HL-Net: Heterophily Learning Network for Scene Graph Generation

Xin Lin[1]* Changxing Ding[1,2]† Yibing Zhan[3] Zijian Li[1] Dacheng Tao[3,4]
[1] South China University of Technology [2] Pazhou Lab, Guangzhou [3] JD Explore Academy
[4] The University of Sydney

eelinxin@mail.scut.edu.cn, chxding@scut.edu.cn, eezijianli@mail.scut.edu.cn,
zhanyibing@jd.com, dacheng.tao@gmail.com

Abstract

Scene graph generation (SGG) aims to detect objects and predict their pairwise relationships within an image. Current SGG methods typically utilize graph neural networks (GNNs) to acquire context information between objects/relationships. Despite their effectiveness, however, current SGG methods only assume scene graph homophily while ignoring heterophily. Accordingly, in this paper, we propose a novel Heterophily Learning Network (HL-Net) to comprehensively explore the homophily and heterophily between objects/relationships in scene graphs. More specifically, HL-Net comprises the following 1) an adaptive reweighting transformer module, which adaptively integrates the information from different layers to exploit both the heterophily and homophily in objects; 2) a relationship feature propagation module that efficiently explores the connections between relationships by considering heterophily in order to refine the relationship representation; 3) a heterophily-aware message-passing scheme to further distinguish the heterophily and homophily between objects/relationships, thereby facilitating improved message passing in graphs. We conducted extensive experiments on two public datasets: Visual Genome (VG) and Open Images (OI). The experimental results demonstrate the superiority of our proposed HL-Net over existing state-of-the-art approaches. In more detail, HL-Net outperforms the second-best competitors by 2.1% on the VG dataset for scene graph classification and 1.2% on the IO dataset for the final score. Code is available at https://github.com/siml3/HL-Net.

1. Introduction

Scene graph generation (SGG) has recently attracted increasing attention from the research community. As illustrated in Figure 1, a visual scene could be depicted in the form of a graph structure, where objects and their pairwise relationships are represented by the nodes and edges, respectively. A triplet constructed by two objects and their corresponding relationship then takes the form *Subject-Predicate-Object*. Intuitively, the key to SGG methods is to model and explore the connections between objects, as well as those between relationships. Due to their remarkable ability to model the connections between graph components, graph neural networks (GNNs) have been widely adopted in SGG tasks [2, 12, 14, 30, 31].

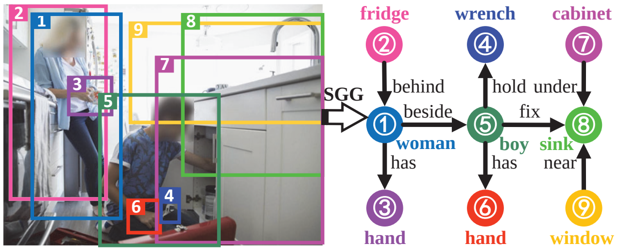


Figure 1. An image and its ground-truth scene graph. Objects and their pairwise relationships are represented as nodes and edges, respectively. Best viewed in color.

Despite their effectiveness, existing GNN-based SGG methods only assume homophily [17] between objects/relationships; in other words, these methods calculate the correlations between objects/relationships by implicitly treating all objects/relationships as belonging to the same categories. However, as Figure 1 demonstrates, scene graphs fall naturally into the category of heterophilic graphs. We, therefore, argue that heterophily, *i.e.*, the interaction between objects/relationships from different categories, should be modeled directly. In this paper, we focus on the heterophily in class labels, following the definition provided in [29, 41].

Exploring heterophily in the SGG is non-trivial. There are at least two problems that must be considered. First, heterophily in both objects and relationships should be taken into account; however, no prior SGG works have explicitly considered heterophily, and no heterophilic GNNs have

*Work done during first author's internship at JD Explore Academy
†Corresponding author.

exploited heterophily in visual relationships. Second, two objects/relationships characterized by significant occlusion usually have similar visual appearances, despite being from different classes, which increases the difficulty of distinguishing heterophily from homophily. In the light of the above scene graph analysis, in this paper, we propose a Heterophily Learning Network (HL-Net) for SGG to comprehensively and efficiently explore the heterophily in objects/relationships. To the best of our knowledge, HL-Net is the first work to consider heterophily for the SGG. The main contributions of HL-Net are summarized below.

We first propose an **Adaptive Reweighting Transformer** (ART), which refines object representation with heterophily considered. In more detail, we arrange the pre-layer normalization [28], residual connection, and feedforward network to deepen the layers and enhance the object feature with contextual information. Furthermore, the refined object representations of different ART layers are aggregated with learnable weights. These weights depend on the contributions of different ART layers and can be both positive and negative. This aggregation procedure is similar to general polynomial graph filtering [20], which is naturally able to deal with both the high-frequency context (*i.e.*, heterophily) and low-frequency context (*i.e.*, homophily) between objects [3].

We then develop a **Relationship Feature Propagation** (RFP) module that explores the connections between heterophilic relationships. Two challenges emerge in the design of RFP: the effectiveness of feature propagation and heterophily modeling. To reduce computational complexity, we only require each relationship to contact neighboring relationships that share the subject or object. Moreover, the contextual coefficients obtained from ART are adopted to represent the correlations between relationships. To propagate heterophilic features between relationships, we extend the PageRank-based GNN [8] to a high-pass graph filter. This approach enables the RFP module to learn the relationship correlation of disparate classes by passing relevant high-frequency graph signals (*i.e.*, heterophily).

Finally, we devise a **Heterophily-aware Message Passing** (HMP) scheme to identify the heterophily and homophily between objects or relationships in complicated visual scenes (*e.g.*, overlapping objects that belong to different classes). More specifically, HMP utilizes the spatial and visual information of object/relationships to produce signed messages, which can subsequently be applied to adjust the contextual coefficients and guide the learning processes in both ART and RFP.

We conduct extensive experiments on two public datasets: Visual Genome (VG) [9] and Open Images (OI) [10]. Experimental results demonstrate that the proposed HL-Net consistently achieves top-level performance. Ablation studies further verify both the necessity and effectiveness of considering heterophily for SGG.

