# Scene Graph Generation Strategy with Co-occurrence Knowledge and Learnable Term Frequency

**Hyeongjin Kim** [1] **Sangwon Kim** [2] **Dasom Ahn** [1] **Jong Taek Lee** [3] **Byoung Chul Ko** [1]

## Abstract

Scene graph generation (SGG) is an important task in image understanding because it represents the relationships between objects in an image as a graph structure, making it possible to understand the semantic relationships between objects intuitively. Previous SGG studies used a message-passing neural networks (MPNN) to update features, which can effectively reflect information about surrounding objects. However, these studies have failed to reflect the co-occurrence of objects during SGG generation. In addition, they only addressed the long-tail problem of the training dataset from the perspectives of sampling and learning methods. To address these two problems, we propose CooK, which reflects the Co-occurrence Knowledge between objects, and the learnable term frequency-inverse document frequency (TF-$l$-IDF) to solve the long-tail problem. We applied the proposed model to the SGG benchmark dataset, and the results showed a performance improvement of up to 3.8% compared with existing state-of-the-art models in SGGen subtask. The proposed method exhibits generalization ability from the results obtained, showing uniform performance improvement for all MPNN models.

## 1. Introduction

Scene graph generation (SGG) is a type of image understanding that infers and interprets the relationships between objects in an image and expresses them as a language graph.

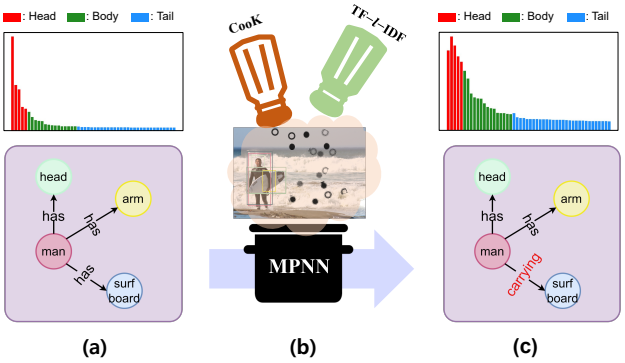


*Figure 1.* A novel learning recipe for SGG. (a) shows the class distribution and scene graph results of SGG performed using a conventional MPNN-based method. The proposed CooK + TF-$l$-IDF layer can be easily applied to existing MPNN-based models, as shown in (b). By updating the features according to the knowledge of object co-occurrence and the label inverse frequency, as shown in (c), it is possible to generate accurate relations between objects and successfully alleviate the long-tail problem.

SGG has been applied to various computer vision tasks, including image retrieval (Johnson et al., 2015; Schroeder & Tripathi, 2020), image captioning (Hossain et al., 2019; Zeng et al., 2022b), visual questions and answers (Ghosh et al., 2019; Guo et al., 2021a; Li et al., 2022c; Zeng et al., 2022a), and action recognition (Hu et al., 2022). The most common SGG approach is using an object detector (Ren et al., 2015; Redmon et al., 2016; Carion et al., 2020) to infer the relationships between detected objects in an image as a $<$ `subject`, `predicate`, `object`$>$ triplet. This triplet is then represented as a graph with a predicate edge between the subject and object nodes. For example, in the scene graph shown in Figure 1, the subject is 'man,' the predicate is 'carrying,' and the object is 'surfboard.' The relationship between 'man' and 'surfboard' is represented by the predicate edge 'carrying,' which indicates that the 'a man is carrying a surfboard'. However, there are many challenges with respect to accurately and effectively inferring the understood content. SGG has evolved to address the challenging problem of inferring relationships between objects. (Johnson et al., 2015) first proposed the use of scene graphs for

[1]Department of Computer Engineering, Keimyung University, Daegu, South Korea [2]Electronics and Telecommunications Research Institute (ETRI), Daegu 42994, South Korea [3]Department of Computer Engineering, Kyungpook National University, Daegu, South Korea. Correspondence to: Byoung Chul Ko <niceko@kmu.ac.kr>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

image retrieval and introduced a conditional random field (CRF) model for inferring relationships between objects. Recently, deep learning methods that utilize graph structures (Yang et al., 2018; Li et al., 2021; Yoon et al., 2023; Kim et al., 2023), such as graph neural networks (GNNs), have shown excellent performance in SGG. These methods use message-passing neural networks (MPNN) to accurately determine relationships between neighboring objects in a scene, enabling a more effective SGG inference.

However, existing SGG training datasets have a serious long-tail distribution, which can lead to the degradation of fine-grained object relationships and biased predictions toward dominant class labels. For example, a relationship such as 'walking on' and 'carrying' may be incorrectly predicted as a more dominant class label such as 'on' and 'has.' Several methods (Zheng et al., 2023; Sudhakaran et al., 2023) have been proposed to mitigate this issue. The methods mentioned above have all contributed to the improvement of performance for successful SGG, but they all have the following limitations: when updating relation features using an MPNN with a graph structure, there is a limitation in that prior knowledge about the mutual correlation between surrounding objects, such as that shown in Figure 1 (b) (e.g., person-surfboard, surfboard-wave), is not reflected in this process. State-of-the-art (SoTA) unbiased SGG studies (Li et al., 2021; Tang et al., 2020a; Zhou et al., 2022; Yu et al., 2021) are also constrained in their ability to solve this limitation because they focus only on de-biasing between class labels.

To overcome the limitations of existing SGG methods, we propose a novel SGG method that learns object relationships based on Co-occurrence Knowledge (CooK). The proposed SGG learning strategy reflects the co-occurrence information between objects in a scene by calculating the co-occurrence between objects from the training data and applying it to an MPNN. This provides the model with knowledge about the co-occurrence between objects, which has not been carefully considered in previous methods, enabling a more accurate inference of object relationships. In addition, we add a Learnable Term Frequency-Inverse Document Frequency (TF-$l$-IDF) layer to the CooK-based SGG model to alleviate the long-tail problem that exists in scene graph datasets. This layer updates features in a manner that emphasizes the features of the tail class and weakens those of the head classes. The contributions of the proposed SGG learning strategy are as follows:

- **Reflecting co-occurrence knowledge for accurate relationship inference**: SGG leverages the co-occurrence information between objects in a scene to improve the accuracy of relationship inference. This knowledge, which had been previously neglected by existing methods, helps the model to infer to more precise relationships between objects.
- **Mitigating the long-tail problem through a learnable TF-$l$-IDF layer**: SGG addresses the long-tail problem inherent in scene graph datasets by incorporating a learnable TF-$l$-IDF layer. This layer boosts the features of underrepresented classes (tail classes) while weakening the features of dominant classes (head classes), leading to more balanced and unbiased predictions.
- **Improving SGG performance**: Our research demonstrates that integrating the CooK module into existing SGG models results in significant performance improvements. In addition, experimental results show that it is possible to develop SGG models with better generalization performance by strengthening CooK knowledge with more data.

## 2. Related Work

### 2.1. SGG Approaches

Conventional SGG approaches typically use CNN (Lu et al., 2016) and RNN (Zellers et al., 2018; Chen et al., 2019c) to model object relationships and understand the visual context. These methods typically perform object detection and relation analysis in stages, and they use heuristic rules to generate scene graphs. However, this can lead to the overfitting of complex images or long-tail head classes.

