Applied Intelligence (2023) 53:26126–26138
https://doi.org/10.1007/s10489-023-04722-1

# LANDMARK: language-guided representation enhancement framework for scene graph generation

Xiaoguang Chang[1] · Teng Wang[2] · Shaowei Cai[2] · Changyin Sun[2]

Accepted: 21 May 2023 / Published online: 18 August 2023

## Abstract

Scene graph generation (SGG) is a sophisticated task that suffers from both complex visual features and the long-tail problem. Recently, various unbiased strategies have been proposed by designing novel loss functions and data balancing strategies. Unfortunately, these unbiased methods fail to emphasize language priors in the feature refinement perspective. Inspired by the fact that predicates are highly correlated with semantics hidden in subject-object pair and global context, we propose LANDMARK (**LAN**guage-gui**D**ed representation enhanceMent frAmewo**RK**) that learns predicate-relevant representations from language-vision interactive patterns, global language context, and object-predicate correlation. Specifically, we first project object labels to three distinctive semantic embeddings for different representation learning. Then, Language Attention Module (LAM) and Experience Estimation Module (EEM) processes subject-object word embeddings to attention vector and predicate distribution, respectively. Language Context Module (LCM) encodes global context from each word embedding, which avoids isolated learning from local information. Finally, module outputs are used to update visual representations and the SGG model's prediction. All language representations are purely generated from object categories so that no extra knowledge is needed. This framework is model-agnostic and consistently improves performance on existing SGG models. Besides, representation-level unbiased strategies endow LANDMARK with compatibility of other methods. Code is available at https://github.com/rafa-cxg/PySGG-cxg.

**Keywords** Scene graph generation · Unbiased method · Vision-language representation learning · Multi-semantics

## 1 Introduction

Scene graph generation (SGG) is a crucial task that benefits image captioning [1, 2], visual question answering [3, 4], video understanding [5, 6] and detection [7, 8]. However, most generated scene graphs face the challenge of trivial predictions, thus far from being applied to practical applications.

Therefore, recent researchers have been working on unbiased methods that elevate Recall of hardly distinguishable predicates. Generally, unbiased methods can be divided into 3 types: data resampling (e.g., BLS [9], GCL [10], DCNet [11]), predicate-aware loss design (e.g., CogTree [12], PCL [13] and FGPL [14]) and logit manipulation (e.g., TDE [15], RTPB [16], FREQ [17]). However, a common drawback is that they rely on explicitly modeling predicate correlations from dataset statistical information [14, 17] or biased predictions [14, 15], which means that they are sensitive to prerequisite changes. For instance, [15] is not effective when training on an unbiased model, hence, confining the SGG model performance. Compared with loss and statistic approaches, language representation learning is much more robust because it learns implicit patterns of predicates and avoids visual feature redundancy, which has not been stressed by unbiased methods before.

However, language representation learning has been adopted by some baseline models. For example, [18] takes word embedding to ground attention on visual features. [10]

✉ Changyin Sun
cysun@seu.edu.cn

Xiaoguang Chang
xg_chang@seu.edu.cn

Teng Wang
wangteng@seu.edu.cn

Shaowei Cai
shaoweicai@seu.edu.cn

[1] School of Cyber Science and Engineering, Southeast University, Nanjing, China

[2] School of Automation, Southeast University, Nanjing, China

utilizes Cross Attention (CA) mechanism for multi-modality learning. [19] introduces transformer-based architecture to bridge the gap between images and texts. However, most of these approaches are not plug-and-play and merely use single representations regardless of different semantic contexts. Therefore, failed to unleash the power of language.

In fact, words have multiple meaning that carries different priors in terms of different semantics, which can guide scene graph generation. Here, we give a multi-semantic reasoning example in Fig. 1. Given this shopping picture, first, human constructs a predicate distribution by the correlation between predicates and "*woman-ball*" as well as their relative position. This knowledge comes from experience and the process is vision-independent. Next, still based on this area, human can build the correlation between subject-object pairs and the visual pattern like "woman's hand is closed to ball", which is a strong "holding" relevant pattern. In contrast, the woman and girl's contours are irrelevant in terms of judging relations. Finally, according to surrounding objects (e.g., balls, girls, lights), humans can infer the *selling* context to avoid predicting *play*, because it is not suitable to this scene.