2. Related Works

Scene Graph Generation. Early SGG works [33–36, 42] tended to detect each object/relationship independently, ignoring the intrinsic connections between objects/relationships. Recent SGG methods [2, 4, 12, 14, 23, 30–32] typically explore visual-contextual information between objects. These methods can be roughly divided into two categories: Recurrent Neural Network (RNN)-based methods and Graph Neural Network (GNN)-based methods. The first category utilizes RNN to encode contextual information. For example, Zeller *et al*. [32] and Tang *et al*. [23] employed a bidirectional long short-term memory (LSTM) module and a tree structure-based LSTM module to refine object representation using context information, respectively. However, RNN-based SGG approaches may not adequately depict connections between distant objects. The second category of methods utilizes GNN to propagate contextual information. For example, Yang *et al*. [31] proposed an attentional graph convolutional network to refine object and relationship representations, while Lin *et al*. [14] proposed a direction-aware message-passing module that encodes the edge direction information. However, recent studies [29, 41] have proven that most existing GNN-based methods struggle to describe connections under heterophily. Unfortunately, scene graphs are naturally heterophilic. To address these problems, we propose HL-Net in an attempt to capture the heterophilic property of scene graphs. To the best of our knowledge, HL-Net is the first work to explicitly consider heterophily in the SGG.

GNNs and Heterophily. Recent works [15, 17, 18, 41] have shown that the use of certain GNNs (*e.g.*, Graph Convolutional Network [7] and Graph Attention Network [25]) can lead to significant performance loss in heterophilous settings. A number of works have attempted to address this issue. For example, Zhu *et al*. [41] proposed a set of designs including embedding separation, higher-order neighborhoods aggregation, and intermediate representations that enable GNN to perform well under heterophilic settings. Zhou *et al*. [39] introduced a new belief propagation-based GNN model. Chien *et al*. [3] devised a generalized PageRank-based GNN architecture that adaptively learns the propagation weights to determine the polynomial graph filter for heterophilic graph. Yan et al [29] proposed a model that allows negative interactions between nodes in order to capture heterophily. However, the above approaches focus primarily on the task of natural language processing, (*e.g.*, node classification for citation graphs). Therefore, applying the above-mentioned heterophilic GNNs directly to the SGG may not adequately solve the heterophily problem for visual content.

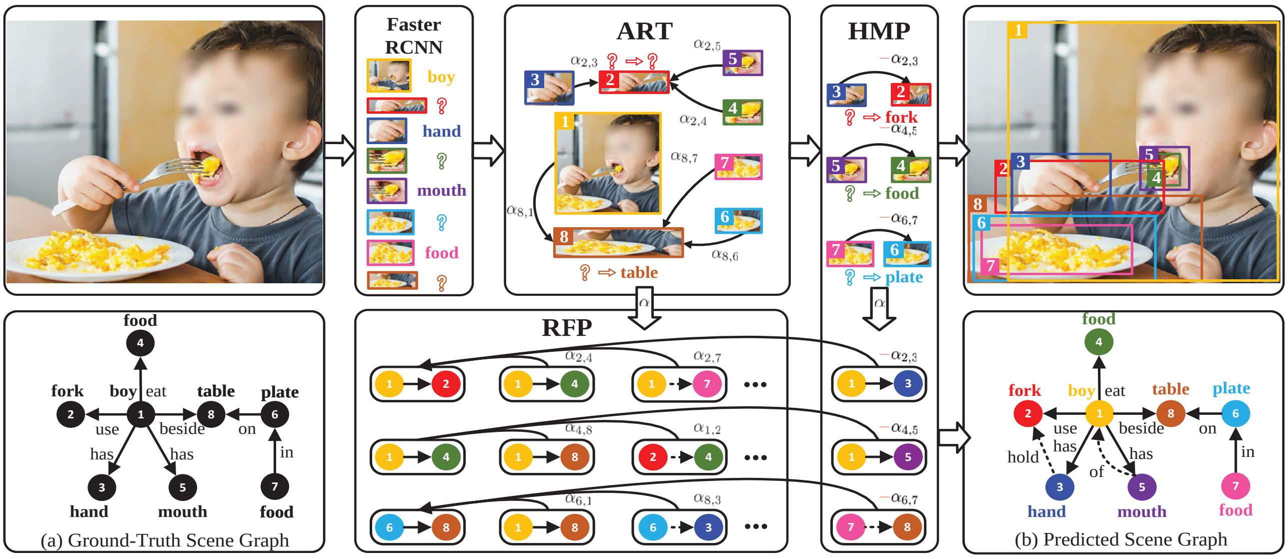


Figure 2. The framework of HL-Net. HL-Net obtains object proposals through Faster R-CNN [19]. It then improves the performance of SGG through the application of two novel modules: (1) an ART module that enables message-passing between objects with heterophily considered; (2) an RFP module that explores connections between heterophilic relationships. Moreover, HL-Net includes an HMP scheme that identifies the heterophily and homophily between objects and those between relationships under complicated visual scenes.

3. Heterophily Learning Network

This section presents the framework of our Heterophily Learning Network (HL-Net). As Figure 2 illustrates, HL-Net comprises an adaptive reweighting transformer (ART) module and a relationship feature propagation (RFP) module. The ART module strengthens the network's object classification ability by means of heterophily-aware message passing between object representations. The RFP module promotes its predicate classification performance by exploring connections between heterophilic relationships. We further propose a heterophily-aware message passing (HMP) scheme that identifies the heterophily and homophily between objects, along with those between relationships, and enhance the power of both ART and RFP. In the below, we will describe these three components sequentially.

