Heterogeneous Learning for Scene Graph Generation

Yunqing He
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China
heyq@smail.nju.edu.cn

Tongwei Ren*
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China
rentw@nju.edu.cn

Jinhui Tang
School of Computer Science, Nanjing University of Science and Technology
Nanjing, China
jinhuitang@njust.edu.cn

Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China
gswu@nju.edu.cn

ABSTRACT

Scene Graph Generation (SGG) task aims to construct a graph structure to express objects and their relationships in a scene at a holistic level. Due to the neglect of heterogeneity of feature spaces between objects and relations, coupling of feature representations becomes obvious in current SGG methods, which results in large intra-class variation and inter-class ambiguity. In order to explicitly emphasize the heterogeneity in SGG, we propose a plug-and-play Heterogeneous Learning Branch (HLB), which enhances the independent representation capability of relation features. The HLB actively obscures the interconnection between objects and relation feature spaces via gradient reversal, with the assistance of a link prediction module as information barrier and an Auto Encoder for information preservation. To validate the effectiveness of HLB, we apply HLB to typical SGG methods in which the feature spaces are either homogeneous or semi-heterogeneous, and conduct evaluation on VG-150 dataset. The experimental results demonstrate that HLB significantly improves the performance of all these methods in the common evaluation criteria for SGG task.

CCS CONCEPTS

• **Computing methodologies** → **Scene understanding**.

KEYWORDS

Heterogeneous learning, scene graph generation, feature representation, relation prediction

ACM Reference Format:
Yunqing He, Tongwei Ren, Jinhui Tang, and Gangshan Wu. 2022. Heterogeneous Learning for Scene Graph Generation. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3548356

*Corresponding Author.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548356

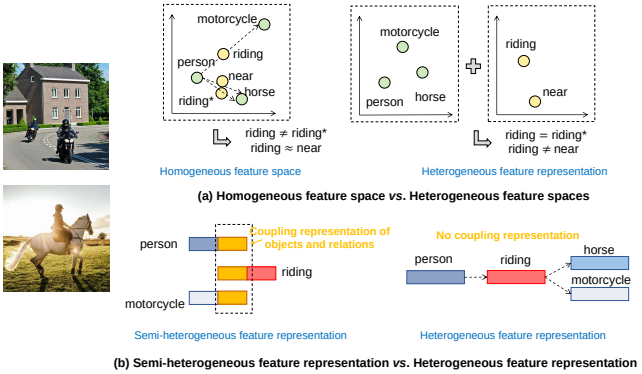

Figure 1: Feature representations in different feature spaces. (a) Large intra-class variation and inter-class ambiguity in Homogeneous feature spaces. (b) Semantic ambiguity between semi-heterogeneous feature representation.

1 INTRODUCTION

Scene Graph Generation (SGG) aims to represent objects and their relationships with a graph structure, which plays a fundamental role in numerous downstream applications, such as captioning [33], multi-modal dialog [16] and visual question answering [1]. Traditional relation prediction [11] treats objects and relations as isolated entities. They only use local contexts in relation prediction, which is insufficient for formatting comprehensive understanding of scenes. To fully exploit global contexts, current SGG methods [2, 15] fuse feature representations of objects and relations, so as to improve the performance of both object and relation prediction.

However, current SGG methods usually fuse the features of objects and relations into the same feature space, which may cause the problem of large intra-class variation and inter-class ambiguity. In addition, co-relative object and relation features may also result in semantic ambiguity. Figure 1 shows the feature spaces modeling with different degree of heterogeneity in SGG methods. As shown in Figure 1(a), if objects and relations are represented in same feature space, we claim these methods have *homogeneous*

feature space. In these approaches, the same predicate like "riding" varies widely due to distinct objects. Meanwhile, relationships between same subjects and objects are more likely to generate approximate representations. These homogeneous approaches are conducted without any information barrier which prevents messages from parsing to another feature space. Some other methods, namely Semi-heterogeneous ones, do not assume that objects and relations share same feature space. Although some variances are implicitly produced by these inactivated neurons or blocked information channels, relevance still exists between object and relation feature spaces. As shown in Figure 1(b), inter-class ambiguity is also allowed. Meanwhile, absurd meaningless associations exist due to the transitivity of such relevance, *e.g.*, motorcycles and forks may be relative for their potential relations with person.

To this end, we propose a Heterogeneous Learning Branch (HLB) to explore the heterogeneity in SGG methods. We analyze the general architecture of SGG methods, and summarize three steps for refining the generation of scene graphs with heterogeneity constraints. Firstly, we attempt to construct two independent features spaces for object and relation representations with Gradient Reversal Layer (GRL). The relation features is then confused to predict object categories by GRL, *i.e.*, object-related information is ambiguous in relation feature space. Secondly, we design an Auto Encoder to preserve the relevance information between relation feature space and probability distribution space. It also ensures the relation feature space can be constructed in a certain direction since the number of potential mutual-independent feature spaces is infinite. Finally, we propagate the heterogeneity to arbitrary relation predictors via a Gaussian mixture modeling assumption. In order to validate the significance of heterogeneity in SGG task, we de-homogenize a typical homogeneous method IMP [29] with the principle of minimum modification, which performs 30%+ better than previous IMP. We also re-train several SGG methods with HLB on VG dataset and prove that the performance of relation prediction, especially in Scene Graph Detection (SGDet) task, can be significantly improved by HLB.

Our contributions can be summarized as: To the best of our knowledge, we are the first to explore the effect of heterogeneity in SGG task. We propose a novel Heterogeneous Learning Branch (HLB), which functions as a lightweight attachment to refine the performance of mainstream SGG methods. Specifically, we propose a self-directed relation feature space construction method which simultaneously retains low-relevance and high-relevance with object feature space and semantic relationships, respectively. We also construct a Link Prediction Module in a hierarchical way to predict potential relationships between detected objects. We conduct extensive experiments on several modern SGG methods, and achieve SOTA performance in VG-150 dataset.

2 RELATED WORK

2.1 Scene Graph Generation

Research on Scene Graph Generation, in general, can be divided into four categories or combinations in the aspect of methodology. Firstly, some researches explore simple and efficient methods with basic feature transformation and combination in model design [17]. Features are extracted by a convolution network and then fed into a classifier for straightforward relation prediction [26]. These methods usually concentrate on complex feature engineering [18] or loss function [38]. Secondly, RNN-based architectures take scene graphs as special sequences for correlation learning [10]. Motifs directly transforms the graph structure to a linear sequence [36]. Thirdly, GNN-based methods are naturally suitable for scene graph generation [13, 14, 28, 32]. Many modern GNN methods collect information from neighboring nodes, and are robust to generate unseen combinations of relation triplets. It is crucial for SGG since there are only 4.22% seen triplets even in the largest dataset Visual Genome. Finally, some modern strategies in knowledge graph have been adopted by SGG [37]. The leverage of external knowledge is also classified into this type for constructing the knowledge graph from commonsense [27, 35].