Recently, there has been increased interest in SGG methods that exploit graph structures. Graph-based approaches can effectively reflect surrounding information by updating nodes based on the features of neighboring nodes. A representative graph-based study, Graph-R-CNN (Yang et al., 2018), proposed adaptive graph convolution networks that can efficiently update information between objects on top of existing graph convolutional networks. GPS-Net (Lin et al., 2020) proposed direction-aware message passing for node-specific contextual information. BGNN (Li et al., 2021) focused on unbiased SGG generation by proposing confidence-aware message propagation. HetSGG (Yoon et al., 2023) applied unbiased heterogeneous graph structures and updated object-predicate correlated features through the proposed relation-aware message passing, enabling more accurate SGG generation. EdgeSGG (Kim et al., 2023) addressed the limitations of existing graph-based SGG models by proposing an edge dual SGG architecture that inverts the roles of each node and edge of the graph. EdgeSGG enables the capture of both object- and edge-centric information, which is essential for generating fine-grained scene graphs. Prior studies on graph-based SGG focused on capturing more accurate relationships between objects and predicates. However, common-knowledge insights have been largely

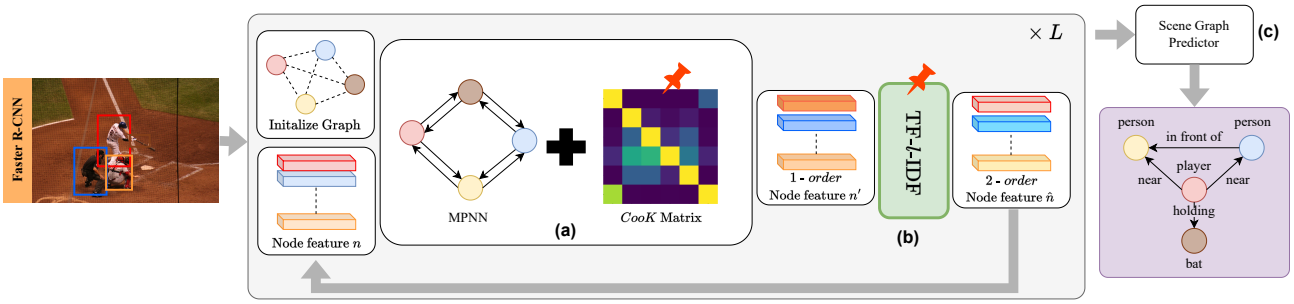


*Figure 2.* The whole training strategy of our proposed CooK + TF-$l$-IDF method. (a) In the MPNN process, we use the prior knowledge value $CooK(c_j|c_i)$ extracted from the training data to enable learning that reflects CooK. The 1-order node feature $n'$ generated in this way is used as an input to (b) TF-$l$-IDF, which can update features by considering the frequency between labels, to create a 2-order node feature $\hat{n}$. Finally, the 2-order node feature $\hat{n}$ that has undergone $L$ times of (a) and (b) processes is used to generate the final SG through the scene graph predictor in (c).

overlooked in SGG, limiting the ability of SGG models to reflect common sense regarding general correlations between objects.

### 2.2. Long-Tail Problem Solving

To improve SGG performance, data-centric approaches that consider the long-tail problem of the training dataset have also attracted attention. The most intuitive approach to addressing the long-tail problem is to design a loss function for long-tail mitigation without modifying the model (Knyazev et al., 2020). GPS-Net (Lin et al., 2020) addressed the long-tail problem in SGG by introducing a reweighted loss. BGNN (Li et al., 2021) used a new resampling strategy to construct scene graphs and improve the prediction performance of tail classes. HetSGG (Yoon et al., 2023) achieved unbiased scene graphs by changing the graph structure from homogeneous to heterogeneous. PE-Net (Zheng et al., 2023) proposed prototype embedding to make the unbiased feature vectors for each predicate more compact during the SGG process. VETO+MEET (Sudhakaran et al., 2023) is a mutually exclusive expert learning strategy that is employed for SGG to address long-tail problems. As noted previously, to address the long-tail problem, prior methods predominantly focused on structural modifications to the model. In this study, we introduce a data-centric approach to alleviate long-tail issues.

### 2.3. Label Correlation

Several approaches have been attempted in various fields to incorporate the correlation between labels into learning. For multi-label classification, SSGRL (Chen et al., 2019a) utilized a graph structure defined by co-occurrence information between labels to improve label classification performance. SALGL (Zhu et al., 2023b) achieved higher performance multi-label classification than previous studies by simultaneously utilizing co-occurrence knowledge and scene-aware knowledge between labels. In the SGG field, BA-SGG (Guo et al., 2021b) enabled more sophisticated scene graph generation by utilizing the co-occurrence information of "predicates" between objects. However, it has the limitation of not considering the co-occurrence knowledge between objects.

## 3. Cook + TF-$l$-IDF Recipe

### 3.1. Preliminaries

**Scene Graph Generation.** The goal of SGG is to successfully generate a graph $\mathcal{G} = (\mathcal{O}, \mathcal{R})$, where $\mathcal{O}$ is a set of objects found in a input image $I$ and $\mathcal{R}$ is a set of relations between them. To generate the graph, we first extract the object set $o_i \in \mathcal{O}$ from the input image $I$ using Faster R-CNN (Ren et al., 2015). Each object $o_i$ is represented as a tuple $(v_i, b_i, c_i)$, where $v_i \in \mathbb{R}^d$ is the visual feature map of $o_i$, $b_i \in [0,1]^4$ is the coordinates of the bounding box of $o_i$, and $c_i \in C$ is the class label of $o_i$. The relation between two objects $o_i$ and $o_j$ is represented as a triplet $r_i = < o_i, p_{i \to j}, o_j >$, where $p_{i \to j}$ is the relation feature map that represents the relation from $o_i$ to $o_j$. The feature map can be constructed by concatenating the features of $o_i$ and $o_j$ (Jung et al., 2023), or by updating the visual feature map of the union box of $o_i$ and $o_j$ (Li et al., 2021; Yoon et al., 2023; Kim et al., 2023). A successful SGG aims to learn the relation feature map $p_{i \to j}$ in a more discriminative manner.

**Bag of the Word (BoW).** BoW is a method for automatically classifying documents by looking at the distribution of words in a text. BoW considers a term to be relevant if it appears in a document, regardless of the word order or

structure of the document. However, BoW does not consider the frequency of words, so term frequency inverse document frequency (TF-IDF) was proposed to address this issue. TF-IDF can reflect the importance of each word by assigning different weights based on the frequency of each word. TF-IDF is calculated as follows:

$$t_{wd} = TF(n_{td}, n_d) \cdot IDF(N, n_t) \tag{1}$$

$$TF(n_{td}, n_d) = \frac{n_{td}}{n_d}, IDF(N, n_t) = \log(\frac{N}{n_t}) \tag{2}$$

where, $n_{td}, n_d, n_t$ and $N$ means occurrences of word $t$ in document $d$, number of word occurrence in document $d$, number of documents that contain word $t$ and number of documents, respectively.

### 3.2. Co-occurrence Knowledge

Existing SGG learning methods ignore the potential for co-occurrence between objects. Inspired by (Zhu et al., 2023a), we propose a CooK that can learn the co-occurrence of objects during the SGG learning stage. CooK is expressed as a matrix, as shown in Figure 2 (a); however, for convenience, we refer to it is CooK. For successful CooK-based SGG learning, CooK was extracted from the training dataset. In a training set $D_{train}$ having $K$ images, we count the number of objects with class $i$, $card(oc_i)$ in each image, and count the number of cases where different object class $i$ and $j$ coexist in the same image $card(oc_i \cap oc_j)$. The CooK probability of the object classes $oc_i$ and object class $oc_j$ occurring simultaneously in all images can be calculated as follows:

$$CooK(c_j|c_i) = \frac{\sum_{k=1}^{K} card_k(oc_i \cap oc_j)}{\sum_{k=1}^{K} card_k(oc_i)} \tag{3}$$

**Advanced CooK.** To obtain more refined knowledge, we adopt two different object recognition datasets, the Visual Genome (Xu et al., 2017b) and Open Images (Kuznetsova et al., 2020), which are used for SGG learning. This advanced CooK is able to store more extensive knowledge. With this advanced CooK, we can expect significant performance improvements for all SGG subtasks. A detailed discussion of the performance improvement is provided in Section 4.7.