3.1. Preliminary

Notations. We first introduce the notations used in this section. We adopt exactly the same approach used in [32] to obtain the representation $\boldsymbol{x}_i$ for the i-th object/node. More specifically, $\boldsymbol{x}_i$ is transformed using linear projection from the concatenation of the visual appearance feature, object classification probabilities, and the spatial feature. We extract appearance feature form the union box of the two nodes i and j, denoted as $\boldsymbol{x}_{ij}$; similarly, the appearance feature for the union box of three nodes i, j, and k is denoted as $\boldsymbol{x}_{ijk}$. The relationship feature between the i-th and j-th nodes is represented as $\boldsymbol{r}_{ij}$, where the i-th node is the *subject* and the j-th node is the *object*. $\boldsymbol{B}_{ij}$ is used to represent the relative spatial feature between the i-th and j-th nodes. It is obtained by applying two convolutional layers and two FC layers to binary maps of size $14 \times 14 \times 2$, with each channel representing the area of one node. Similarly, $\boldsymbol{B}_{ij,k}$ denotes the relative spatial feature between the union box of nodes i, j and the bounding box for the node k, and $\boldsymbol{B}_{i,jk}$ represent the relative spatial feature between the bounding box of node i and the union box of nodes j, k. $\odot$ represents the Hadmard product. $\mathcal{N}_i$ denotes the set of neighboring objects for the i-th node. $\mathcal{N}_{\boldsymbol{r}_{ij}}$ indicates the set of neighboring relationships of $\boldsymbol{r}_{ij}$. Finally, $\boldsymbol{W}$ stands for a linear transformation matrix and $\boldsymbol{w}$ means a linear projection vector.

Homophily and Heterophily. Given a set of node classes, *homophily* describes the tendency of a node to have the same class as its neighbors, while *heterophily* describes the tendency of a node to have different classes as its neighbors. In more detail, [3, 18, 29] proposed a metric for measuring the level of homophily of nodes in a graph: $h = \frac{1}{\|\mathcal{V}\|} \sum_{i \in \mathcal{V}} \frac{\|\mathcal{N}_i^s\|}{\|\mathcal{N}_i\|}$, here $\mathcal{V}$ represents a node set, $\mathcal{N}_i^s$ denotes the set of neighboring nodes with the same label as the i-th node, and $\|\cdot\|$ is the cardinality operator. Accordingly, $h \rightarrow 1$ corresponds to strong homophily, while $h \rightarrow 0$ indicates strong heterophily. This definition could be extended to describe the homophily and heterophily of edges.

3.2. Adaptive Reweighting Transformer

Graph attention network has been widely adopted in existing SGG methods [2, 12, 14, 31]. However, recent works [3, 29, 41] have shown that graph attention network implicitly assumes homophily between nodes; therefore and accordingly ignores the property of heterophily in scene graph. To address this problem, we propose the ART module, which includes two components: namely, the **Pre-LN Transformer** and **Adaptive Graph Filter**.

Pre-LN Transformer: We adopt the same approach as in [14] to obtain the contextual coefficient c_{ij} between two nodes i and j as follows:

$$c_{ij} = \boldsymbol{w}_c^T(\boldsymbol{W}_{c1}\boldsymbol{x}_i \odot \boldsymbol{W}_{c2}\boldsymbol{x}_j \odot (\boldsymbol{x}_{ij} + \boldsymbol{B}_{ij})). \tag{1}$$

Inspired by [28], we employ pre-layer normalization (Pre-LN) to stabilize the model training. The neighboring messages for the i-th node can be aggregated as follows:

$$\mathcal{F}(\mathcal{N}_i) = \sum\nolimits_{j\in\mathcal{N}_i} \alpha_{ij}\sigma(\boldsymbol{W}_{\mathcal{F}}\mathrm{LN}(\boldsymbol{x}_j)), \tag{2}$$

where σ denotes the ReLU activation function, while α_{ij} is a contextual coefficient, which is obtained by normalizing c_{ij} with softmax. Furthermore, we adopt layer normalization [1], FFN layer [24], and residual connection sequentially to refine the node representations. Consequently, the output of the u-th layer for the i-th node can be denoted as follows:

$$\boldsymbol{x}_i^{u+1} = z_i^u + \mathrm{FFN}(\mathrm{LN}(\overbrace{\boldsymbol{x}_i^u + \mathcal{F}^u(\mathcal{N}_i)}^{z_i^u})). \tag{3}$$

Adaptive Graph Filter: As proven in [40], existing GNN approaches [6, 8, 26] typically focus on emphasizing homophily by aggregating the outputs of different GNN layers with non-negative weights. This aggregation step can be understood as a low-pass graph filter that emphasizes the low-frequency part of the graph signal (*i.e.*, homophily). However, this filter suppresses high-frequency components (*i.e.*, heterophily) in the graph signal. In comparison, if the outputs of GNN layers can be aggregated with negative weights, a polynomial graph filter [20] for heterophilic graphs can be obtained [3].

Motivated by the above analysis, ART calculates the final node representation as follows:

$$\hat{\boldsymbol{x}}_i = \mathrm{LN}(\sum\nolimits_{u=1}^{U} \gamma_u \boldsymbol{x}_i^u), \tag{4}$$

Here, U denotes the number of GNN layers while γ_u represents the weight of the u-th GNN layer. Note that γ_u can be a negative number and is optimized simultaneously with the whole HL-Net in an end-to-end manner. To properly capture the heterophilic property of the scene graph, we heuristically initialize γ_u with a high-pass filter based formulation (the proof of which is provided in Appendix C.1.2), as follows:

$$\gamma_u = \frac{(-\tau)^{u-1}}{\sum_{u=1}^{U} |(-\tau)^{u-1}|}, \tag{5}$$

where $\tau \in (0, 1)$ is a hyperparameter. More details regarding the initialization of γ_u can be found in the appendix.

Finally, the classification score vector of the i-th node can be obtained as follows: $\boldsymbol{v}_i = \mathrm{softmax}(\boldsymbol{W}_v\hat{\boldsymbol{x}}_i)$. Comparisons between ART and existing message passing modules can be found in the supplementary materials.

3.3. Relationship Feature Propagation

Existing SGG works typically ignore correlations between relationships. In this subsection, we propose the RFP module to use the inter-relationship connections under heterophilic settings. To the best of our knowledge, no existing heterophilic GNNs have explicitly explored the connections between edges.