We also investigate into some other modern plug-and-play methods for SGG. TDE is an attachment in the inference period of SGG, which directly manipulates the generated scene graphs [23]. Since the purpose of this method is to solve the problem of long tail distribution in SGG, substantial reduction in Recall metric is presented, which means the real-world data distribution is neglected. PUM is another inference-period tool which optimizes the relation prediction results by whitening the distribution of data [31]. EMB proposes an energy-based model to substitute the Binary Cross Entropy Loss [22]. This method is effective but still requires modification to original SGG method. In contrast, we propose a heterogeneous learning method, which can easily improve the performance by 10%+ without any additional inference cost or any modification to the framework and network structure of the original methods.

2.2 Heterogeneous Learning

The concept of Heterogeneity is usually used in big-data and real-world scenario, *e.g.*, Commodity Recommendation System. Most of the existing Heterogeneous Learning methods for Graph Learning and Knowledge Graph only deal with undirected graph and few relationship categories, which is not sufficient for SGG. RGCN learns weighted matrices for each type of nodes [20]. WGCN divides the overall graph into several weighted sub-graph, and only include one type of relation in each sub-graph [21]. Recent CompGCN is the first to learn relation embedding and is, to some extent, capable of relation prediction [25].

Heterogeneity in SGG consists of two main types. One is the heterogeneity between distinct relations in semantics. For instances, vtranse classifies relations into verb, spatial-type, preposition, and comparative [37]. PUM leverages synonymy, hyponymy, and multi-view ambiguity to handle the coupling and bias between relations [31]. Another type of heterogeneity in SGG is object-relationship heterogeneity, which have not been discussed yet, and therefore faces more challenges than general heterogeneous learning studies. Generally, heterogeneous learning aims to control the unbalanced message parsing with weighted information fusion between different types of nodes, and further studies the representations of different nodes. However, in SGG, there is naturally a strong coupling between the feature representations of objects and relations, because the relations themselves are invisible and often need to be computed by using the visual features of the objects. In other words, the SGG approach is inherently unfriendly to heterogeneity.

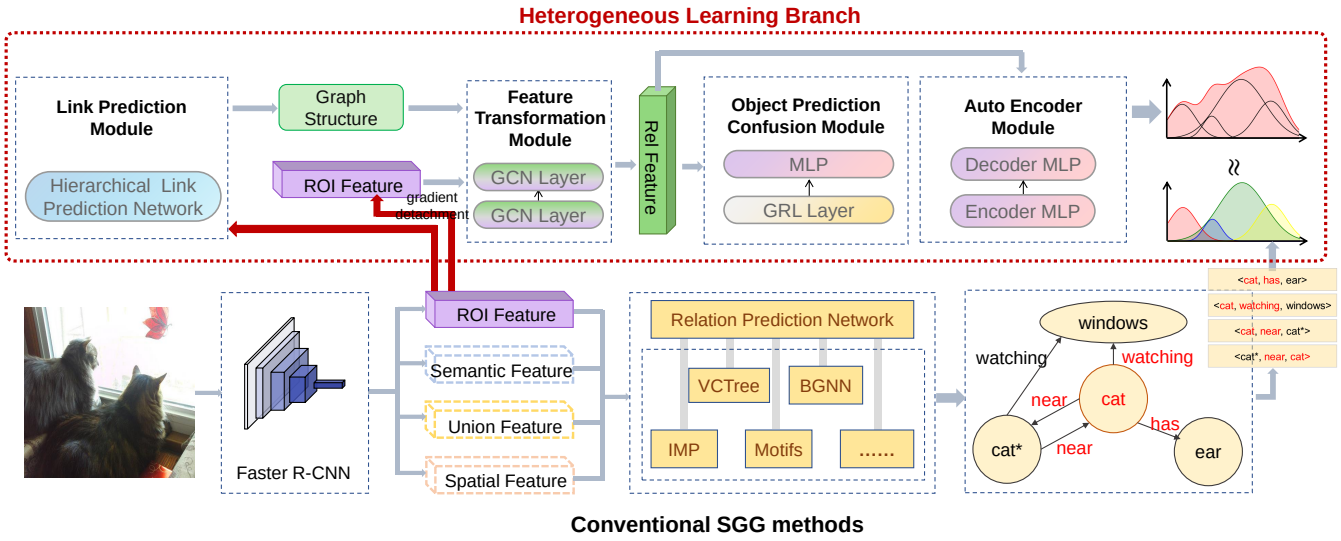

Figure 2: The overview of our method. We leverage visual features of objects (ROI Feature) to construct relation feature space via Feature Transformation Module. The heterogeneity of relation feature space is guaranteed by Object Prediction Confusion Module. Link Prediction Module is an auxiliary section for feature transformation, which diminishes the co-relationship between unrelated objects. An Auto-Encoder is attached following the Feature Transformation Module, including an encoder to fit the results from conventional SGG models and a decoder to preserve information in relation probabilities.

3 OUR APPROACH

3.1 Problem Setting

A standard Scene Graph is usually acknowledged as a directional graph $\mathcal{G}=(\mathcal{O},\mathcal{R})$, where objects are denoted as $\mathcal{O}=\{o_i \mid i \in [1...n]\}$ and relations between objects denoted as $\mathcal{R}=\{r_i \mid i \in [1...m]\}$. Here n and m present the number of object and relation instances respectively. To be specific, for $\forall o_i \in \mathcal{O}$, there exists $o_i=\{b_i, c_i^o \mid b_i \in \mathbb{R}^4, c_i^o \in C^o\}$, where b_i indicates bounding box coordinates, and for $\forall r_i \in \mathcal{R}$, there exists $r_i=\{c_i^r \mid c_i^r \in C^r\}$. C^o and C^r refer to object and relation categories, respectively. Even though relations in real-world are of extremely complexity, in usual researches, only the most 'salient' relation will be taken into consideration. In other words, SGG is modeled as an one-shot problem.

In our research, the heterogeneity constraint is explicitly used in the progress of scene graph generation. We assume that the heterogeneity is mainly expressed in the hidden feature space $\mathcal{F}$ other than the ultimate semantic scene graph $\mathcal{G}$. In other words, prior knowledge still works in heterogeneous SGG methods. Supposing that the object and relation feature spaces are respectively represented as $\mathcal{F}^{obj}=\{f_i^{obj} \mid i \in [1...m]\}$ and $\mathcal{F}^{rel}=\{f_i^{rel} \mid i \in [1...m]\}$, our objective can be described as for $\forall f_i^{obj} \in \mathcal{F}^{obj} and \forall f_j^{rel} \in \mathcal{F}^{rel}$, and there exists no $\mathcal{K} = \{k_i \mid i \in [1...r] and\ \exists\ k_i \neq 0\}$ that satisfies:

$$\begin{cases} f_i^{obj} = k_1 f_1^{rel} + k_2 f_2^{rel} + ... + k_m f_m^{rel}, & if \quad r = m, \\ f_j^{rel} = k_1 f_1^{obj} + k_2 f_2^{obj} + ... + k_n f_n^{obj}, & if \quad r = n. \end{cases} \tag{1}$$

Briefly, the vectors in one feature space are linearly independent with any vector set in another feature space.

However, the strict mathematical linearly independence is not a reachable optimization goal, because the number of features in the overall dataset is much larger than that of the feature dimension. Considering that the $\mathcal{F}^{obj}$ is highly related with C^o, we further translate the optimization problem to depress the relevance between the relation features $\mathcal{F}^{rel}$ and the object labels C^o. Finally, we define the heterogeneous learning problem as studying a relation feature space which has low relevance with object labels.