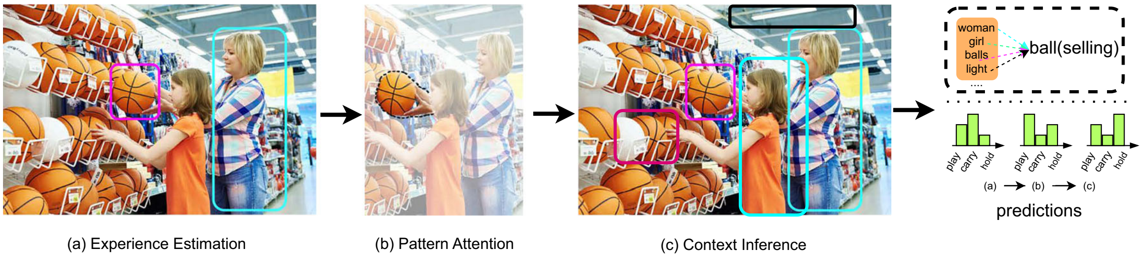
Motivated by these observations, we heuristically design 3 plug-and-play language modules that exploit different language priors behind object categories. Each of them takes detected object classes as input and generates semantic embeddings into different semantic spaces, which are used for extracting priors from language-visual pattern correlations, language context and pair-predicate correlation, respectively. Concretely, 1) **Language Attention Module** projects subject-object word embeddings to a unified semantic matrix, then, channel attention is used to extract attention vector for relation visual feature map, which can learn relevance between object pair and specific predicate-relevant visual patterns 2) **Language Context Module** employs transformer-based encoder to encode the global language context into entity's semantic embedding from a sequence of entity labels. Compared with pretrained word embedding, this module can generate semantic representation that fits the context. These two modules are used for initializing entity and relation visual-based representations, respectively. 3) **Experience Estimation Module** is supervised by marginal probability of subject-predicate and object-predicate to learn the class and spatial aware predicate distribution as likelihood offset. It is worth mentioning that language processing is disentangled from visual feature at the very beginning, so this framework is applicable to most SGG baseline models.

To the best of our knowledge, we are the first to utilize multi-semantic language representation within object labels to achieve unbiased Scene graph generation. The main contributions could be summarized as follows:

1. We propose LANDMARK to introduce language representation learning into unbiased scene graph generation, which stresses the under-explored multiple semantics utilization in the object label.
2. We devise three modules that divide object labels into distinctive semantic spaces, then extract priors of language-vision interactive patterns and semantic context as well as pair-predicate correlation, respectively.
3. Experiments on the SGG benchmark show consistent improvements upon baseline models and compatibility with other unbiased methods, which indicate the effectiveness of multi-semantic language representations induced from object labels.

## 2 Related works

**Scene Graph Generation:** There are mainly two mainstream methods for scene graph generation: Based on context modeling or graph convolutional network (GCN) [20]. The first approach is focused on modeling global information by sequential architecture [16, 17, 21–28]. Chen et al. [16] uses two stacks of Transformer to encode global information. Zellers et al. [17] uses LSTM to encode global context that informs relation prediction. However, merely modeling



(a) Experience Estimation (b) Pattern Attention (c) Context Inference

**Fig. 1** An example of multi-semantic language assistance for relation prediction. Bottom-right corner (green bars) shows 3 candidates' possibility updating process. a) **Experience Estimation**: Humans recall a rough predicate distribution based on the co-occurrence possibility of predicates and object pairs. b) **Pattern Attention**: Using the internal relationship between object pair and visual information for locating predicate-relevant visual patterns. c) **Context inference**: Using context to refine the meaning of subject and object, preventing prediction biases caused by isolated considering object pairs

global context is not sufficient for Scene graph tasks. Another approach [9, 29–31] propagates massage between node and edge features and focuses more on regional pair-wise information. [9] applys a multi-stage graph message propagation between entities and relationship representations. In [30], Yang *et al.* prunes graphs to sparse ones, then an attentional graph convolution network is applied for modulating information flow. [29] utilizes GCN for updating state representations as energy value. Chen *et al.*[32] constructs a graph between entities and all relationship representations and aggregated messages by GRU. Yet, this approach suffers from insufficient global context encoding. Our method considers both pair-wise and global contexts for representation refinement.

