENG S., LU J., ZHAO H., ZHU X., LUO Z., WANG Y., FU Y., FENG J., XIANG T., TORR P. H., ET AL.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 6881–6890. 3

[ZSL*20] ZHU X., SU W., LU L., LI B., WANG X., DAI J.: Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* (2020). 3

[ZSY*21] ZHONG Y., SHI J., YANG J., XU C., LI Y.: Learning to generate scene graph from natural language supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 1823–1834. 3

[ZTBZ21] ZHU Y., TREMBLAY J., BIRCHFIELD S., ZHU Y.: Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021), IEEE, pp. 6541–6548. 2

[ZWC*20] ZHONG Y., WANG L., CHEN J., YU D., LI Y.: Comprehensive image captioning via scene graph decomposition. In *European Conference on Computer Vision* (2020), pp. 211–229. 1

[ZWYC20] ZAREIAN A., WANG Z., YOU H., CHANG S.-F.: Learning visual commonsense for robust scene graph generation. In *European Conference on Computer Vision* (2020), Springer, pp. 642–657. 3

[ZYSC21] 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* (2021), pp. 9705–9715. 1, 2, 7

[ZYTC18] 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* (2018), pp. 5831–5840. 6, 7