3.2 Architecture

Conventional SGG framework consists of an object detector and a relation predictor, and we extend the framework with an Heterogeneous Learning Branch (HLB). The overview of our method architecture is shown in Figure 2.

Generally, an object detector works as the basic backbone network to detect possible objects in scenes, and is usually implemented as a Faster R-CNN network [8, 19]. For fair comparison, we also adopt the Faster R-CNN pre-trained on ImageNet [4] as our object detector. In general, an object detector can provide some useful object-relative information, including instance-level spatial localization, ROI Features [9], predicted object labels, and also union features of paired objects. Some methods will further construct diverse representations according to these features, *e.g.*, word embedding from object labels and relative spatial features from bounding box coordinates [6]. For the sake of generality, we only leverage the ROI Features, which is used in almost every SGG method, as the input of the HLB.

Furthermore, we select several representative methods as the relation prediction network and integrate them with the HLB to validate the effectiveness of our approach. In the early period of training, these relation predictors will generate semi-heterogeneous scene

graphs. The produced scene graphs are described as mess of relation triplets $< subject, predicate, object >$. These triplets are collected and grouped by corresponding object instances to fit the calculation result of HLB, so as to learn the heterogeneity constraints. Finally, the scene graph generation progress is restricted by heterogeneity constraints via the HLB attachment.

3.3 Feature Transformation

The purpose of the Feature Transformation Module is to generate relation representation vectors from visual object information. In general, the transformation progress from object representations to relation features can be functioned as follows:

$$\{f_1^o, f_2^o, ..., f_n^o\} \hookrightarrow \{f_{1,1}^r, f_{1,2}^r, ..., f_{n,(n-1)}^r\}, \tag{2}$$

where f_i^o is the feature vector of i-th object and $f_{j,k}^r$ is the representation of relationship between j-th and k-th object. It should be noted that there exist two latent defects in Eq. (2). Firstly, the cost of calculation and data storage will grow exponentially with the number of objects increases. Even though generating relation feature vector for each object pair seems to be inevitable in common SGG method, we suppose that it is unnecessary in our HLB framework, because it exactly does not predict relation labels directly. Secondly, since the $n(n-1)$ relationship features is derived from less object instances, redundant information is included in the former ones. For instance, when the region of a person in an image is pictured as a bounding box, we usually describe the overall behavior like 'holding and eating on seat', other than single actions. In this case, we assume that relation features transformed from object visual features are appropriate representations of the overall relationships of corresponding objects. In this way, we re-formulate the transformation progress as follows:

$$\{f_1^o, f_2^o, ..., f_n^o\} \hookrightarrow \{f_1^{R_{o_1}}, f_2^{R_{o_2}}, ..., f_n^{R_{o_n}}\}, \tag{3}$$

where R_{o_i} means the collection of relationships of i-th object with other objects.

According to Eq. (3), we adopt Graph Convolution Network (GCN) as the implementation of Feature Transformation Module. Graph-based Deep Neural Networks are often affected by the problem of over-smooth. Consequently, each graph node tends to share similar feature representations with the training proceeding and the network layers deepening, especially in long-tailed datasets, where severe biased training progress exacerbates the over-smooth problem. SGG is a typical long-tailed problem with both biased labels and biased object-predicate combination. To tackle this problem, we investigate into the over-smooth-proof formulation of the Feature Transformation Module. Inspired by [7], we formulate the message propagation function as follows:

$$\mathbf{x}_i' = \sigma(\omega_1 \cdot \text{mean}_{j \in \mathcal{N}(i)}(\mathbf{x}_j) + \omega_2 \cdot \mathbf{x}_i), \tag{4}$$

where $\mathbf{x}_i'$ indicates the aggregation result of node $\mathbf{x}_i$. The set of neighboring nodes of $\mathbf{x}_i$ is represented as $\{\mathbf{x}_j \mid j \in \mathcal{N}(i)\}$. ω_1 and ω_2 refer to learnable weights. In Eq. (4), the sub-term of $\omega_2 \cdot \mathbf{x}_i$ is utilized as a over-smooth-proof design. We attempt to prevent over-smooth by retaining the information characteristics of each node as much as possible when fusing the messages from neighboring nodes. We also conduct some experiments to validate this simple but effective design.

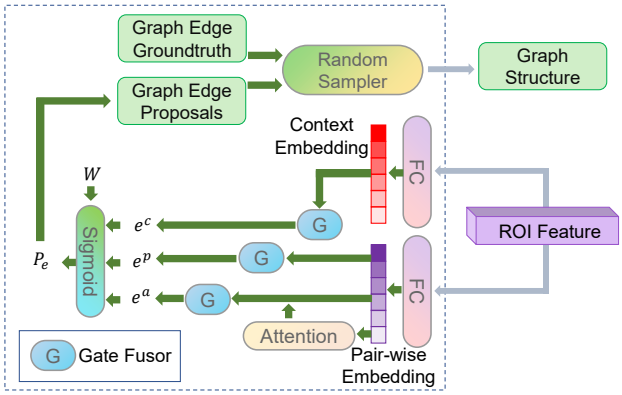

Figure 3: Network Structure of Link Prediction Module

3.4 Link Prediction

In period of Scene Graph Detection, object candidates derived from the object detector are much more complex and numerous than those in period of Scene Graph Prediction, in which the locations and number of objects are restricted. Some GNN-based methods have noticed that sparse graph connections are beneficial to scene graph generation, *e.g.*, [32] cuts down rare-appeared edges on the basis of statistic prior, and [13] leverages a MLP branch to predict confidence scores of each edge.

We construct a hierarchical Link Prediction Module to diminish the co-relationship between unrelated objects. We assume that in a scene graph, the association between an object and other entities can be observed in a three-levels way. The basic judgment on whether there is a possible link can be drawn from simply concatenating the features of object pairs. However, these concatenated pair-wise features may interrelate with each other, *e.g.*, a person who has high-confident relationship with a motorcycle, is less likely to be related to food. In addition, the overall scenario might also affect the concrete relationships of objects.