### 3.3. Learnable TF-$l$-IDF Layer

Despite the successful generation of CooK, there remains a long-tail problem. To address this, previous studies have focused primarily on relation classes; however, object classes used for relation inference can also cause serious long-tail problems. Consequently, CooK can be biased in its configuration, which can severely hinder the generalization of overall model learning. To address this issue, we propose a novel method for updating node features by introducing a learnable TF-$l$-IDF layer inspired by TF-IDF scores and adding it to the output of MPNN, as shown in Figure 2 (b). The 1-order node features updated by the MPNN and CooK are input to TF-$l$-IDF, which updates them to new 2-order node features. The updated 2-order node features using the TF-$l$-IDF layer reduce the influence of the head class that can occur in the base backbone MPNN and increase the influence of rare body and tail classes.

**TF-$l$-IDF Layer.** Let the MPNN block be repeated $L$ times, the image batch size be $B$, and the object label set be $\mathcal{O}$. $Z_i^l = \{z_1^l, z_2^l, \cdots z_{\mathcal{O}}^l\} \in \mathbb{R}^{\mathcal{O} \times d_i}$ is the set of 1-order node feature of the $i$-th object and $d_i$ is the feature dimension. Let the set of $Z_i^l$ of a batch $B$, $X^l = \{Z_1^l, Z_2^l, \cdots Z_B^l\} \in \mathbb{R}^{B \times \mathcal{O} \times d_i}$. $X^l$ is fed to the TF-$l$-IDF layer, and the output is then the updated node feature set $Z_i^{(l+1)}$ :

$$Z_i^{l+1} = TF\text{-}l\text{-}IDF(X^l) \tag{4}$$

The TF-$l$-IDF layer can be expressed as the product of two terms as follows:

$$TF(X^l|n_{cb}, n_b) \cdot l\text{-}IDF(X^l|B, n_{z_i}; \epsilon, \gamma) \tag{5}$$

where $n_{cb}$ is the total number of occurrences of a specific class label $c$ observed in the $i$-th image of the batch, and $n_b$ denotes the total number of occurrences of all object labels in the $b$-th image of the batch. $n_{z_i}$ is the number of images in $z_i$ class. In Equation 6, the TF value represents the frequency of appearance of class $c$ in image $b$.

$$TF(n_{cb}, n_b) = \frac{n_{cb}}{n_b} \tag{6}$$

The $l$-IDF value is the inverse of the TF value.

$$l\text{-}IDF(B, n_c; \epsilon, \gamma) = \log(\frac{B + \epsilon}{n_c + \gamma}) \tag{7}$$

where $n_c$ is the number of images containing class label $c$. To address potential biases introduced during training owing to uneven sampling, we add trainable parameters $\epsilon$ and $\gamma$. These parameters allow for dynamic adjustments to the log term, thereby minimizing the impact of scenarios in which the body or tail labels are oversampled. The learnable TF-$l$-IDF layer aims to create a balanced feature representation that addresses the long-tail problem within object classes. By combining the TF with the $l$-IDF, the layer effectively updates the node features, ensuring a more nuanced and unbiased knowledge representation in CooK.

### 3.4. Training Strategy

In this section, we introduce a novel training strategy for SGG that exploits the capabilities of the CooK and TF-$l$-IDF

layers. The TF-$l$-IDF layer demonstrates seamless integration with any MPNN-based SGG task, improving model performance in capturing complex relationships within scenes. Figure 2 shows the overall pipeline that seamlessly combines CooK and TF-$l$-IDF components. To provide a more concrete illustration of this strategy, we examine the equation using Graph-R-CNN, which is a widely adopted and representative MPNN framework in SGG. The core of MPNN for SGG is determined by the node feature update using the following formula:

$$z^{l+1}_{o(u\to v)} = z^{l}_{o(u\to v)} + \sigma(z^{l}_{o(u)} + \sum_{v\in\mathcal{N}(u)} \alpha_{uv} W z^{l}_{o(v)}) \quad (8)$$

where $z^{l}_{o(u)}$ denotes the node features of node $u$ included in object $o$ after the $l$-th iteration, with $z^{0}_{o(u)}$ defined as the initial node feature of node $u$. $\alpha_{uv}$ represents the attention score between nodes $u$ and $v$.

$$\alpha_{uv} = \frac{\exp(W_{att}u)}{\exp(W_{att}u) + \exp(W_{att}v)} \quad (9)$$

where $W_{att}$ is the weight used to compute the attention score between $u$ and $v$. To leverage CooK in the MPNN process, Equation 8 is replaced with Equation 10.

$$z^{l+1}_{o(u\to v)} = z^{l}_{o(u\to v)} + \sigma(z^{l}_{o(u)} + \sum_{v\in\mathcal{N}(u)} cook_{u\to v}\alpha_{uv} W z^{l}_{o(v)}) \quad (10)$$

Here, $cook_{u\to v}$ refers to the CooK value that $v$ will occur when object $u$ occurs, and can be easily mapped to the $(u, v)$ values of the CooK calculated in advance. Through this process, CooK is successfully reflected during the MPNN process. For a more detailed explanation of the TF-$l$-IDF-based node feature updating process, refer to Algorithm 1.

### 3.5. Inference

Finally, we infer the scene graph $\mathcal{G} = (\mathcal{O}, \mathcal{R})$ for the input image $I$ using the successfully trained feature $Z$. The proposed CooK + TF-$l$-IDF learning method applies to all models in the MPNN format. Therefore, we describe the inference process using Graph-R-CNN as an example. The feature $z^{L}_{u\to v}$ that has passed through the final $L$ iterations is projected to the final relation class probability vector $p_{u\to v}$ through a simple linear classifier of weights $W_{rel}$ and softmax function:

$$p_{u\to v} = \text{softmax}(W_{rel} z^{L}_{u\to v}) \quad (11)$$

**Training Losses.** We use cross-entropy loss to train the MPNN model using the CooK + TF-$l$-IDF layer. The $\mathcal{L}_{obj}$ and $\mathcal{L}_{rel}$ losses for object classification and relation classification are jointly used for the final training as follows:

$$\mathcal{L}_{obj} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\mathcal{L}_{ce}(y_i, \tilde{u}_i),$$

$$\mathcal{L}_{rel} = \frac{1}{|\mathcal{R}|}\sum_{i=1}^{|\mathcal{R}|}\mathcal{L}_{ce}(s_{u\to v}, p_{u\to v})$$

$$\mathcal{L} = \mathcal{L}_{obj} + \mathcal{L}_{rel} \quad (12)$$

In Equation 12, $y_i$ and $s_{u\to v}$ represent the ground-truth (GT) for object and relation, respectively. $\tilde{u}$ represents the feature of the $u$ node that has passed through the linear classifier of weights $W_{obj}$ for object classification (e.g., $\tilde{u}_i = W_{obj}u^{L}_{i}$). Please refer to the PySGG (Tang, 2020) for detailed training environments of additional models, such as GPS-Net and BGNN.