An intuitive design choice for RFP is to use the same architecture as ART. However, there are $N(N-1)$ potential relationships for an image containing N objects, implying that using the same structure as ART for RFP incurs a high computational cost. Moreover, there are no meaningful relationships between the majority of object pairs. To address the above issues, we adopt two strategies. First, we only model the connections between relationships that share the same *subject* or *object*. Second, we utilize the message-passing coefficients between nodes to guide edges since the connections between relationships can be decoupled into connections between their related objects.

Specifically, the representation of one relationship $\boldsymbol{r}_{ij}$ is obtained as follows:

$$\boldsymbol{r}_{ij} = \hat{\boldsymbol{x}}_i * \hat{\boldsymbol{x}}_j * (\boldsymbol{x}_{ij} + \boldsymbol{B}_{ij}), \tag{6}$$

where $*$ denotes a fusion function defined in [23]: $\boldsymbol{x} * \boldsymbol{y} = \mathrm{ReLU}\,(\boldsymbol{W}_x\boldsymbol{x} + \boldsymbol{W}_y\boldsymbol{y}) - (\boldsymbol{W}_x\boldsymbol{x} - \boldsymbol{W}_y\boldsymbol{y}) \odot (\boldsymbol{W}_x\boldsymbol{x} - \boldsymbol{W}_y\boldsymbol{y})$. We then obtain the initial classification score vector of the relationship between the i-th and j-th nodes as follows:

$$\boldsymbol{p}_{ij}^0 = \boldsymbol{W}_p\sigma(\boldsymbol{W}_r\boldsymbol{r}_{ij}). \tag{7}$$

Subsequently, we obtain the messages passed from neighboring relationships to $\boldsymbol{r}_{ij}$ as follows:

$$\mathcal{H}(\mathcal{N}_{\boldsymbol{r}_{ij}}) = \sum\nolimits_{l\in\mathcal{N}_j} \hat{\alpha}_{jl}\boldsymbol{p}_{il} + \sum\nolimits_{m\in\mathcal{N}_i} \hat{\alpha}_{im}\boldsymbol{p}_{mj}, \tag{8}$$

where $\hat{\alpha}_{jl}$ and $\hat{\alpha}_{im}$ indicate the normalized contextual coefficient according to the elements in $\mathcal{N}_j + \mathcal{N}_i$.

To reduce the computational complexity, some GNN models [8, 11] have utilized PageRank-based approaches to propagate the label information. However, as proven in [3], these methods act as low-pass graph filters, which invariably suppress the high-frequency component, namely the heterophily. To address this issue, we formulate the output of the n-th propagation layer for the relationship $\boldsymbol{r}_{ij}$ as follows:

$$\boldsymbol{p}_{ij}^{k+1} = \beta(\boldsymbol{p}_{ij}^k + \mathcal{H}^k(\mathcal{N}_{\boldsymbol{r}_{ij}})) + (1-\beta)\boldsymbol{p}_{ij}^0. \tag{9}$$

Here, β indicates the teleport probability [3], which controls how fast Eq. (9) moves away from $\boldsymbol{p}_{ij}^{0}$. As described in Theorem 4.1 of [3], Eq. (9) could be considered to operate as a high-pass graph filter such that it allows the teleport probability β to be negative. In other words, Eq. (9) enables the model to pass relevant high-frequency graph signals, such as heterophily.

Finally, the classification score vector for the relationship between the i-th and j-th nodes can be written as follows:

$$\boldsymbol{t}_{ij} = \mathrm{softmax}(\boldsymbol{p}_{ij}^{K} + \boldsymbol{f}_{ij}), \tag{10}$$

Here, K denotes the number of RFP layer. $\boldsymbol{f}_{ij}$ indicates the relationship distribution vector between the object categories of the i-th and j-th nodes in the training set. It functions in the same way as frequency bias and has been widely adopted in existing works [12, 14, 30, 32].

3.4. Heterophily-aware Message Passing

Heterophily causes GNNs to experience performance degradation. Recent works [29] in GNN architecture design mitigate this problem by allowing the messages from interclass neighbors to be multiplied by a negative sign. This operation enables the mean distance between inter-class nodes to be less affected in the aggregation procedure. To better distinguish between the homophily and heterophily in the scene graph, especially within the complicated visual scene (*i.e.*, occlusion), we define a sign function to adjust the non-negative contextual coefficient between nodes or edges. Furthermore, this sign function indicates whether they belong to the same category. In more detail, we formulate the sign message between two nodes with the features of their union box, defined as follows:

$$s_{ij} = \tanh(\boldsymbol{w}_s^T \sigma(\boldsymbol{W}_s[\boldsymbol{x}_{ij} + \boldsymbol{B}_{ij}, \boldsymbol{v}_i, \boldsymbol{v}_j])), \tag{11}$$

where $[,]$ represents the concatenation operation. Tanh is utilized to approximate the sign function and has the additional benefit of being differential. In the training process, a binary cross-entropy (BCE) loss is utilized for supervision with ground-truth sign labels $y_{ij}^{s} \in \{-1, 1\}$; here, 1 and -1 indicate that the two nodes belong to the same and different object categories, respectively. By integrating the sign information into Eq. (2), the message between two nodes can be refined as follows:

$$\mathcal{F}(\mathcal{N}_i) = \sum\nolimits_{j \in \mathcal{N}_i} s_{ij} \alpha_{ij} \sigma(\boldsymbol{W}_{\mathcal{F}} \mathrm{LN}(\boldsymbol{x}_j)). \tag{12}$$

Similarly, the sign function for two neighboring edges with the same *subject* or *object* can be approximated as follows:

$$\begin{aligned} q_{il \to ij} &= \tanh(\boldsymbol{w}_q^T(\hat{\boldsymbol{x}}_i * \boldsymbol{x}_{lj} * (\boldsymbol{x}_{ijl} + \boldsymbol{B}_{i,lj}))) \\ q_{mj \to ij} &= \tanh(\boldsymbol{w}_q^T(\boldsymbol{x}_{im} * \hat{\boldsymbol{x}}_j * (\boldsymbol{x}_{ijm} + \boldsymbol{B}_{im,j}))), \end{aligned} \tag{13}$$

where $q_{il \to ij}$ denotes two edges $\boldsymbol{r}_{ij}$ and $\boldsymbol{r}_{il}$ that share the same *subject*, while $q_{mj \to ij}$ indicates two edges $\boldsymbol{r}_{ij}$ and $\boldsymbol{r}_{mj}$ that share the same *object*.