Given ROI features $\mathcal{F}^{roi} = \{f_1, f_2, ..., f_N\}$, we firstly calculate the global context representation for current scene, which consists of N predicted objects. The context $\mathcal{F}^c$ and context probability e^c is constructed as in Eq. (5), where ω^c is a learnable parameter and $\mathcal{G}$ is a gate fusion function which maps feature vectors to single-value probabilities.

$$e^c = G^c(\mathcal{F}^c) = G^c(\frac{\sum(f_i)}{N} \cdot \omega^c). \tag{5}$$

Further, pair-wise embedding features of object i and j are initialized as follows:

$$\mathcal{F}_{i \to j}^p = \omega^e \cdot (f_i \oplus f_j). \tag{6}$$

Another gate fusor is adopted to calculate the probability e^p, which indicates whether there should be an edge between object i and j. Finally, we introduce a self-attention module to analyze the co-relationship e^a between each object pair. Since the purpose of gate fusor is merely a transformation from feature representations in hidden layer to probability of link existence, a fully connection

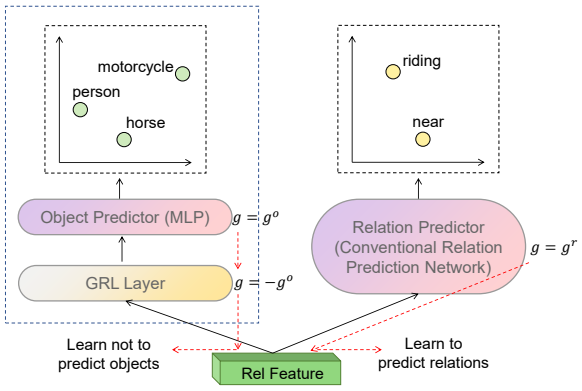

Figure 4: Principle of Object Prediction Confusion Module

layer with single-digit output is adopted as a simple implementation. The overall link prediction function is shown in Eq. (7):

$$P_e = \sigma(\omega^{\mathrm{L}}(e^c + e^p + e^a)), \tag{7}$$

where ω^{L} works as a global interceptor to determine the approximate proportion of the number of possible links, and σ is an activation function.

3.5 Object Prediction Confusion

The Object Prediction Confusion Module is aimed at confounding the Feature Transformation Module in the aspect of object prediction. As shown in Figure 4, we leverage a Gradient Reversal Layer (GRL) to achieve this goal. The GRL is originally proposed for Domain Adaptation [5], which simply reverses the gradient in the period of backwards propagation without any other computation. In this case, the network layers before GRL are misdirected to opposite gradient descent direction. To avoid the risk of destruction of entire model, the influence of GRL is strictly restricted by gradient detachment, and only the Feature Transformation Module is confused.

Following the setting of [5], we implement the GRL as Eq. (8):

$$g' = -(\frac{2}{1 + exp(-\epsilon \cdot iter)} - 1)g, \tag{8}$$

where g is the gradient of object predictor in the period of backward propagation, and g' is used to update weights of the Feature Transformation Module. The $iter$ denotes the number of iteration steps in training progress, and ϵ is a small number of hyperparameter which decides how fast the g' will be closed to $-g$. This formula ensures that the classifier has enough time to achieve better classification ability since the early performance of classifier will not introduce too much noise into the training of the Feature Transformation Module.

3.6 Auto Encoder

Despite that the Object Prediction Confusion Module has confirmed the heterogeneity of object and relation feature spaces, the similarity between these calculated feature spaces and their prediction target distributions is still not constrained. The similarity between object feature space and the object labels is usually restricted by a refine object prediction branch in conventional SGG methods. Thus we leverage a similar relation prediction branch to limit the relation feature space. This branch is implemented as a three-layers classifier, *i.e.*, the encoder in an Anto Encoder. Since the relation feature space is semantically defined as the overall behaviors of specific object, the output logits of the encoder should be similar to the sum of relationships of objects. Therefore, the optimization goal of the encoder is formulated as:

$$\text{minimize}(P_{o_i}^{collection} - (\mathcal{T}(P_{(o_i,o_1)}) + ... + \mathcal{T}(P_{(o_i,o_n)}))), \tag{9}$$

where $P_{o_i}^{collection}$ refers to the output logits of i-th object from the encoder and $P_{(o_i,o_1)}$ is the relation prediction results form conventional SGG methods. $\mathcal{T}$ is a threshold function to eliminate low-confident prediction results.

We also demonstrate that why an Auto-Encoder is adopted other than a single classifier. Although the relevance between the HLB and the conventional SGG methods is constrained by the encoder, some latent problems still exist in the optimization progress. In Eq. (9), the optimization target can be summarized as $minimize(A - B)$. However, if the optimization function collapse to the form of $minimize(A) + minimize(B)$, the value of $(A-B)$ is also minimized outwardly. We assume that the progress of collapse indicates that many valuable information is forgotten by the encoder. In this way, a decoder is proposed to reconstruct the Relation Feature that fed into the encoder.

3.7 Training Loss

The training loss of our method is deigned as Eq. (10):

$$\mathcal{L}_{total} = \mathcal{L} + \mathcal{M}_L + \mathcal{M}_C + \mathcal{N}_D + \mathcal{N}_R, \tag{10}$$

where the sum of losses of conventional SGG methods is presented as $\mathcal{L}$, which is customized by distinct relation predictors. $\mathcal{M}_L$ refers to the link loss of link prediction, and $\mathcal{M}_C$ is confusion loss in Object Prediction Confusion Module. The loss function $\mathcal{M}$ is calculated as following:

$$\mathcal{M} = -\sum_{i=1}^{|\Phi|} \gamma_r^i \log(\hat{\rho}_r^i) + (1 - \gamma_r^i) \log(1 - \hat{\rho}_r^i), \tag{11}$$

where the prediction results r is represented with a binary vector γ_r, whose length $|\Phi|$ is equal to that of the grounding truth $\hat{\rho}$. Moreover, the loss function $\mathcal{N}$ calculates the cosine similarity between prediction results and actual values:

$$\mathcal{N} = 1 - \frac{\gamma_r^i \cdot \hat{\rho}_r^i}{||\gamma_r^i|| \cdot ||\hat{\rho}_r^i||}, \tag{12}$$

where $\mathcal{N}_D$ and $\mathcal{N}_R$ respectively denote the distribution loss of the encoder and the reconstruction loss of the decoder in the Auto Encoder.

4 EXPERIMENTS

4.1 Dataset and Evaluation Metrics

SGG models are usually evaluated in three tasks, namely Predicate Classification (PredCls), Scene Graph Classification (SGCls) and Scene Graph Detection (SGDet). SGDet is the most basic metric for SGG performance evaluation. It assumes that only raw images are visible during inference period. In this way, both the object detector and the relation predictor are confronted with significant difficulties, and the bond between them should be emphasized. SGCls is proposed for object and relation recognition with provided object locations. This metric takes into consideration of the object recognition capability of relation predictor. PredCls eliminates all effects derived from object detector, which only focuses on the performance of relation prediction with groundtruth objects.

We use Visual Genome-150 (VG) as the benchmark dataset [12], which is the largest dataset in Scene Graph Research. Although the VG dataset has been widely used in SGG research, there still exist many problems about this dataset, *e.g.*, messy annotations, fuzzy labeling of categories, overlapped bounding boxes and severe long-tailed distribution of both objects and relations. 108,077 images are notated in the VG dataset, including 1,366,673 object instances and 1,531,448 pairs of relations in 108,249 isolated scene graphs, along with about 21 object and 22 relation instances for each image.

The most widely accepted evaluation metrics are Recall@K (R@K) and Mean Recall@K (mR@K). R@K is introduced from recommendation system research, and aims to evaluate the Recall performance of the top-K predicted results. Considering that SGG models are usually affected by long-tailed dataset and biased training process, mR@K is proposed to enhance the sensitivity to tailed data. Conventionally, the setting of K is fixed to 20, 50 and 100.