**Algorithm 1** Processing of the TF-$l$-IDF layer

**Input:**
B: batch size
$\epsilon$ and $\gamma$: parameters for TF-$l$-IDF layer
$n_{cb}$: the total occurrences of a specific class label $c$ observed in the $b$-th image
$n_b$: the total number of occurrences of all object labels in the $b$-th image
$n_{z_i}$: the number of images including $z^{l}_{i}$ class
$Z^{l}_{i} = \{z^{l}_{1}, z^{l}_{2}\cdots z^{l}_{\mathcal{O}}\} \in \mathbb{R}^{\mathcal{O}\times d_i}$ : the set of 1-order node feature of $i$-th image of $l$ times
$X^{l} = \{Z^{l}_{1}, Z^{l}_{2}\cdots Z^{l}_{B}\} \in \mathcal{R}^{B\times\mathcal{O}\times d_i}$: the set of $Z^{l}_{i}$ of a batch B

// TF-$l$-IDF score value init.
$n_{cb}, n_b, n_{z_i} = 0$

// TF score set init.
**TF-score** = $\emptyset$

**for** $Z^{l}_{i} \in X^{l}$ **do**
  $n_b = |Z^{l}_{i}|$
  **for** $z^{l}_{i} \in Z^{l}_{i}$ **do**
    $n_{cb} = count(b, z^{l}_{i})$ // count $z^{l}_{i}$ label in image $b$
    $n_{z_i}$+=$\mathbb{I}_{\{z_i\in b\}}$ // if image $b$ contain label $z_i$
    **TF-score** = **TF-score** $\cup\, TF(n_{cb}, n_b)$
    $\cdot\, l\text{-}IDF(B, n_{z_i}; \epsilon, \gamma)$
  **end**
**end**
**2-order** $X^{l}$ = **TF-score** $\times X^{l}$ // elemental-wise multiplication

**Output: 2-order** $X^{l} \in \mathcal{R}^{B\times\mathcal{O}\times d_i}$

## 4. Experiment

### 4.1. Datasets

To verify the performance of the proposed CooK + TF-$l$-IDF method, experiments were conducted on the following two datasets: Visual Genome (Xu et al., 2017b) and Open

| Methods | PredCls | | SGCls | | SGGen | |
|---|---|---|---|---|---|---|
| | mR@ 50 / 100 | R@ 50 / 100 | mR@ 50 / 100 | R@ 50 / 100 | mR@ 50 / 100 | R@ 50 / 100 |
| IMP (Xu et al., 2017a) | 11.0 / 11.8 | 61.1 / 63.1 | 6.4 / 6.7 | 37.4 / 38.3 | 3.3 / 4.1 | 23.6 / 28.7 |
| KERN (Chen et al., 2019b) | 17.7 /19.2 | 65.8 / 67.6 | 9.4 /10.0 | 36.7 / 37.4 | 6.4 / 7.3 | 27.1 / 29.8 |
| Motifis (Zellers et al., 2018) | 14.6 / 15.8 | 66.0 / 67.9 | 8.0 / 8.5 | 39.1 / 39.9 | 5.5 / 6.8 | 32.1 / 36.9 |
| VCTree (Tang et al., 2019) | 15.4 / 16.6 | 65.5 / 67.4 | 7.4 / 7.9 | 38.9 / 39.8 | 6.6 / 7.7 | 31.8 / 36.1 |
| G-RCNN (Yang et al., 2018) | 16.4 / 17.2 | 65.4 / 67.2 | 9.0 / 9.5 | 37.0 / 38.5 | 5.8 / 6.6 | 29.7 / 32.8 |
| MSDN (Li et al., 2017) | 15.9 / 17.5 | 64.6 / 66.6 | 9.3 / 9.7 | 38.4 / 39.8 | 6.1 / 7.2 | 31.9 / 36.6 |
| Unbiased (Tang et al., 2020b) | 25.4 / 28.7 | 47.2 / 51.6 | 12.2 / 14.0 | 25.4 / 27.9 | 9.3 / 11.1 | 19.4 / 23.2 |
| GPS-Net (Lin et al., 2020) | 15.2 / 16.6 | 65.2 / 67.1 | 8.5 / 9.1 | 37.8 / 39.2 | 6.7 / 8.6 | 31.1 / 35.9 |
| R-CAGCN (Yang et al., 2021) | 18.3 / 19.9 | 66.6 / 68.3 | 10.2 / 11.1 | 38.3 / 39.0 | 7.9 / 8.8 | 28.1 / 31.3 |
| Nice-Motif (Li et al., 2022a) | 29.9 / 32.3 | 55.1 / 57.2 | 16.6 / 17.9 | 33.1 / 34.0 | 12.2 / 14.4 | 27.8 / 31.8 |
| PPDL (Li et al., 2022b) | 32.2 / 33.3 | 47.2 / 47.6 | 17.5 / 18.2 | 28.4 / 29.3 | 11.4 / 13.5 | 21.2 / 23.9 |
| RU-Net (Lin et al., 2022) | - / 24.2 | - / 46.9 | - / 14.6 | - / 29.0 | - / 10.8 | - / 24.2 |
| BGNN (Li et al., 2021) | 30.4 / 32.9 | 59.2 / 61.3 | 14.3 / 16.5 | 37.4 / 38.5 | 10.7 / 12.6 | 31.0 / 35.8 |
| IS-GGT (Kundu & Aakur, 2023) | 26.4 / 31.9 | - / - | 15.8 / 18.9 | - / - | 9.1 / 11.3 | - / - |
| HetSGG (Yoon et al., 2023) | 31.6 / 33.5 | 57.8 / 58.9 | 17.2 / 18.7 | 37.6 / 38.7 | 12.2 / 14.4 | 30.0 / 34.6 |
| HetSGG++ (Yoon et al., 2023) | 32.3 / 34.5 | 57.1 / 59.4 | 15.8 / 17.7 | 37.6 / 38.5 | 11.5 / 13.5 | 30.2 / 34.5 |
| PE-Net (Zheng et al., 2023) | 31.5 / 33.8 | 68.2 / 70.1 | 17.8 / 18.9 | 39.4 / 40.7 | 12.4 / 14.5 | 30.7 / 35.2 |
| SQUAT (Jung et al., 2023) | 30.9 / 33.4 | - / - | 17.5 / 18.8 | - / - | 14.1 / **16.5** | - / - |
| Transformer+CFA (Li et al., 2023) | 30.1 / 33.7 | 59.2 / 61.5 | 15.7 / 17.2 | 36.3 / 37.3 | 12.3 / 14.6 | 27.7 / 32.1 |
| VETO+Rwt (Sudhakaran et al., 2023) | 33.1 / 35.1 | 61.9 / 63.9 | 16.1 / 17.1 | 35.1 / 36.3 | 10.0 / 11.7 | 26.2 / 30.4 |
| CooK (ours) | 33.7 / 35.8 | 62.1 / 64.2 | 17.5/ 18.6 | 39.1 / 40.0 | 12.6 / 14.9 | 30.1 / 34.6 |
| TF-*l*-IDF (ours) | 33.6 / 35.8 | 61.7 / 63.4 | 18.5 / 19.4 | 38.4 / 39.8 | 12.8 / 15.0 | 29.3 / 32.6 |
| CooK + TF-*l*-IDF (ours) | **35.4 / 37.2** | 60.4 / 62.3 | **19.1 / 20.3** | 36.4 / 37.6 | **14.2** / 16.3 | 27.7 / 32.7 |

*Table 1.* Performance comparison with the SoTA SGG methods on the VG dataset.

Images (Kuznetsova et al., 2020).

**Visual Genome (VG).** The VG dataset consists of 108k images, with 150 object class labels and 50 relation labels. The dataset was divided into 70% training data and 30% test data, and preprocessing was performed according to a previous study (Xu et al., 2017a).

**Open Images (OI).** The OI dataset consists of 133k images, with 301 object class labels and 31 relation labels. A total of 126,368 images were used for training; 1,813 and 5,322 images were used for validation and testing, respectively.

### 4.2. Evaluation Metrices

**Visual Genome (VG).** To evaluate the performance of SGG on the VG dataset, we report the performance of three subtasks: Predicate Classification (PredCls), Scene Graph Classification (SGCls), and Scene Graph Generation (SGGen), which have been used in previous studies (Lyu et al., 2022; Tang et al., 2020a; 2019). PredCls are given labels and bounding box information for the object. SGCls is given only the bounding box information for the object, and SGGen uses only the values detected by the object detector without the GT label for the object. Following previous studies, we used recall@K (R@K) and mean recall@K (mR@K) as the primary evaluation metrics.