A BCE loss is adopted for the supervision on $q_{il \to ij}$ and $q_{mj \to ij}$, respectively. The ground-truth labels are 1 and -1 for two edges that belong to the same and different categories, respectively. Finally, the sign information is utilized to refine the messages between neighboring edges defined in Eq. (8) as follows:

$$\mathcal{H}(\mathcal{N}_{\boldsymbol{r}_{ij}}) = \sum_{l \in \mathcal{N}_j} q_{il \to ij} \hat{\alpha}_{jl} \boldsymbol{p}_{il} + \sum_{m \in \mathcal{N}_i} q_{mj \to ij} \hat{\alpha}_{im} \boldsymbol{p}_{mj}. \tag{14}$$

3.5. SGG by HL-Net

During training, the overall loss function $\mathcal{L}$ of HL-Net can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_v + \mathcal{L}_e + \mathcal{L}_{bce}^{v} + \mathcal{L}_{bce}^{e}, \tag{15}$$

where $\mathcal{L}_v$ and $\mathcal{L}_e$ are the standard cross-entropy loss for object and relationship classification, respectively. Moreover, $\mathcal{L}_{bce}^{v}$ and $\mathcal{L}_{bce}^{e}$ represent the BCE loss for sign prediction in object and relationship classification, respectively.

During testing, the object category for the i-th node is predicted by the following equation:

$$o_i = \arg\max\nolimits_{o \in \mathcal{O}}(\boldsymbol{v}_i(o)), \tag{16}$$

where $\mathcal{O}$ represents the set of object categories. The relationship category of the edge between the i-th and j-th nodes can be obtained as follows:

$$e_{ij} = \arg\max\nolimits_{r \in \mathcal{R}}(\boldsymbol{t}_{ij}(r)), \tag{17}$$

where $\mathcal{R}$ represents the set of relationship categories.

4. Experiments

4.1. Dataset and Evaluation Settings

Visual Genome: We follow the same data cleaning strategy [4] that has been widely adopted in several recent works. More specifically, the most frequently occurring 150 object categories and 50 relationship categories are utilized for evaluation. The scene graph for each image consists of 11.6 objects and 6.2 relationships on average. We follow three conventional protocols for evaluation: 1) Scene Graph Detection (SGDET): Given an image, the model detects object bounding boxes and predicts both the object and relationship categories for each bounding box pair. 2) Scene Graph Classification (SGCLS): Given the ground-truth location of object bounding boxes, the model predicts both the object and relationship categories. 3) Predicate Classification (PREDCLS): Given the ground-truth object bounding

Backbone	Model	SGDET R@20	SGDET R@50	SGDET R@100	SGCLS R@20	SGCLS R@50	SGCLS R@100	PREDCLS R@20	PREDCLS R@50	PREDCLS R@100	Mean
VGG-16	IMP$^\diamond$ [4]	14.6	20.7	24.5	31.7	34.6	35.4	52.7	59.3	61.3	39.3
	MOTIFS$^\diamond$ [32]	21.4	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1	43.7
	KERN$^\diamond$ [2]	-	27.1	29.8	-	36.7	37.4	-	65.8	67.6	44.1
	GPI$^\diamond$ [5]	-	-	-	-	36.5	38.8	-	65.1	66.9	-
	VCTREE$^\diamond$ [23]	22.0	27.9	31.3	35.2	38.1	38.8	60.1	66.4	68.1	45.1
	GPS-Net$^\diamond$ [14]	22.6	28.4	31.7	36.1	39.2	40.1	60.7	66.9	68.8	45.9
	R-CAGCN$^\diamond$ [30]	22.1	28.1	31.3	35.4	38.3	39.0	60.2	66.6	68.3	45.3
	HL-Net$^\diamond$	**22.9**	28.5	31.9	37.2	39.8	40.8	61.3	67.5	69.5	46.3
	RelDN‡ [37]	-	-	32.7	-	-	36.8	-	-	68.4	-
	Seq2Seq-RL‡ [16]	22.1	30.9	34.4	34.5	38.3	39.0	60.3	66.4	68.5	46.3
	HL-Net ‡	22.5	**31.3**	**34.7**	**37.4**	**40.4**	**41.3**	**61.6**	**67.7**	**69.7**	**47.5**
RX-101	VTransE [22]	23.0	29.7	34.3	35.4	38.6	39.4	59.0	65.7	67.6	45.9
	VCTREE [23]	24.7	31.5	36.2	37.0	40.5	41.4	59.8	66.2	68.1	47.3
	MOTIFS [32]	25.1	32.1	36.9	35.8	39.1	39.9	59.5	66.0	67.9	47.0
	SGGNLS [38]	24.6	31.8	36.3	36.5	40.0	40.8	58.7	65.6	67.4	47.0
	HL-Net	**26.0**	**33.7**	**38.1**	**38.8**	**42.6**	**43.5**	**60.7**	**67.0**	**68.9**	**49.0**

Table 1. Performance comparisons with state-of-the-art methods on the VG dataset. We compute the mean on all tasks over R@50 and R@100. $^\diamond$ and ‡ denote the methods using the same Faster-RCNN detector as [32] and [37], respectively.

boxes and their object categories, the model predicts only the relationship categories. All three settings are evaluated according to Recall@K (R@K) metrics, where K is set to 20, 50, and 100, respectively.