4.2 Toy Experiments on Heterogeneity

To validate the effectiveness of heterogeneity, we first construct a de-homogenized method IMP-H from the homogeneous IMP. As shown in Table 2, IMP-H improves mR by 36.28% and R by 47.55% compared to the IMP method in average. The concept of IMP-H construction is shown as following explanation:

IMP is an implicit homogeneous SGG method based on RNN [29]. The core computing process of IMP can be summarized as follows (assuming that $\eta = 3$):

$$\begin{cases} \mathcal{F}'_{obj} = \omega_1^{GRU} \cdot \omega_2^{GRU} \cdot \omega_1^{GRU} \cdot \mathcal{F}_{obj}, \\ \mathcal{F}'_{rel} = \omega_2^{GRU} \cdot \omega_1^{GRU} \cdot \omega_2^{GRU} \cdot \mathcal{F}_{rel}, \end{cases} \tag{13}$$

where the complex parameters and computing process of GRU are simplified to ω^{GRU}, and ω_1^{GRU} indicates the parameters of Node GRU, while ω_2^{GRU} represents Edge GRU.

IMP-H Moreover, we propose a semi-heterogeneous IMP-H predictor on the basis of IMP method with minimum changes. This method de-homogenizes IMP by adopting merely one-character modification in code and achieve 30%+ improvement in SGDet task. Compared with Eq. (13), the IMP-H method simply arrange the message parsing function as follows:

$$\begin{cases} \mathcal{F}'_{context} = \omega_1^{GRU} \cdot \omega_2^{GRU} \cdot \omega_1^{GRU} \cdot \mathcal{F}_{obj}, \\ \mathcal{F}'_{rel} = \omega_2^{GRU} \cdot \omega_1^{GRU} \cdot \omega_2^{GRU} \cdot \mathcal{F}_{rel}, \\ \mathcal{F}'_{obj} = \mathcal{F}_{obj}. \end{cases} \tag{14}$$

Here, all the GRU units are trained in the same way as in Eq. (13), but the object embedding $\mathcal{F}'_{obj}$ is selected as the original $\mathcal{F}_{obj}$. $\mathcal{F}'_{context}$ only works as the context information to assist the relation prediction training progress.

Table 1: Types of some typical SGG baselines

Method	Degree of Heterogeneity	Implementation
IMP [29]	Homogeneous(Implicit)	RNN
IMP-H	Semi-heterogeneous	RNN
VTransE [37]	Homogeneous(Explicit)	Fully Connection
KERN [3]	Semi-heterogeneous	Knowledge Graph+GNN
MOTIFS [36]	Semi-heterogeneous	RNN+Prior
VCTree [24]	Semi-heterogeneous	Tree-Structured RNN
BGNN [13]	Semi-heterogeneous	GNN

4.3 Comparisons with State-of-the-Art Methods

We select several typical methods with distinct characteristics to validate the HLB approach. As shown in Table 1, we list the core designs of these baselines, and divide them into approaches with homogeneous feature spaces and semi-heterogeneous ones.

As shown in Figure 2, some SOTA methods are reported in the top few lines. In SGDet task, the extended BGNN+HLB method achieve significant better performance than other SOTA methods. Even though the BGNN+HLB model is not trained for PredCls and SGCls task, it still achieve compatible results on both tasks.

Meanwhile, seven methods are selected for heterogenization with HLB, including 4 semi-heterogeneous methods, 2 homogeneous methods and 1 de-homogenized approach. For all semi-heterogeneous methods, training with HLB shows comprehensive improvement on all the tasks and evaluation matrices, which demonstrates the effectiveness of the proposed method.

Moreover, we comprehensively analyze the performance of HLB on all the baselines. Considering that semi-heterogeneous and homogeneous methods can enhance the accuracy of relation prediction via the correlation between object and relation representations, these methods usually perform well with given ground-truth object information, *i.e.*, in PredCls task. Even though that object representations are already independent in PredCls, the HLB still slightly optimize the performance of part methods. However, object information is generally not accessible in real-world scenario. It indicates that object information is unreliable, especially in SGDet task. Thus, it is crucial to jointly learn more expressive and independent representations of both objects and relations. Table 2 shows the effectiveness of the HLB in both SGCls and SGDet task.

Finally, we analyze the influence of HLB on feature space. In order to reduce the computational complexity, we obtain the cluster-center c_i by averaging all feature vectors for each predicate category i. As shown in Figure 5(a), HLB increases the similarity in the same class, which means less intra-class variation. Considering that the inter-class ambiguity problem is not only related to the distance between cluster center, but also the degree of intra-class aggregation A^{intra}, we measure inter-class variation V^{inter} with Eq. (15):

$$\begin{aligned} A_i^{intra} &= G_{conf}(F_i), \\ V_{i,j}^{inter} &= 2 \times S_{i,j}^{center} / (A_i^{intra} + A_j^{intra}), \end{aligned} \tag{15}$$

Table 2: Comparison Results on Visual Genome Dataset

Model	PredCls				SGCls				SGDet			
	mR@50	R@50	mR@100	R@100	mR@50	R@50	mR@100	R@100	mR@50	R@50	mR@100	R@100
GBNet-β [34]	22.1	66.6	24.0	68.2	12.7	37.3	13.4	38.0	7.1	26.3	8.5	29.9
Graph R-CNN [32]	16.4	54.2	17.2	59.1	9.0	29.6	9.5	31.6	5.8	11.4	6.6	13.7
ReIDN [38]	15.8	68.7	17.2	68.8	9.3	38.9	9.6	38.9	6.0	31.0	7.3	36.7
FCSGG [17]	6.3	41.0	7.1	45.0	3.7	23.5	4.1	25.7	3.6	21.3	4.2	25.1
GPS-Net [31]	19.2	69.7	21.4	69.7	11.7	42.3	12.5	42.3	7.4	28.9	9.5	33.2
IMP [29]	9.8	59.3	10.5	61.3	5.8	34.6	6.0	35.4	3.8	20.7	4.8	24.5
IMP+HLB	**10.63**	**60.91**	**11.37**	**62.95**	**6.62**	**38.10**	**6.98**	**39.01**	**4.19**	**26.67**	**5.23**	**31.85**
IMP-H	10.17	58.89	10.97	61.31	6.05	34.89	6.47	36.59	5.37	31.21	6.30	35.36
IMP-H+HLB	**10.44**	**59.43**	**11.17**	**61.52**	**7.07**	**38.21**	**7.47**	**39.09**	**5.87**	**31.79**	**6.84**	**35.91**
VTransE [37]	14.7	65.7	15.8	67.6	8.2	38.6	8.7	39.4	5.0	29.7	6.0	34.3
VTransE+HLB	**15.26**	65.68	**16.40**	67.60	8.24	**39.72**	8.74	**40.61**	**5.14**	29.74	**6.22**	**34.47**
KERN [3]	**17.7**	**65.8**	**19.2**	**67.6**	**9.4**	36.7	**10.0**	37.4	6.4	27.1	7.3	29.8
KERN+HLB	15.89	61.17	17.15	64.17	9.01	**38.16**	9.69	**39.37**	**7.11**	**28.70**	**8.58**	**33.41**
MOTIFS [36]	14.0	**65.2**	15.3	**67.1**	7.7	35.8	8.2	36.5	5.7	27.2	6.6	30.3
MOTIFS+HLB	**15.39**	64.91	**16.74**	66.80	**8.90**	**39.48**	**9.44**	**40.32**	**7.19**	**32.57**	**8.43**	**37.01**
VCTree-SL [24]	17.0	**66.2**	18.5	**67.9**	9.8	**37.9**	10.5	**38.6**	6.7	27.7	7.7	31.1
VCTree-SL+HLB	**17.47**	65.73	**18.79**	67.35	**11.98**	36.95	**12.73**	38.50	**7.46**	**32.04**	**8.75**	**36.34**
BGNN [13]	**30.4**	59.2	**32.9**	61.3	14.3	**37.4**	16.5	**38.5**	10.7	**31.0**	12.6	**35.8**
BGNN+HLB	28.20	**61.06**	30.43	**63.22**	**16.72**	35.27	**18.09**	36.64	**12.57**	27.80	**15.03**	32.28
On Average	**+1.45%**	-0.21%	**+0.96%**	-0.02%	**+11.73%**	**+4.08%**	**+10.74%**	**+4.39%**	**+12.63%**	**+8.83%**	**+14.20%**	**+10.48%**
	+0.54%				**+7.73%**				**+11.53%**			