**Open Images (OI).** Following previous studies, we report the performance of the Recall@50 (R@50), weighted mean AP of relation $\mathbf{wmAP}_{rel}$, and weighted mean AP of phrase $\mathbf{wmAP}_{phr}$ metrics (Li et al., 2021), which are the main metrics used to measure performance on the OI dataset. In addition, we reported the final score $\mathbf{score}_{wtd}$, which was calculated as the weighted sum of the following three indicators:

$$\mathbf{score}_{wtd} = 0.2 \times \mathrm{R@50} + 0.4 \times \mathbf{wmAP}_{rel} + 0.4 \times \mathbf{wmAP}_{phr} \quad (13)$$

### 4.3. Implementation Details

All of the experiments were conducted on a private machine equipped with two Intel(R) Xeon(R) CPUs, that is, a Gold 6230R CPU @ 2.10 GHz; 128 GB RAM, and an NVIDIA RTX 3090 GPU. To detect objects in the image, we adopted the Faster R-CNN (Ren et al., 2015) with ResNeXt-101-FPN (Xie et al., 2017). GloVe (Pennington et al., 2014) was used to word embedding method.

### 4.4. Increasing Performance Using CooK + TF-*l*-IDF

**Visual Genome.** Table 1 compares the performances of the proposed CooK + TF-*l*-IDF and SoTA models. As shown in Table 1, our proposed method significantly improves the performance of most evaluation metrics. For CooK, which includes the co-occurrence knowledge between objects, PredCls showed a performance improvement of 2.1% / 2.3% on mR@50 / 100 because the object GT

| Methods | mR@50 | R@50 | $\text{wmAP}_{rel}$ | $\text{wmAP}_{phr}$ | $\text{score}_{wtd}$ |
|---|---|---|---|---|---|
| RelDN (Zhang et al., 2019) | 37.2 | 75.3 | 32.2 | 33.4 | 42.0 |
| VCTree (Tang et al., 2019) | 33.9 | 74.1 | 34.2 | 33.1 | 40.2 |
| G-RCNN (Yang et al., 2018) | 34.0 | 74.5 | 33.2 | 34.2 | 41.8 |
| Motifs (Zellers et al., 2018) | 32.7 | 71.6 | 29.9 | 31.6 | 38.9 |
| Unbiased (Tang et al., 2020b) | 35.5 | 69.3 | 30.7 | 32.8 | 39.3 |
| GPS-Net (Lin et al., 2020) | 38.9 | 74.7 | 32.8 | 33.9 | 41.6 |
| BGNN (Li et al., 2021) | 40.5 | 75.0 | 33.5 | 34.1 | 42.1 |
| RU-Net (Lin et al., 2022) | - | 76.9 | 35.4 | 34.9 | 43.5 |
| HetSGG (Yoon et al., 2023) | 42.7 | 76.8 | 34.6 | 35.5 | 43.3 |
| HetSGG++ (Yoon et al., 2023) | 43.2 | 74.8 | 33.5 | 34.5 | 42.2 |
| PE-Net (Zheng et al., 2023) | - | 76.5 | **36.6** | 37.4 | 44.9 |
| CooK (ours) | 42.9 | 75.5 | 34.6 | 36.4 | 43.5 |
| TF-$l$-IDF (ours) | 43.3 | 76.5 | 35.4 | 36.8 | 44.2 |
| CooK + TF-$l$-IDF (ours) | **43.8** | **77.0** | **36.6** | **37.6** | **45.1** |

*Table 2.* Performance comparison with the SoTA methods on OI dataset.

| Method | SGGen | | | |
|---|---|---|---|---|
| | mR@50 | mR@100 | R@50 | R@100 |
| G-RCNN (Yang et al., 2018) | 5.8 | 6.6 | 29.7 | 32.8 |
| G-RCNN + ours | 7.1 | 8.7 | 30.1 | 33.2 |
| GPS-Net (Lin et al., 2020) | 6.7 | 8.6 | 31.1 | 35.9 |
| GPS-Net + ours | 8.3 | 10.6 | 33.5 | 37.4 |
| BGNN (Li et al., 2021) | 10.7 | 12.6 | 31 | 35.8 |
| BGNN+ ours | 11.4 | 14.2 | 29.8 | 34.6 |
| **Mean improv.(%)** | **17.6**↑ | **22.6**↑ | **1.6**↑ | **0.7**↑ |

*Table 3.* Performance changes when the proposed CooK+TF-$l$-IDF are applied to various MPNN based models in the VG dataset.

| Learnable | PredCls | | |
|---|---|---|---|
| | mR@20 | mR@50 | mR@100 |
| $w/o$ | 26.8 | 31.6 | 33.4 |
| $w$ | **29.9** | **33.6** | **35.8** |
| Learnable | SGCls | | |
| | mR@20 | mR@50 | mR@100 |
| $w/o$ | 12.2 | 15.1 | 16.3 |
| $w$ | **16.1** | **18.5** | **19.4** |
| Learnable | SGGen | | |
| | mR@20 | mR@50 | mR@100 |
| $w/o$ | 8.3 | 11.1 | 13.2 |
| $w$ | **9.7** | **12.8** | **15.0** |

*Table 4.* Performance changes depending on the with ($w$) or without ($w/o$) learnable parameters in the TF-$l$-IDF layer.

was reflected in CooK. In addition, SGCls and SGGen each improved the performance by 0.3% / 0.1% and 0.4% / 0.5% on mR@50 / 100, respectively, but the performance improvements were smaller than those of PredCls. This is because neither method reflects CooK information and uses object labels directly predicted by the model. When only the TF-$l$-IDF layer was used, similar levels of performance improvement were observed for all three subtasks. In particular, as evidence of the performance improvement for the tail classes, which is the role of TF-$l$-IDF, the performance improvement of R@K was larger than that of mR@K. Finally, the largest performance improvement was observed when using information for both CooK and TF-$l$-IDF. This is the result of considering the knowledge of both object co-occurrence and class balance, which shows that more accurate SGG is possible through these considerations.

**Open Images.** To verify the generalized performance improvement of the proposed method, we conducted experiments on the SGGen task using the OI dataset, as listed in Table 2. Table 2 presents the results of the performance evaluation of the OI dataset. As with the VG dataset, our CooK+TF-$l$-IDF method showed a performance improvement of 0.2% over the SoTA models. In particular, unlike PE-NET, which requires training with prototypes, our proposed method can achieve a higher performance improvement using CooK and TF-$l$-IDF extracted from the training data.

**CooK + TF-$l$-IDF with MPNN-based models.** To verify the generalized performance improvement of our proposed method, we examined the changes in the SGGen performance when CooK and TF-$l$-IDF layers were applied to representative MPNN-based SGG methods (Yang et al., 2018; Lin et al., 2020; Li et al., 2021). As shown in Table 3, we can confirm that there was a performance improvement in all MPNN-based models. This shows that the proposed method can be broadly applied to MPNN-based SGG tasks and can achieve high generalization performance.

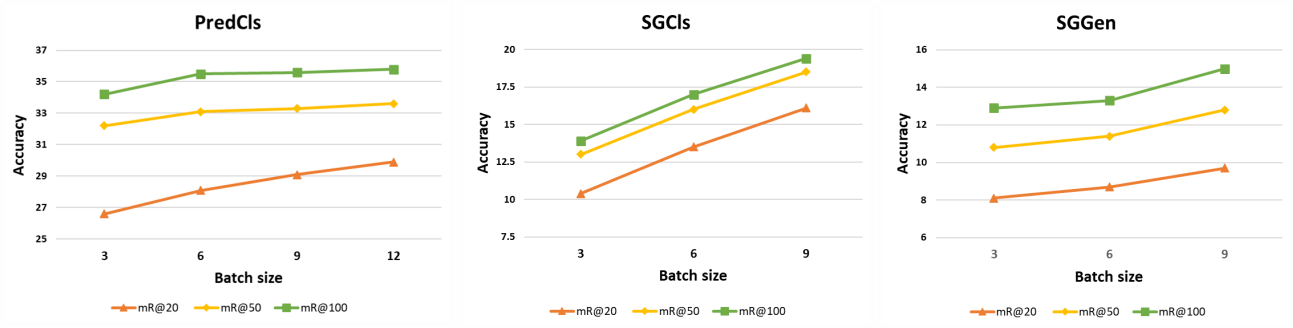


*Figure 3.* Difference in TF-$l$-IDF performance according to the batch size. As the proposed TF-$l$-IDF is performed in batches, it can be confirmed that the performance increases proportionally as the batch size increases.

| CooK-Type | SGGen mR@20 | SGGen mR@50 | SGGen mR@100 |
|---|---|---|---|
| One CooK for VG | 11.1 | 14.2 | 16.3 |
| Advanced | **11.4** | **14.9** | **17.1** |

*Table 5.* Performance changes when using Advanced CooK, a prior knowledge collected from a wider variety of environments. Advanced CooK achieved a higher performance than when it was not used. Because the OI dataset has more object labels than the VG dataset, $Map : OI \rightarrow VG$ was excluded as many cases occurred where mapping was not possible.