**Unbiased Scene Graph Generation:** USGG has been a hotspot research area since existing SGG datasets are long-tail. FREQ [17] uses a distributed-based prior bias to predictions. Chen et al. [16] utilizes a resistance bias item for the relationship classifier during training to optimize the loss value. CogTree [12] proposed a loss based on the automatically built cognitive structure of the relationships from the biased SGG predictions. [33] designs two separate classifiers for head and tail predicates. Through mentioned methods get a remarkable boost on specific baseline models in terms of mRecall@k [34], they are sensitive to training data distribution and easily overfitting to tail classes. More recently, [35] observes that directly using visual features results in biased relation predictions. Later, [36] avoids directly using visual features by utilizing Hermitian inner product to embed visual features into complex space. These observations inspire us to design a network that learns language-guided visual features and explores multi-semantics from words.

**Multimodel Learning:** Multimodel representation learning has been widely explored in zero-shot image retrieval [37], text-video retrieval [38, 39], and object detection [40]. In the SGG task, language and commonsense are treated as multimodel representations. [19] parses sentences in the image-text dataset to extract triplet as supervision. [41] incorporated commonsense by unifying the form of scene graph and commonsense graph. [10] adopts cross-attention modules between vision and text embedding. However, existing methods treat language modality as a single representation, which loses a lot of information from other perspectives.

## 3 Methodology

**Problem formulation:** Given an image $I$, scene graph generation aims to predict entity class set $\mathcal{C}_e$, coordinate set $\mathcal{B}_e$ and relation set $\mathcal{C}_r$. Generally, existing SGG models receive visual entity and predicate (or node and edge) representations from the backbone. Then a graph $\mathcal{G} = \{\mathcal{C}_e, \mathcal{B}_e, \mathcal{C}_r\}$ can be formulated as

$$\mathcal{G} = P(\mathcal{C}_r, \mathcal{B}_e, \mathcal{C}_e | I) = SGG(N, E), \tag{1}$$

where $N = \{e_i\}_{i=1}^{n}$ and $E = \{e_{ij}\}$ are the set of entity and predicate representations. In this paper, we aimed to update $N$ and $E$ by incorporating semantic priors.

**Framework overview:** LANDMARK consists of three semantic learning modules, i.e., *Language Attention Module* (LAM), *Language Context Module* (LCM), and *Experience Estimation Module* (EEM). The framework architecture is shown in Fig. 2. First, we obtain $N$, $E$ from ROI Pooling, $\mathcal{C}_e$, and $\mathcal{B}_e$ from the classifier head. Then, the semantic extraction operation converts labels $\{c_i\}$ to semantic embeddings for each module. For LAM, semantic embeddings of subject $c_i$ and object $c_j$ are transposed and multiplied to a semantic matrix, then channel attention transfers the matrix to the attention vector and updates relation representation. LCM encodes semantic embeddings of all entity labels $\mathcal{C}_e$ presented in the image, generating context-aware semantic entity feature and concatenating it with visual entity representation. The updated entity and relation representation are