Table 3: Ablation study on HLB network structure design

	PredCls		SGCls		SGDet	
	mR @20 @50 @100	R @20 @50 @100	mR @20 @50 @100	R @20 @50 @100	mR @20 @50 @100	R @20 @50 @100
IMP-H	8.04 10.17 10.97	51.47 58.89 61.31	5.01 6.05 6.47	30.37 34.89 36.59	3.98 5.37 6.30	24.45 31.21 35.36
IMP-H-AD	7.78 9.67 10.43	51.67 58.95 61.38	4.83 5.80 6.21	30.49 35.00 36.68	3.90 5.25 6.14	24.49 31.32 35.49
IMP-H-LP	7.72 9.54 10.24	51.61 58.94 61.36	4.76 5.69 6.09	30.42 34.91 36.56	4.02 5.38 6.23	24.44 31.26 35.39
IMP-H-GE	7.76 9.82 10.66	50.74 58.28 60.90	4.83 5.85 6.27	29.85 34.20 35.80	3.87 5.27 6.23	23.17 30.03 34.36
IMP-H-HLB	**8.50** **10.44** **11.17**	**52.73** **59.43** **61.52**	**5.84** **7.07** **7.47**	**34.89** **38.21** **39.09**	**4.34** **5.87** **6.84**	**24.78** **31.79** **35.91**

Figure 5: Feature Representations analysis

where F_i represents all the feature vectors of predicate class i. G_{conf} is a confidence boundary function to determine the degree of intra-class aggregation. It reserves 95% credible feature vectors and selects the feature vector with the lowest similarity with the cluster-center as the aggregation boundary. $S_{i,j}^{center}$ is the similarity between cluster-centers of predicate i and j.

4.4 Ablation Study

In order to make the results simpler to observe and to mitigate the complexity, we choose IMP-H as the baseline for ablation study, which has relatively simpler network structure and more obvious metrics variation.

Network Design. As shown in Table 3, several significant modules are removed or modified to validate the network structure design of the HLB. We first remove the decoder in Auto Encoder module, namely IMP-H-AD, which means that less information is preserved in the progress of generating low-dimension relation logits. In general, low-frequency data that have less impact on the overall performance are more likely to be discarded, *i.e.*, the long-tail problem is more prominent. Compared with IMP-H, the IMP-H-AD performs 3.51% worse in mean mR and 0.27% better in R, which demonstrates the effectiveness of the decoder.

Further, we remove the Link Prediction Module and initialize the object graph as a complete graph, namely IMP-H-LP. Within IMP-H-LP messages will propagate through all other nodes without restriction, which exacerbates the problem of over-smooth in graph learning. The experimental results in Table 3 show that IMP-H-LP performs 8.38% worse in mean mR and 1.67% worse in mean R compared with IMP-H-HLB.

Compared with IMP-H-LP and IMP-H-HLB, another possible way to alleviate over-smooth is to replace the complete graph with the ground truths relation edges in training. We also study this approach with IMP-H-GE. It should be noted that as only ground facts are utilized for training, a small amount of the data is employed. Meanwhile, the IMP-H-GE can not deal with the period of test. As shown in Table 3, the performance of IMP-H-GE is 7.47% worse in mean mR and 4.03% worse in mean R in comparison with IMP-H-HLB, which shows slight advantage in mR against IMP-H-LP but even worse R metric.

Graph Formulation. To validate the effectiveness of the over-smooth-proof graph formulation of HLB module, we report the original and re-formulated evaluation results of SGDet in Table 4. Two modern graph learning networks without such over-smooth

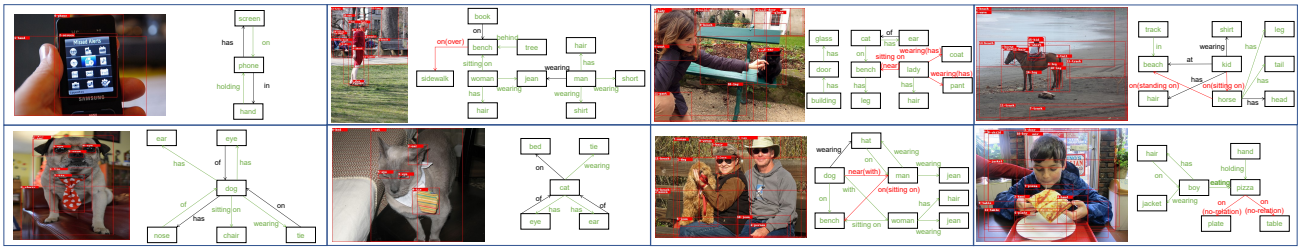

Figure 6: Qualitative results of our Scene Graph Generation method on VG dataset. The words marked with green denote correctly detected objects and relations, the red words and lines represent wrongly predicted ones with notated labels in brackets, and words marked with black color refer to considered positive but unlabeled predicated relations.

Table 4: Results of IMP-H with different graph formulation

	SGDet					
	mR@20	**mR@50**	**mR@100**	**R@20**	**R@50**	**R@100**
GCN	3.14	4.17	4.83	23.05	29.58	33.54
GCN+	**4.02**	**5.45**	**6.32**	**24.53**	**31.47**	**35.52**
GAT	3.96	5.35	6.21	**24.55**	**31.41**	**35.50**
GAT+	**4.06**	**5.50**	**6.40**	24.38	31.28	35.29
HLB-	3.96	5.27	6.09	24.36	31.08	35.13
HLB	**4.34**	**5.87**	**6.84**	**24.78**	**31.79**	**35.91**

sub-item design are considered for comparison:

$$GCN : \mathbf{x}_i' = \sigma(\omega \sum_{j \in \mathcal{N}(i)} \frac{e_{j,i}}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{x}_j), \tag{16}$$

$$GAT : \mathbf{x}_i' = \sigma(\sum_{j \in \mathcal{N}(i)} a_{i,j} \omega \mathbf{x}_j). \tag{17}$$

By attaching the over-smooth-proof term, we re-formulate their propagation functions as follows:

$$GCN+ : \mathbf{x}_i' = \sigma(\omega_1 x_i + \omega_2 \sum_{j \in \mathcal{N}(i)} \frac{e_{j,i}}{\sqrt{\hat{d}_j \hat{d}_i}} \mathbf{x}_j), \tag{18}$$

$$GAT+ : \mathbf{x}_i' = \sigma(\omega_1 x_i + \sum_{j \in \mathcal{N}(i)} a_{i,j} \omega_2 \mathbf{x}_j). \tag{19}$$

Meanwhile, we also remove the over-smooth-proof term in the HLB method for further validation:

$$HLB- : \mathbf{x}_i' = \sigma(\omega \cdot \text{mean}_{j \in \mathcal{N}(i)}(\mathbf{x}_j)). \tag{20}$$

The experimental results in Table 4 show that a simple self-enhancement item can alleviate the over-smooth problem in a shallow graph network.