### 4.5. Abaltion Studies

**Use of Learnable TF-$l$-IDF.** The most significant contribution of the proposed TF-$l$-IDF is the learnable design of the IDF(,) function, which calculates the inverse document frequency. To compare the extent of the performance improvement, we compared the performance of the structure with the learnable design of TF-$l$-IDF and the performance of the structure without it. As can be seen in Table 4, the use of learnable parameters led to an average 2.4% performance improvement when compared to the case where they were not used. This is because learnable parameters can mitigate the cases in which a specific label is oversampled during training.

**Difference in TF-$l$-IDF Performance According to Batch Size.** As discussed in the previous section, TF-$l$-IDF is highly dependent on the batch size. Figure 3 depicts the performance of the TF-$l$-IDF layer on the three subtasks of PredCls, SGCls, and SGGen on the VG dataset according to the batch size. As shown in the figure, the performance gradually increased for all subtasks as the batch size increased. Therefore, it is necessary to increase the batch size to perform more sophisticated feature updates.

### 4.6. Long-tail Alleviation

In this experiment, we analyzed the effects of the proposed learning method on long-tail problem mitigation. Figure 4 shows the change in mR@100 for each class when the proposed method was applied. As illustrated in the figure, the mR@100 value for the head decreased, whereas those for the body and tail parts increased significantly. This demonstrates that CooK's 'knowledge of object co-occurrence' and TF-$l$-IDF's 'feature update' were successfully applied to each class part.

### 4.7. Advanced CooK

Human knowledge has become more generalized and reliable owing to extensive experience and activities. To verify the applicability of this human knowledge paradigm to CooK, we generated an advanced CooK based on additional datasets and applied it to the model. In this experiment, we combined two CooKs from the VG and OI datasets to create an advanced CooK. The mapping function $Map : OI \rightarrow VG$ for the combination was hand-crafted. Table 5 shows the performance results when using advanced CooK. Similar to the general improvement effect of knowledge, the advanced CooK achieved a higher performance than individual CooK. This demonstrates that CooK can improve the performance if the task uses knowledge obtained from similar datasets.

## 5. Limitation and Future work

The proposed paper successfully improves the performance of MPNN-based SGG models. However, exploring the application of CooK and TF-IDF to non-MPNN models could present a novel approach for SGG. As future work, we intend to investigate the feasibility and effectiveness of integrating the proposed CooK and TF-IDF methods into other approaches beyond MPNN-based models.

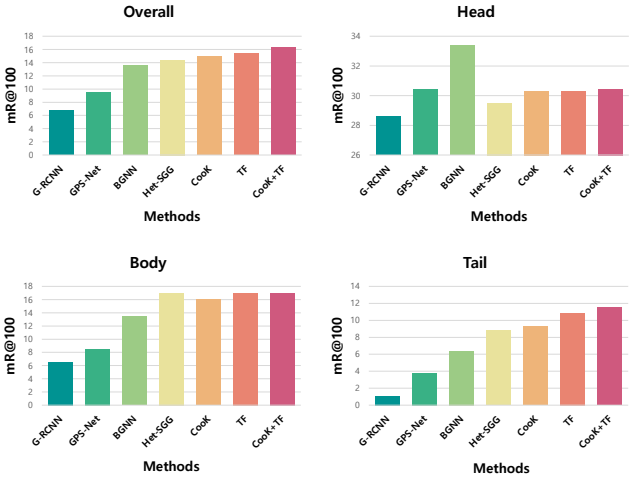


*Figure 4.* TF-$l$-IDF effect on long-tail problem. The proposed TF-$l$-IDF successfully reduces the mR@100 for common labels in head and focuses more on rare labels in body and tail.

## Conclusion

In this study, we proposed a CooK and TF-$l$-IDF layer that can solve co-occurrence knowledge and long-tail problems. Our proposed method has a significant advantage in that it can improve the performance of SGG tasks because it can be easily applied without significantly changing the existing model. In addition, by performing experiments using advanced CooK, we verify that this study realized a new approach for generating more general knowledge matrices. However, limitation is that it can only be applied to existing MPNN-based models. In addition, CooK learning is currently applicable only to supervised learning; therefore, it is difficult to apply it to foundation models conducted with self-supervision. In future research, we plan to study SGG, which enables CooK to learn the relationships between objects on its own and is applicable regardless of the SGG model.

## Impact Statement

This research presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## Acknowledgments

This work was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0020536, HRD Program for Industrial Innovation) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2022R1I1A3058128).

## References

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213–229. Springer, 2020.

Chen, T., Xu, M., Hui, X., Wu, H., and Lin, L. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 522–531, 2019a.

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

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

Ghosh, S., Burachas, G., Ray, A., and Ziskind, A. Generating natural language explanations for visual question answering using scene graphs and visual attention. *arXiv preprint arXiv:1902.05715*, pp. 1–7, 2019.

Guo, W., Zhang, Y., Yang, J., and Yuan, X. Re-attention for visual question answering. *IEEE Transactions on Image Processing (TIP)*, 30:6730–6743, 2021a.

Guo, Y., Gao, L., Wang, X., Hu, Y., Xu, X., Lu, X., Shen, H. T., and Song, J. From general to specific: Informative scene graph generation via balance adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16383–16392, 2021b.

Hossain, M. Z., Sohel, F., Shiratuddin, M. F., and Laga, H. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

Hu, Y., Gao, J., and Xu, C. Learning scene-aware spatio-temporal gnns for few-shot early action prediction. *IEEE Transactions on Multimedia (TMM)*, pp. 2061–2073, 2022.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M., and Fei-Fei, L. Image retrieval using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3668–3678, 2015.

Jung, D., Kim, S., Kim, W. H., and Cho, M. Devil's on the edges: Selective quad attention for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18664–18674, 2023.

Kim, H., Kim, S., Lee, J. T., and Ko, B. C. Semantic scene graph generation based on an edge dual scene graph and message passing neural network. *arXiv preprint arXiv:2311.01192*, pp. 1–9, 2023.

Knyazev, B., de Vries, H., Cangea, C., Taylor, G. W., Courville, A., and Belilovsky, E. Graph density-aware losses for novel compositions in scene graph generation. In *Proceeding of the British Machine Vision Conference (BMVC)*, pp. 1–14, 2020.

Kundu, S. and Aakur, S. N. Is-ggt: Iterative scene graph generation with generative transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6292–6301, 2023.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision (IJCV)*, 128(7):1956–1981, 2020.

Li, L., Chen, L., Huang, Y., Zhang, Z., Zhang, S., and Xiao, J. The devil is in the labels: Noisy label correction for robust scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18869–18878, 2022a.

Li, L., Chen, G., Xiao, J., Yang, Y., Wang, C., and Chen, L. Compositional feature augmentation for unbiased scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21685–21695, 2023.

Li, R., Zhang, S., Wan, B., and He, X. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11109–11119, 2021.