Open Images: Open Images (OI) [10] is a large-scale dataset proposed by Google. We conduct our experiments on Open Images V4 and V6. In more detail, the Open Images V4 dataset contains 53,953 and 3,234 images as the training and validation sets, respectively. It comprises a total of 57 object categories and 9 predicate categories. Open Images V6 contains 126,368/1,813/5,322 images used for training/validation/testing, respectively. It has 301 object categories and 31 predicate categories. We follow the same data processing and evaluation protocols outlined in [12, 14, 37]. More specifically, the results are evaluated by calculating Recall@50 (R@50), the weighted mean AP of relationships (wmAP_{rel}), and the weighted mean AP of phrase (wmAP_{phr}). The final score is given by score$_{wtd}$ = $0.2\times\mathrm{R@50}+0.4\times\mathrm{wmAP}_{rel}+0.4\times\mathrm{wmAP}_{phr}$. Note that wmAP_{rel} evaluates the AP of the predicted triplet in which both the subject and object boxes have an IoU of at least 0.5 with ground truth, while wmAP_{phr} evaluates the AP of the predicted triplet where the union area of the subject and object boxes has an IoU of at least 0.5 with ground truth.

4.1.1 Implementation Details

To facilitate a fair comparison with the majority of existing works, we utilize ResNeXt-101-FPN [13, 27] as the backbone for the OI database. We adopt both ResNeXt-101-FPN [13, 27] and VGG-16 [21] as the backbones for the VG database. During training, we freeze the layers before the ROIAlign layer and optimize the remaining layers in the model using both the object and relationship classification losses. We optimize HL-Net via Stochastic Gradient Descent (SGD) with momentum, using an initial learning rate of 10^{-3} and a batch size of 6. The top-64 object proposals in each image are selected following per-class non-maximal suppression (NMS) with an IoU of 0.3. Moreover, the sampling ratio between pairs without any relationship (background pairs) and those with relationships during training is set to 3:1. We further set the teleport probability β to -0.5.

4.2. Comparisons with State-of-the-Art Methods

Visual Genome: Table 1 shows that HL-Net outperforms all state-of-the-art methods on all metrics. More specifically, HL-Net outperforms one very recent GNN-based SGG model, named R-CAGCN [30], by 1.0% on average at R@50 and R@100 over the three protocols. It further outperforms R-CAGCN [30] by 0.6 %, 1.8 %, and 1.2 % on SGDET, SGCLS, and PREDCLS at R@100, respectively. Moreover, HL-Net outperforms VCTREE [22, 23] using the same ResNeXt-101-FPN backbone by 2.1% and 1.9% on SGCLS and SGDET at R@100, respectively.

To demonstrate the effectiveness of HL-Net in exploring heterophilic information under occlusion scenarios, we propose to calculate two different R@K metrics for the SGCLS task. More specifically, we decompose the SGCLS task into two subtasks, namely C-SGCLS and S-SGCLS. The former determines the SGCLS performance on triplets where at least one object is heavily occluded by others, *i.e.*, IoU>0.5. Otherwise, we term the subtask as S-SGCLS. As shown in Table 2, HL-Net outperforms all state-of-the-art methods on both tasks. In particular, when compared with

Backbone	Method	C-SGCLS		S-SGCLS	
		R@50	R@100	R@50	R@100
VGG-16	MOTIFS [32]	32.5	33.4	35.5	36.4
	KERN [2]	33.7	34.6	36.8	37.7
	VCTREE [23]	34.9	35.9	38.0	38.9
	GPS-Net [14]	35.8	37.1	38.4	39.3
	HL-Net	**38.3**	**39.4**	**38.7**	**39.6**
RX-101	VTransE [22]	34.9	36.0	38.7	39.9
	MOTIFS [32]	35.4	36.5	39.9	40.9
	HL-Net	**41.0**	**42.2**	**41.8**	**42.7**

Table 2. SGCLS performance comparison under occlusion and non-occlusion scenarios. C-SGCLS denotes the SGCLS performance of triplets where at least one object is heavily occluded by others, otherwise, results are marked S-SGCLS.

Model	SGDET	SGCLS	PREDCLS
	mR@100	mR@100	mR@100
IMP [4]	4.8	6.0	10.5
FREQ [32]	7.1	8.5	16.0
MOTIFS [32]	6.6	8.2	15.3
KERN [2]	7.3	10.0	19.2
VCTREE-SL [23]	7.7	10.5	18.5
VCTREE-HL [23]	8.0	10.8	19.4
R-CAGCN [30]	8.8	11.1	19.9
HL-Net	**9.2**	**13.5**	**22.8**

Table 3. Performance comparison on mean recall $(\%)$ across all 50 relationship categories. All methods in this table adopt the same Faster-RCNN from [32] model as object detector.

Dataset	Model	R@50	WmAP		score$_{wtd}$
			rel	phr	
V4	RelDN [37]	74.9	35.5	38.5	44.6
	BGNN [12]	75.5	37.8	41.7	46.9
	HL-Net	**78.1**	**38.9**	**42.2**	**48.1**
V6	MOTIFS [32]	71.6	29.9	31.6	38.9
	VCTREE [23]	74.1	34.2	33.1	40.2
	RelDN [37]	73.1	32.2	33.4	40.8
	GPS-Net [14]	74.8	32.9	34.0	41.7
	BGNN [12]	75.0	33.5	34.2	42.1
	HL-Net	**76.5**	**35.1**	**34.7**	**43.2**

Table 4. Comparisons with state-of-the-art methods on OI. We adopt the same evaluation metric as in [37].

MOTIFS [32] using the same ResNeXt-101-FPN backbone, HL-Net achieves an advantage of 5.7% and 1.8% at R@100 for C-SGCLS and S-SGCLS, respectively.

Moreover, to demonstrate the robustness of HL-Net to the class imbalance problem on VG, we additionally compare its performance with that of state-of-the-art methods using the Mean Recall metric [2, 23]. As shown in Table 3, HL-Net achieves a notable absolute performance gain without specifically considering the imbalance problem in model design. These results indicate that HL-Net also has advantages to handle the class imbalance problem in SGG.

Exp	Module			SGCLS		PREDCLS	
	ART	RFP	HMP	R@50	R@100	R@50	R@100
1	-	-	-	40.1	40.9	65.5	67.3
2	✓	-	-	41.7	42.5	65.8	67.6
3	-	✓	-	40.6	41.3	66.4	68.3
4	-	-	✓̽	41.2	41.9	65.9	67.7
5	✓	✓	-	41.8	42.7	66.6	68.5
6	-	✓	✓	41.3	42.1	66.8	68.7
7	✓	-	✓̽	42.4	43.3	66.1	67.9
8	✓	✓	✓	**42.6**	**43.5**	**67.0**	**68.9**

Table 5. Ablation studies. We consistently adopt the same object detection backbone as in [22]. "✓̽" denotes that we only apply HMP to refine the representation of objects.