Feature Transformation Network Setting. Considering that GNN is mainly affected by the number of network layers than CNN, we additionally explore the design of the number of GNN layers in the Feature Transformation Module from a theoretical perspective. Since the purpose of SGG is to comprehensively predict the relationships between objects in the overall scenario, we expect the information fusion in the Transformation Module to cover most of the object nodes in the graph. In this case, we analyze the graph structures in the VG-150 training set.

As shown in Table 5, the analysis is conducted at two levels:

Node-level: What is the number/proportion of nodes that have the nearest distances of L with all other nodes in a same graph.

Graph-level: What is the number/proportion of graphs that the maximum length of shortest path of all nodes is L.

Table 5: The shortest path length between object nodes in VG training set

	$L=1$	$L=2$	$L \geq 3$	**Avg-L**
Node-level	278249|74.3%	79467|21.2%	16777|4.5%	1.31
Graph-level	70795|65.5%	29519|27.3%	7759|7.2%	1.43

The length of shortest path in Table 5 is compiling with Dijkstra algorithm. We notice that 95.5% of object nodes can be associated with others through a distance of no more than 2, and 65.5% + 27.3% = 92.8% scene graphs are fully composed of such nodes. In other words, a GNN network that can fuse the information of neighbor nodes with a distance of 2 is sufficient in the SGG. The experimental results is shown in Table 6.

Table 6: Results of IMP-H with different number of Graph Network layers

	PredCls					
Layers	**mR@20**	**mR@50**	**mR@100**	**R@20**	**R@50**	**R@100**
$L=1$	8.04	9.69	10.31	51.32	58.58	60.85
$L=2$	**8.50**	**10.44**	**11.7**	**52.73**	**59.43**	**61.52**
$L=3$	6.95	8.50	9.09	50.53	57.59	59.82

5 CONCLUSION

In this paper, we explored the effect of heterogeneity in SGG task, and proposed a novel Heterogeneous Learning Branch. The HLB can be attached to many SGG methods without any additional inference cost. Inside the HLB, an over-smooth-proof formulated GNN and a hierarchical Link Prediction module are constructed to deal with long-tailed distributed data and dense relation proposals. We applied HLB in seven typical SGG methods and conducted comprehensive experiments to demonstrate the effectiveness of HLB on VG-150 dataset. The results evidently show that HLB can significantly improve the performance of the existing SGG methods.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation of China (62072232), Natural Science Foundation of Jiangsu Province (BK2019 1248) and Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] S Antol, A Agrawal, J Lu, M Mitchell, D Batra, C Lawrence Zitnick, and D Parikh. 2015. VQA: Visual Question Answering. In *ICCV*.

[2] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. 2019. Counterfactual critic multi-agent training for scene graph generation. In *ICCV*.

[3] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. 2019. Knowledge-Embedded Routing Network for Scene Graph Generation. In *CVPR*.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

[5] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*.

[7] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NuerIPS*.

[8] Kaiming He, Georgia Gkioxari, Piotr Doll¨¢r, and Ross Girshick. 2017. Mask R-CNN. In *ICCV*.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *TPAMI*.

[10] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. 2018. Mapping Images to Scene Graphs with Permutation-Invariant Structured Prediction. In *NuerIPS*.

[11] J. Johnson, R. Krishna, M. Stark, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. 2015. Image retrieval using scene graphs. In *CVPR*.

[12] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David Shamma, Michael Bernstein, and Fei-Fei Li. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*.

[13] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. 2021. Bipartite Graph Network with Adaptive Message Passing for Unbiased Scene Graph Generation. In *CVPR*.

[14] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. 2018. Factorizable Net: An Efficient Subgraph-Based Framework for Scene Graph Generation. In *ECCV*.

[15] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. 2017. Scene graph generation from objects, phrases and region captions. In *ICCV*.

[16] L Liao, Y Ma, X He, R Hong, and T Chua. 2018. Knowledge-aware multimodal dialogue systems. In *ACM MM*.

[17] Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. 2021. Fully Convolutional Scene Graph Generation. In *CVPR*.

[18] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. 2019. Attentive Relational Networks for Mapping Images to Scene Graphs. In *CVPR*.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *TPAMI*.

[20] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC*.

[21] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *AAAI*.

[22] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. 2021. Energy-Based Learning for Scene Graph Generation. In *CVPR*.

[23] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased Scene Graph Generation From Biased Training. In *CVPR*.

[24] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. 2019. Learning to Compose Dynamic Tree Structures for Visual Contexts. In *CVPR*.

[25] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based multi-relational graph convolutional networks. *ICLR* (2020).

[26] Hai Wan, Yonghao Luo, Bo Peng, and Wei-Shi Zheng. 2018. Representation Learning for Scene Graph Completion via Jointly Structural and Visual Embedding. In *IJCAI*.

[27] Weitao Wang, Ruyang Liu, Mingle Wang, Sen Wang, Xiaojun Chang, and Yang Chen. 2020. Memory-Based Network for Scene Graph with Unbalanced Relations. In *ACM MM*.

[28] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2019. Exploring Context and Visual Pattern of Relationship for Scene Graph Generation. In *CVPR*.

[29] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. 2017. Scene Graph Generation by Iterative Message Passing. In *CVPR*.

[30] Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. 2020. PCPL: Predicate-Correlation Perception Learning for Unbiased Scene Graph Generation. In *ACM MM*.

[31] Gengcong Yang, Jingyi Zhang, Yong Zhang, Baoyuan Wu, and Yujiu Yang. 2021. Probabilistic Modeling of Semantic Ambiguity for Scene Graph Generation. In *CVPR*.

[32] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. Graph R-CNN for Scene Graph Generation. In *ECCV*.

[33] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring Visual Relationship for Image Captioning. In *ECCV*.

[34] Alireza Zareian, Svebor Karaman, and Shih-Fu" Chang. 2020. Bridging Knowledge Graphs to Generate Scene Graphs. In *ECCV*.

[35] Alireza Zareian, Zhecan Wang, Haoxuan You, and Shih-Fu Chang. 2020. Learning Visual Commonsense for Robust Scene Graph Generation. In *ECCV*.

[36] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural Motifs: Scene Graph Parsing with Global Context. In *CVPR*.

[37] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. 2017. Visual Translation Embedding Network for Visual Relation Detection. In *CVPR*.