Li, W., Zhang, H., Bai, Q., Zhao, G., Jiang, N., and Yuan, X. Ppdl: Predicate probability distribution based loss for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19447–19456, 2022b.

Li, W., Zhang, H., Bai, Q., Zhao, G., Jiang, N., and Yuan, X. Ppdl: Predicate probability distribution based loss for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19447–19456, 2022c.

Li, Y., Ouyang, W., Zhou, B., Wang, K., and Wang, X. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1261–1270, 2017.

Lin, X., Ding, C., Zeng, J., and Tao, D. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3746–3753, 2020.

Lin, X., Ding, C., Zhang, J., Zhan, Y., and Tao, D. Ru-net: Regularized unrolling network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19466, 2022.

Lu, C., Krishna, R., Bernstein, M., and Fei-Fei, L. Visual relationship detection with language priors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 852–869, 2016.

Lyu, X., Gao, L., Guo, Y., Zhao, Z., Huang, H., Shen, H. T., and Song, J. Fine-grained predicates learning for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19467–19475, 2022.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pp. 1–9, 2015.

Schroeder, B. and Tripathi, S. Structured query-based image retrieval using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 178–179, 2020.

Sudhakaran, G., Dhami, D. S., Kersting, K., and Roth, S. Vision relation transformer for unbiased scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21882–21893, 2023.

Tang, K. A scene graph generation codebase in pytorch, 2020. https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch.

Tang, K., Zhang, H., Wu, B., Luo, W., and Liu, W. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6619–6628, 2019.

Tang, K., Niu, Y., Huang, J., Shi, J., and Zhang, H. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3716–3725, 2020a.

Tang, K., Niu, Y., Huang, J., Shi, J., and Zhang, H. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3716–3725, 2020b.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1492–1500, 2017.

Xu, D., Zhu, Y., Choy, C. B., and Fei-Fei, L. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5410–5419, 2017a.

Xu, D., Zhu, Y., Choy, C. B., and Fei-Fei, L. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5410–5419, 2017b.

Yang, G., Zhang, J., Zhang, Y., Wu, B., and Yang, Y. Probabilistic modeling of semantic ambiguity for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12527–12536, 2021.

Yang, J., Lu, J., Lee, S., Batra, D., and Parikh, D. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 670–685, 2018.

Yoon, K., Kim, K., Moon, J., and Park, C. Unbiased heterogeneous scene graph generation with relation-aware message passing neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3285–3294, 2023.

Yu, J., Chai, Y., Wang, Y., Hu, Y., and Wu, Q. Cogtree: Cognition tree loss for unbiased scene graph generation. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pp. 1274–1280, 2021.

Zellers, R., Yatskar, M., Thomson, S., and Choi, Y. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5831–5840, 2018.

Zeng, P., Zhang, H., Gao, L., Song, J., and Shen, H. T. Video question answering with prior knowledge and object-sensitive learning. *IEEE Transactions on Image Processing (TIP)*, 31:5936–5948, 2022a.

Zeng, P., Zhang, H., Song, J., and Gao, L. S2 transformer for image captioning. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, pp. 1608–1614, 2022b.

Zhang, J., Shih, K. J., Elgammal, A., Tao, A., and Catanzaro, B. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11535–11543, 2019.

Zheng, C., Lyu, X., Gao, L., Dai, B., and Song, J. Prototype-based embedding network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22783–22792, 2023.

Zhou, L., Hu, J., Zhou, Y., Lam, T. L., and Xu, Y. Peer learning for unbiased scene graph generation. *arXiv preprint arXiv:2301.00146*, pp. 1–7, 2022.

Zhu, X., Liu, J., Liu, W., Ge, J., Liu, B., and Cao, J. Scene-aware label graph learning for multi-label image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1473–1482, 2023a.

Zhu, X., Liu, J., Liu, W., Ge, J., Liu, B., and Cao, J. Scene-aware label graph learning for multi-label image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1473–1482, 2023b.

# Scene Graph Generation Strategy with Co-occurrence Knowledge and Learnable Term Frequency

## Appendix

Paper 435

## A. Training Configuration

| Hyperparmameters | Datasets | | |
|---|---|---|---|
| | VG | | OI |
| | PredCls | SGCls, SGGen | SGGen |
| LR | 0.008 | 0.008 | 0.008 |
| LR decay | WarmupMultiStepLR | WarmupMultiStepLR | WarmupMultiStepLR |
| Weight decay | $5 \times 10^{-5}$ | $5 \times 10^{-5}$ | $5 \times 10^{-5}$ |
| Iteration | 49,500 | 49,500 | 40,000 |
| Batch size | 12 | 9 | 9 |
| Num layers | 4 | 4 | 4 |
| Object dim | 128 | 128 | 128 |
| Relation dim | 128 | 128 | 128 |

*Table 6.* Model configurations for the benchmark datasets used in the experiments.

In order to ensure the reproducibility of the proposed model and the reliability of the experiments, we provide detailed hyperparameter values employed in the experiment. Table 6 shows the model configurations on each benchmark dataset.

## B. CooK Visualization

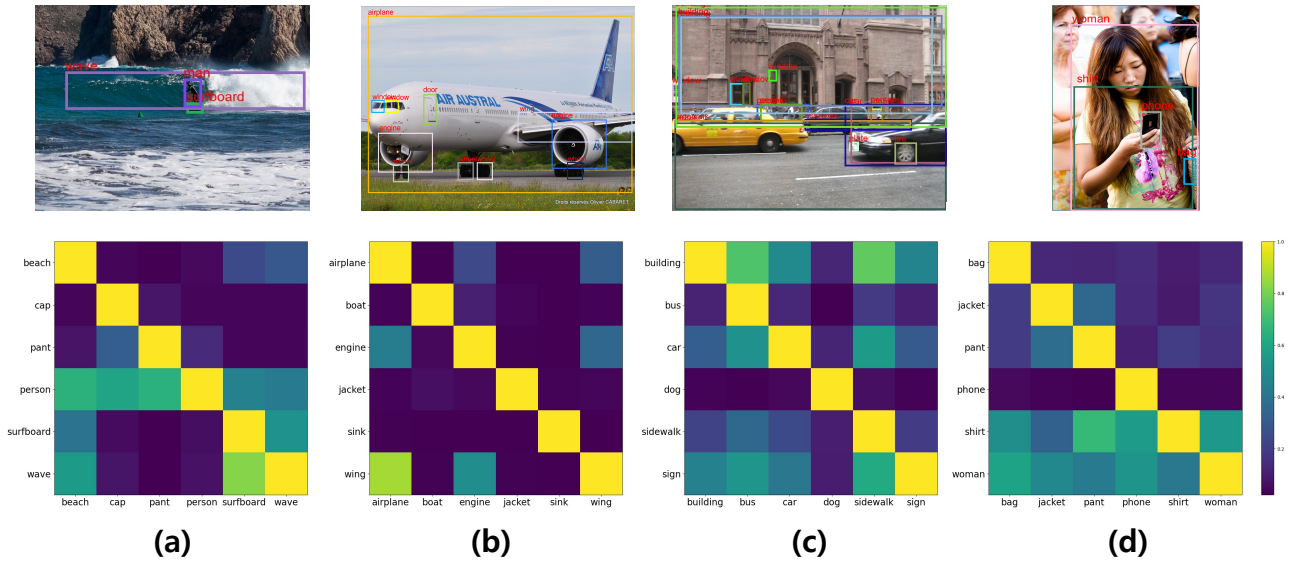


*Figure 5.* Visualization of CooK matrix. CooK above is a visualization of CooK that reflects object co-occurrence in VG dataset.