Open Images: Table 4 compares the performance of HL-Net with state-of-the-art methods. RelDN is an improved version of the model that won the Google Open Images Visual Relationship Detection Challenge V4. Using the same object detector, HL-Net outperforms RelDN [37] by 3.5% and 2.4% on the overall metric score$_{wtd}$ for OI V4 and V6, respectively. In more detail, on OI V4, HL-Net outperforms RelDN [37] by 3.2%, 3.4%, and 3.7% at R@50, wmAP$_{rel}$, and wmAP$_{phr}$, respectively. Moreover, when compared with other approaches that use the same backbone on OI V6, HL-Net consistently achieves the best performance.

4.3. Ablation Studies

Effectiveness of the Proposed Modules. We first perform an ablation study to validate the effectiveness of ART, RFP, and HMP, respectively. The results are summarized in Table 5. Details of the baseline can be found in Appendix D.2. From Exps 1-8, we can clearly see that the performance improves consistently when more modules are involved. This shows that each module is helpful in promoting the performance of SGG.

Design Choices in ART and RFP. We verify the impact of hyperparameters on the ART and RFP modules. As shown in Table 6(a), HL-Net achieves the best performance when τ is set to 0.5 in Eq. (5). In Table 6(b), we compare the performance of HL-Net with different numbers of ART layers, ranging from two to five; it is evident that the performance of HL-Net improves with an increasing number of ART layers (due to limitations on GPU memory size, we only conducted experiments up to five ART layers). In Table 6(c), we compare the performance of HL-Net with different numbers of RFP layers, ranging from two to five; it is shown that the best performance is achieved when the number of RFP layers is set to four.

Qualitative Evaluation. Figure 3 presents a qualitative comparison between HL-Net and MOTIFS [32]. As can

		$\tau = 0.2$	$\tau = 0.5$	$\tau = 0.7$
SGCLS	R@20	38.5	**38.8**	38.6
	R@50	42.3	**42.6**	42.4
	R@100	43.2	**43.5**	43.3

(a) Evaluation on the value of τ in Eq. (5).

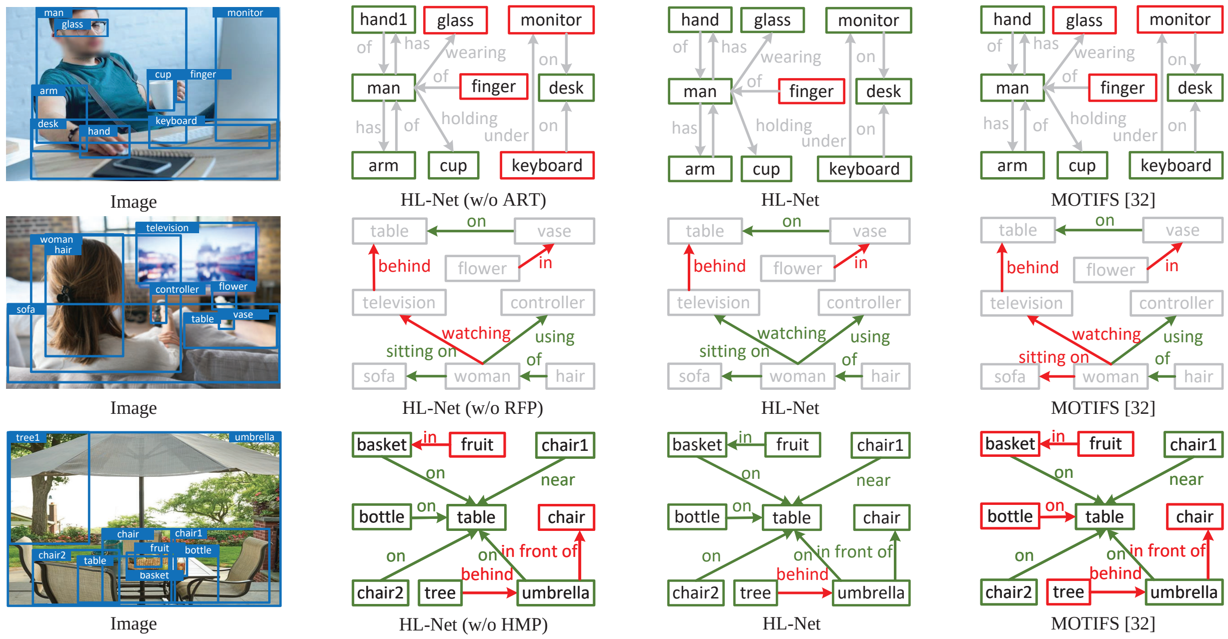
		2-step	3-step	4-step	5-step
SGCLS	R@20	38.1	38.3	38.6	**38.8**
	R@50	41.9	42.1	42.3	**42.6**
	R@100	42.8	43.0	43.3	**43.5**

(b) Evaluation on the number of ART layers U.

		2-step	3-step	4-step	5-step
PREDCLS	R@20	60.3	60.5	**60.7**	60.4
	R@50	66.6	66.8	**67.0**	66.7
	R@100	68.5	68.7	**68.9**	68.6

(c) Evaluation on the number of RFP layers K.

Table 6. The impact of hyperparameters on the ART and RFP modules, respectively.



Image

HL-Net (w/o ART)

HL-Net

MOTIFS [32]

Image

HL-Net (w/o RFP)

HL-Net

MOTIFS [32]

Image

HL-Net (w/o HMP)

HL-Net

MOTIFS [32]

Figure 3. Qualitative comparisons between HL-Net and MOTIFS [32]. Specifically, we show the comparisons at R@100 in the SGCLS setting in the first and third rows. In the second row, we show the comparisons at R@100 in the PREDCLS setting. The green color indicates correctly classified objects or predicates; the red indicates those that have been misclassified. Best viewed in color.

be seen from the first row of Figure 3, HL-Net makes better predictions than MOTIFS for "monitor" and "keyboard" that are hard to recognize from its proposal. Therefore, we owe this performance gain to the ART module that utilizes the heterophilic context to refine the node prediction. In the second row of Figure 3, it is shown that HL-Net can identify "watching" by inferring from their neighboring ones. We give this credit to the RFB module. Finally, in the third row of Figure 3, it can be observed that HL-Net has clear advantages in predicting the categories of both nodes and edges under heavy occlusion scenarios (*e.g.*, "umbrella in front of chair"), via the HMP scheme.