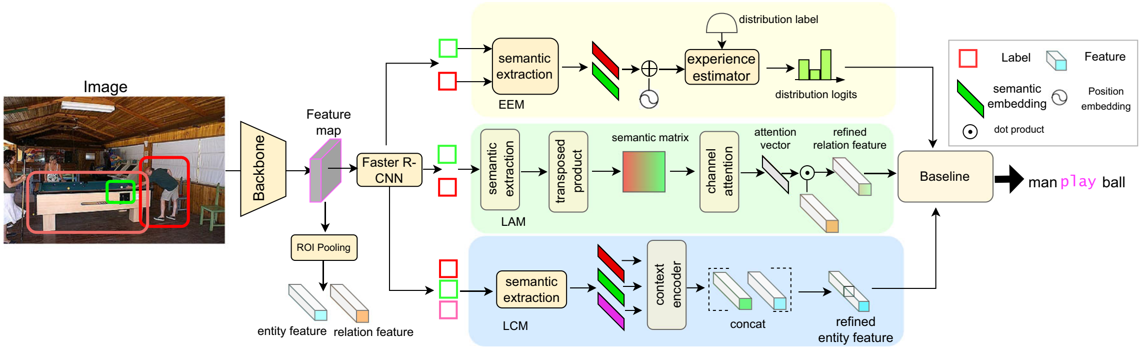


**Fig. 2** LANDMARK architecture. The image first goes through a generic object detector (Faster R-CNN) to get the predicted object's label and ROI feature. Then labels are served as three modules input (i.e., EEM, LAM, LCM) and calculated distinctive semantics. Finally, LAM and LCM outputs are used to refine relation and entity representations, respectively. EEM's output is served as a prediction offset

passed through the baseline model. For EEM, the distribution label is generated to supervise experience estimator, which combines subject-object semantic embedding with position embedding to yield distribution logits. Finally, generated logits are used to update the final predicate likelihood.

### 3.1 Semantic extraction

Semantic extraction is used to transfer labels to the corresponding semantic space, which is applied to three modules independently. Specifically, the semantic extractor consists of three operations:

$$\begin{cases} f_{se}^{sub}(c_i) = w_s^T c_i, \\ f_{se}^{obj}(c_j) = w_o^T c_j, \\ f_{se}^{ent}(c_e) = w_e^T c_e, \end{cases} \tag{2}$$

where the first and second operations are used in LAM and EEM for subject and object projection. Considering that the same word as subject or object may have contrastive meanings (e.g., *eating* could be a possible predicate if “man” is subject, which is impossible when “man” is object), we use different weights $w_s$ and $w_o$ to project subject and object to semantic embedding. The last operation is used in LCM, since all labels are treated as objects, we use unified $w_e$ as semantic embedding weight.

In fact, $f_{se}$ could be any projection function as long as the input is object labels. Here, we only use a naive 1-layer linear function to prove the extraction’s effectiveness.

### 3.2 Language Attention Module

This module aims to learn the prior between object pair and visual predicate-relevant patterns within visual relation representation $e_{ij}$. The original feature extraction network (e.g., Reset [42]) keeps both spatial and semantic information. Specifically, different channels focus on different visual patterns. However, relation visual features inevitably mix up with a huge amount of irrelevant background information, so there is a need for channel selection. Heuristically, given a specific subject-object pair (e.g., *boy-basketball* or *boy-street*), the visual feature should have different activation. Therefore, we design a label-aware channel attention mechanism. Specifically, given subject i and object j, we first generate a semantic matrix $x_{ij}$ as a unique representation of word-vision correlation:

$$x_{ij} = f_{se}^{sub}(c_i) \otimes f_{se}^{obj}(c_j)^T, \tag{3}$$

where $\otimes$ refers to matrix multiplication. We achieve channel attention by a series of 2D convolutions with the spatial pooling on $x_{ij}$ to get attention vector $e_{ij}^c$:

$$e_{ij}^c = \sigma(G_{\text{pooling}}(G_{\text{conv}}^{n_c}...\sigma(G_{\text{conv}}^1(x_{ij})))) \in \mathcal{R}^{C,1}, \tag{4}$$

where $C$ is as the channel number of visual relation feature $e_{ij}$, $n_c$ is the number of 2D convolution layers, $G_{\text{pooling}}$ is the pooling operation, $\sigma$ is the activation function. Finally, the channel weights $e_{ij}^c$ will be used for updating $e_{ij}$, so that irrelevant channels to relation discrimination will be suppressed:

$$\hat{e}_{ij} = e_{ij} \times e_{ij}^c, \tag{5}$$

where $\hat{e}_{ij}$ is the refined relation representation, × is the dot product operator.