We visualize CooK to make it easier to explain how well the proposed CooK represents the correlation between objects. As you can see in the figure, it clearly reflects the relationship between objects that are closely related in the real-world of CooK. When looking at the relationship between the 'surfboard' and 'wave' in the case of (a), if there is a 'surfboard', it can be seen that it occurs together with high probability of 'wave' and 'beach'. Also, 'beach' and 'surfboard' show lower probability of co-occurrence than 'surfboard' and 'beach'. This is, of course, an accurate reflection of the fact that the 'surfboard' does not necessarily exist in the 'beach'. In other (b), (c), and (d), it can be seen that CooK reflects real-world co-occurrence knowledge well.

## C. Long-Tail Mitigation

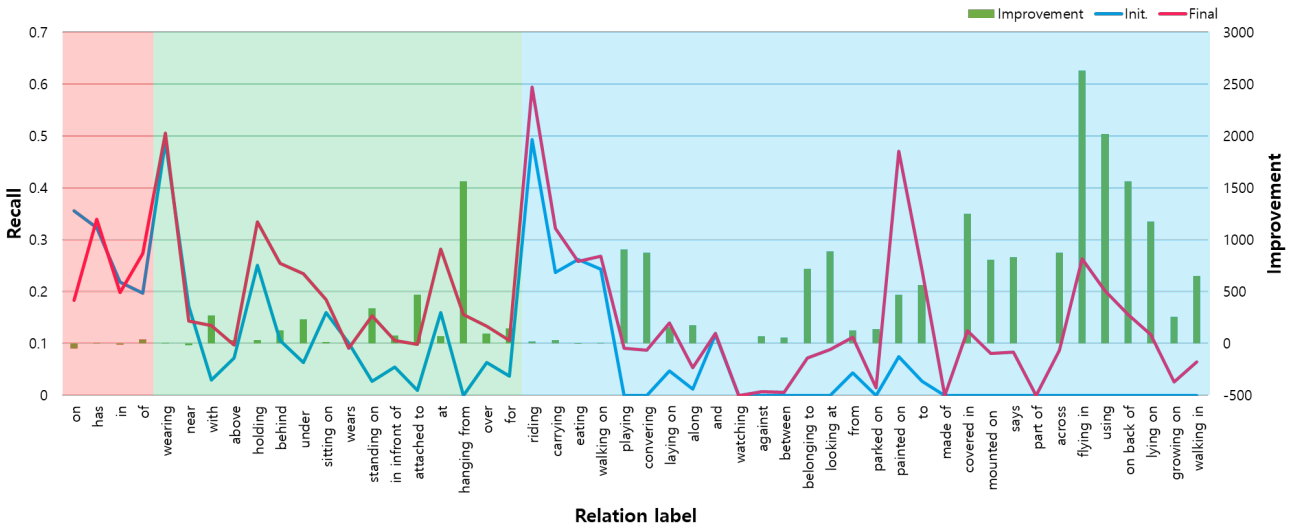


*Figure 6.* Mitigation of long-tail problems after training.

To add clarity to the long-tail problem resolution of the proposed method, we additionally report the recall (R@100) value and degree of improvement of the relocation labels in the SGGen task as learning progresses. The red area in the figure represents head classes, the green area represents body classes, and the blue area represents tail classes. The line graph represents the recall change of the relocation label according to model learning (right axis). As you can see in the figure, Figure 6 the change in tail is more dynamic than the change in head or body. This is more evident when you check the bar graph indicating the degree of improvement (left axis). It can be seen that the proposed TF-$l$-IDF decreases the frequency of the common class and increases the frequency of the rare class dramatically.

## D. Scene Graph Visualization

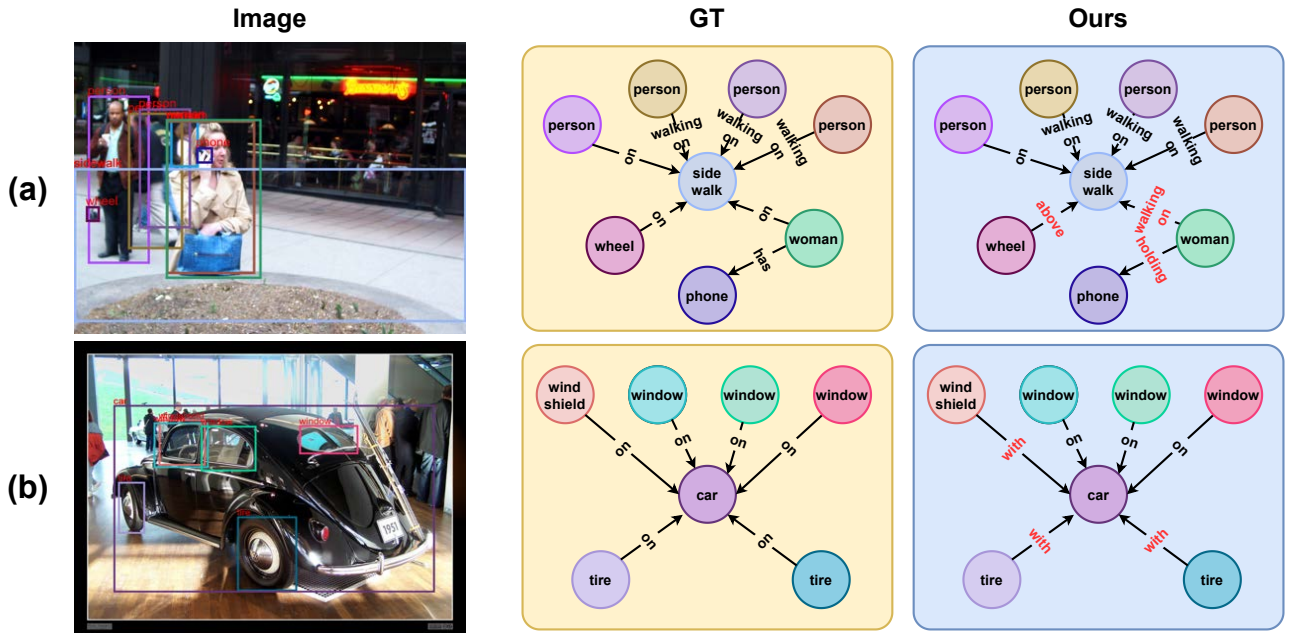


*Figure 7.* Visualization results of scene graph in PredCls subtask on VG dataset.

To prove the qualitative results of the proposed CooK + TF-$l$-IDF, we introduce the visualization results in PredCls subtask. As you can see in the Figure 7, the proposed method makes more accurate relation predictions. As shown in Figure 7 (a) and (b), we can see that our method avoids prediction to the dominant head class and performs prediction to a more delicate class.

## E. TF-/-IDF Code

```
import torch
import torch.nn as nn
import math
import numpy as np

class TfIdfLayer(nn.Module):
  def __init__(self, epsilon=0.0, gamma=0.0, bias=False):
    super(TfIdfLayer, self).__init__()
    self.epsilon = epsilon
    self.gamma = gamma
    self.bias = bias

    if self.bias:
      self.epsilon = nn.Parameter(torch.randn(1), requires_grad=True)
      self.gamma = nn.Parameter(torch.randn(1), requires_grad=True)

  def forward(self, x, labels):

    num_img = len(labels)
    t_id_list = []

    for label in labels:
      for ob_label in label.extra_fields['pred_labels']:
        tf_idf = self.tfidf(ob_label, label.extra_fields['pred_labels'], num_img, labels)
        t_id_list.append(tf_idf)

    weighted_x = torch.tensor(t_id_list).unsqueeze(-1).cuda() * x

    return weighted_x

  def tf(self, t, d):
    return torch.count_nonzero(torch.where(d == t, True, False)).item()

  def idf(self, t, N, docs):
    ni = 0
    for doc in docs:
      ni += t in doc.extra_fields['pred_labels']

    if self.bias:
      return math.log((N + self.epsilon) / (ni + 1 + self.gamma))
    else:
      return math.log(N / (ni + 1))

  def tfidf(self, t, d, N, docs):
    return self.tf(t, d) * self.idf(t, N, docs)
```