[38] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. 2019. Graphical Contrastive Losses for Scene Graph Parsing. In *CVPR*.

APPENDICES

A MORE EXPERIMENTAL RESULTS

	PredCls						SGCls						SGDet					
	mR			R			mR			R			mR			R		
Model	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**
IMP [29]	-	9.8	10.5	52.7	59.3	61.3	-	5.8	6.0	31.7	34.6	35.4	-	3.8	4.8	14.6	20.7	24.5
IMP+HLB	8.67	**10.63**	**11.37**	**54.13**	**60.91**	**62.95**	5.55	**6.62**	**6.98**	**34.74**	**38.10**	**39.01**	2.86	**4.19**	**5.23**	**18.97**	**26.67**	**31.85**
IMP-H	8.04	10.17	10.97	51.47	58.89	61.31	5.01	6.05	6.47	30.37	34.89	36.59	3.98	5.37	6.30	24.45	31.21	35.36
IMP-H+HLB	**8.50**	**10.44**	**11.17**	**52.73**	**59.43**	**61.52**	**5.84**	**7.07**	**7.47**	**34.89**	**38.21**	**39.09**	**4.34**	**5.87**	**6.84**	**24.78**	**31.79**	**35.91**
KERN [3]	-	**17.7**	**19.2**	-	**65.8**	**67.6**	-	**9.4**	**10.0**	-	36.7	37.4	-	6.4	7.3	-	27.1	29.8
KERN+HLB	12.33	15.89	17.15	51.91	61.17	64.17	7.28	9.01	9.69	34.40	**38.16**	**39.37**	4.75	**7.11**	**8.58**	21.15	**28.70**	**33.41**
VTransE [37]	11.6	14.7	15.8	**59.0**	65.7	67.6	6.7	8.2	8.7	35.4	38.6	39.4	3.7	5.0	6.0	23.0	29.7	34.3
VTransE+HLB	**12.03**	**15.26**	**16.40**	59.07	65.68	67.60	**6.75**	8.24	8.74	**36.41**	**39.72**	**40.61**	**3.82**	**5.14**	**6.22**	22.95	29.74	**34.47**
MOTIFS [36]	10.8	14.0	15.3	**58.5**	**65.2**	**67.1**	6.3	7.7	8.2	32.9	35.8	36.5	4.2	5.7	6.6	21.4	27.2	30.3
MOTIFS+HLB	**11.99**	**15.39**	**16.74**	58.20	64.91	66.80	**7.20**	**8.90**	**9.44**	**36.13**	**39.48**	**40.32**	**5.37**	**7.19**	**8.43**	**25.29**	**32.57**	**37.01**
MOTIFS+PCPL [30]	-	**35.2**	**37.8**	-	50.8	52.6	-	**18.6**	**19.6**	-	27.6	28.4	-	9.5	11.7	-	14.6	18.6
MOTIFS+PCPL+HLB	19.90	24.96	26.78	48.42	**55.37**	**57.43**	11.02	13.53	14.25	29.53	**33.52**	**34.57**	7.63	**10.21**	**12.06**	18.52	**24.54**	**28.33**
VCTree-SL [24]	13.4	17.0	18.5	**59.8**	**66.2**	**67.9**	8.0	9.8	10.5	**35.0**	**37.9**	**38.6**	5.0	6.7	7.7	21.7	27.7	31.1
VCTree-SL+HLB	**13.69**	**17.47**	**18.79**	59.58	65.73	67.35	**9.74**	**11.98**	**12.73**	32.61	36.95	38.50	**5.40**	**7.46**	**8.75**	**24.91**	**32.04**	**36.34**
BGNN [13]	-	**30.4**	**32.9**	-	59.2	61.3	-	14.3	16.5	-	**37.4**	**38.5**	-	10.7	12.6	-	**31.0**	**35.8**
BGNN+HLB	23.35	28.20	30.43	53.51	**61.06**	**63.22**	13.91	**16.72**	**18.09**	31.01	35.27	36.64	9.16	**12.57**	**15.03**	21.08	27.80	32.28

	PredCls						SGCls						SGDet					
	ng-R			zR			ng-R			zR			ng-R			zR		
Model	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**	**@20**	**@50**	**@100**
IMP [29]	-	75.2	83.6	-	-	-	-	43.4	47.2	-	-	-	-	22.0	27.4	-	-	-
IMP+HLB	61.70	**76.55**	**84.53**	11.74	17.42	20.06	39.64	**47.59**	**51.65**	2.16	3.60	4.59	19.26	**27.99**	**34.86**	0.18	0.47	1.02
IMP-H	58.60	73.84	82.68	9.75	14.89	17.61	32.65	40.71	45.55	2.00	3.20	3.94	25.95	35.02	41.41	0.62	1.39	2.17
IMP-H+HLB	**60.07**	**74.97**	**83.32**	**11.47**	**16.83**	**19.49**	**39.81**	**47.69**	**51.57**	**2.38**	**3.76**	**4.55**	**26.33**	**35.60**	**42.07**	**0.63**	**1.50**	**2.31**
KERN [3]	-	**81.9**	**88.9**	-	-	-	-	**45.9**	**49.0**	-	-	-	-	30.9	35.8	-	-	-
KERN+HLB	58.61	75.31	84.36	1.44	2.59	3.65	35.28	43.74	48.17	1.04	1.65	2.05	21.88	**31.10**	**38.29**	0.17	0.25	0.41
VTransE [37]	-	-	-	-	**11.3**	**14.7**	-	-	-	-	2.5	3.3	-	-	-	-	0.8	1.5
VTransE+HLB	67.40	82.08	89.24	5.73	10.73	14.27	41.59	49.43	53.19	1.26	2.46	3.29	24.17	32.66	39.54	0.21	0.84	**1.59**
MOTIFS [36]	-	**81.1**	**88.3**	-	**10.9**	**14.5**	-	44.5	47.7	-	**2.2**	**3.0**	-	30.5	35.8	-	0.1	0.2
MOTIFS+HLB	65.90	80.54	87.89	0.81	2.55	4.33	41.21	**49.07**	**52.81**	0.33	0.77	1.16	26.81	**36.44**	**43.23**	0.02	0.06	**0.26**
MOTIFS+PCPL [30]	-	72.1	81.5	-	-	-	-	39.9	44.5	-	-	-	-	15.2	20.6	-	-	-
MOTIFS+PCPL+HLB	59.42	**76.44**	**85.54**	1.24	2.61	3.66	36.17	**46.20**	**51.10**	0.35	0.70	0.96	20.21	**29.53**	**36.87**	0.02	0.07	0.11
VCTree-SL [24]	-	-	-	-	**10.8**	**14.3**	-	-	-	-	**1.9**	**2.6**	-	-	-	-	0.2	0.7
VCTree-SL+HLB	68.03	82.18	89.16	1.45	3.87	6.07	48.98	58.61	62.85	0.57	1.36	2.04	26.47	36.06	42.57	0.12	**0.44**	**0.81**
BGNN+HLB	62.14	78.58	86.99	2.21	4.00	5.40	35.28	43.99	48.54	1.17	1.90	2.41	22.73	32.05	39.06	0.14	0.33	0.66

Table 7: Comparison Results on Visual Genome Dataset