4.4. Conclusion and Limitations

Scene graphs are naturally heterophilic. In this paper, we devise HL-Net to comprehensively explore homophily and heterophily for both object and relationship prediction in the SGG. More specifically, the heterophily between nodes is encoded in the message passing via an Adaptive Reweighting Transformer module. The connections between heterophilic relationships are explored by means of a Relationship Feature Propagation module. Moreover, the heterophily and homophily between objects and those between relationships in complicated visual scenes are considered using a Heterophily-aware Message Passing scheme. Extensive experiments on two popular databases justify the effectiveness of HL-Net for SGG. The same as the majority of existing SGG models, one limitation of our method is its dependency on sufficiently labeled data. In the future, we will explore how to train the HL-Net more robustly in the face of a large number of missing annotations.

Broader Impacts. SGG can potentially provide valuable assistance for many real-world applications (*e.g.*, autonomous driving). To the best of our knowledge, our work is not harmful in ethical aspects or with future societal consequences.

Acknowledgment. This work is supported by the NSF of China (Nos: 62076101, 62002090), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No: 2017ZT07X183).

References

[1] J. Ba, J. Kiros, and G. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[2] T. Chen, W. Yu, R. Chen, and L. Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, 2019. 1, 2, 3, 6, 7

[3] E. Chien, J. Peng, P. Li, and O. Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021. 2, 3, 4, 5

[4] D.Xu, Y. Zhu, C. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017. 2, 5, 6, 7

[5] R. Herzig, M. Raboh, G. Chechik, J. Berant, and A. Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *NeurIPS*, 2018. 6

[6] k. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka. Representation learning on graphs with jumping knowledge networks. In *ICML*, 2018. 4

[7] T. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2

[8] J. Klicpera, A. Bojchevski, and S. Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2019. 2, 4

[9] R. Krishna, Y. Zhu, O. Groth, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 2

[10] A. Kuznetsova, H. Rom, et al. The open images dataset v4. *IJCV*, 2020. 2, 6

[11] P. Li, E. Chien, and O. Milenkovic. Optimizing generalized pagerank methods for seed-expansion community detection. In *NeurIPS*, 2019. 4

[12] R. Li, S. Zhang, B. Wan, and X. He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, 2021. 1, 2, 3, 5, 6, 7

[13] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[14] X. Lin, C. Ding, J. Zeng, and D. Tao. Gps-net: Graph property sensing network for scene graph generation. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6, 7

[15] M. Liu, Z. Wang, and S. Ji. Non-local graph neural networks. *arXiv preprint arXiv:2005.14612*, 2020. 2

[16] Y. Lu, H. Rai, J. Chang, B. Knyazev, G. Yu, S. Shekhar, G. Taylor, and M. Volkovs. Context-aware scene graph generation with seq2seq transformers. In *ICCV*, 2021. 6

[17] Y. Ma, X. Liu, N. Shah, and J. Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021. 1, 2

[18] H. Pei, B. Wei, K. Chang, Y. Lei, and B. Yang. Geom-gcn: Geometric graph convolutional networks. In *ICLR*, 2019. 2, 3

[19] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 3

[20] I. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *SPM*, 2013. 2, 4

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[22] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang. Unbiased scene graph generation from biased training. In *CVPR*, 2020. 6, 7

[23] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019. 2, 4, 6, 7

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2

[26] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph convolutional networks. In *ICML*, 2019. 4

[27] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 6

[28] R. Xiong, Y. Yang, D. He, et al. On layer normalization in the transformer architecture. In *ICML*, 2020. 2, 4

[29] Y. Yan, M. Hashemi, K. Swersky, Y. Yang, and D. Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv preprint arXiv:2102.06462*, 2021. 1, 2, 3, 5

[30] G. Yang, J. Zhang, Y. Zhang, B. Wu, and Y. Yang. Probabilistic modeling of semantic ambiguity for scene graph generation. In *CVPR*, 2021. 1, 2, 5, 6, 7

[31] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018. 1, 2, 3

[32] R. Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 2, 3, 5, 6, 7, 8

[33] Y. Zhan, J. Yu, T. Yu, and D. Tao. On exploring undetermined relationships for visual relationship detection. In *CVPR*, 2019. 2

[34] Yibing Zhan, Jun Yu, Ting Yu, and Dacheng Tao. Multi-task compositional network for visual relationship detection. *International Journal of Computer Vision*, 128(8):2146–2165, 2020. 2

[35] H. Zhang, Z. Kyaw, S. Chang, and T. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. 2

[36] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal. Relationship proposal networks. In *CVPR*, 2017. 2

[37] J. Zhang, K. Shih, A. Elgammal, A. Tao, and B. Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, 2019. 6, 7

[38] Y. Zhong, J. Shi, J. Yang, C. Xu, and Y. Li. Learning to generate scene graph from natural language supervision. In *ICCV*, 2021. 6

[39] K. Zhou, Y. Dong, K. Wang, W. Lee, B. Hooi, H. Xu, and J. Feng. Understanding and resolving performance degradation in graph convolutional networks. *arXiv preprint arXiv:2006.07107*, 2020. 2

[40] J. Zhu, R. Rossi, A. Rao, T. Mai, N. Lipka, N. Ahmed, and D. Koutra. Graph neural networks with heterophily. *arXiv preprint arXiv:2009.13566*, 2020. 4

[41] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*, 2020. 1, 2, 3

[42] B. Zhuang, L. Liu, C. Shen, and I. Reid. Towards context-aware interaction recognition for visual relationship detection. In *ICCV*, 2017. 2