### 3.3 Language Context Module

Compared with visual information, a single word is semantically isolated from other components in a sentence. Though we devise LAM and EEM for semantic extraction, the utilized pairwise labels are confined to local semantics, which is insufficient for comprehensive semantic inference. Hence, LCM is aiming at addressing the global context deficiency problem. This module includes a semantic extractor and context encoder. The context encoder consists of a multi-layer transformer encoder with Multi-Head Self-Attention (MHSA) [26] and Feed Forward Network (FFN) [26]. The structure is illustrated in Fig. 3. Concretely, given an image, supposed there are $n$ entities, the input sequence $X$ could be described as follows:

$$X = \{s_0, s_1, ..., s_n\}, \tag{6}$$

where

$$s_i = \gamma(f_{se}^e c_i + p_i) \in \mathbb{R}^d. \tag{7}$$

Here, $c_i$ and $p_i = \phi[x_i, y_i, w_i, h_i]$ refer to the class label and entity’s position embedding of the $i$-th entity. $x_i, y_i, w_i, h_i$ are center coordinates, width, and height of the object i. $\gamma$ denotes the learnable linear transformation. Where $d$ is the dimension of each element in the sequence. We first reiterate standard Scaled Dot-Product Attention [26] as below:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{8}$$

then, Multi-Head Self Attention is formulated as

$$\begin{aligned} \text{MHSA}(X) &= \text{Concat }(\text{head}_1, \ldots, \text{head}_\text{h})\, W^O, \\ \text{head}_\text{i} &= \text{Attention}\left(XW_i^Q, XW_i^K, XW_i^V\right), \end{aligned} \tag{9}$$

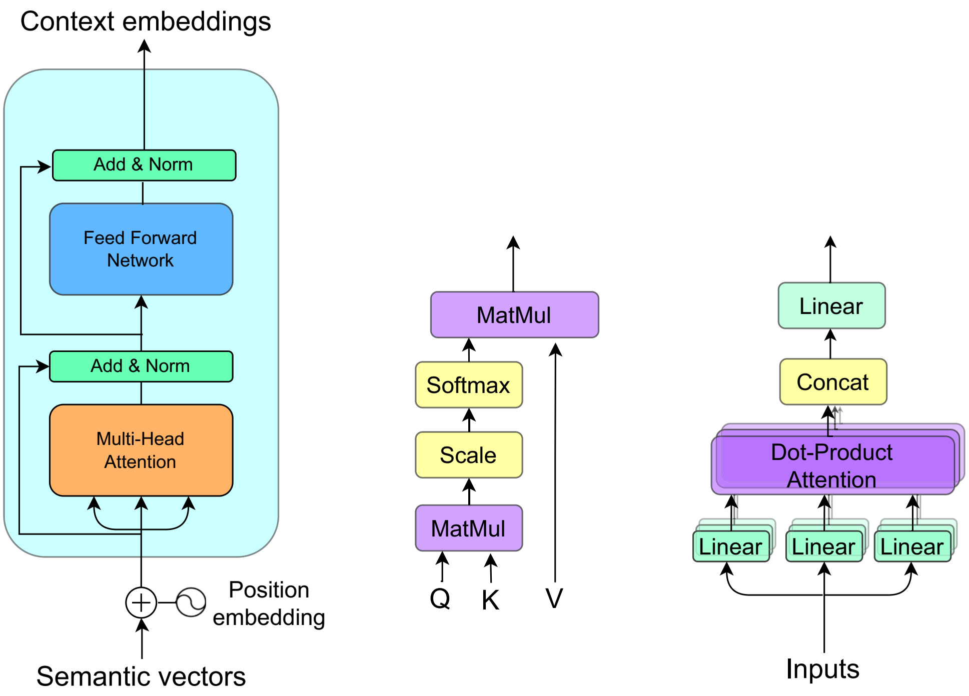


**Fig. 3** Context Encoder block structure (left). Dot-Product Attention (middle). Multi-Head Self Attention (right)

where $W_i^Q$, $W_i^K$, $W_i^V$ and $W^O$ are parameter matrices. The $b$-th layer output $X_b$ can be denoted as

$$X'_b = \mathrm{MHSA}(\mathrm{LN}(X_{b-1})) + X_{b-1}, \tag{10}$$

$$X_b = \mathrm{FFN}(\mathrm{LN}(X'_b)) + X'_b, \tag{11}$$

where $X_{b-1}$ is the $(b-1)$th layer, and we set $X_0 = X$. LN is layer normalization. Differing from previous works [16], LCM takes a sequence of entity labels as inputs, so the module can learn semantic entity representation on top of high-level information.

### 3.4 Experience Estimation Module

This module is designed to learn the relationship distribution prior to the subject-object pair, as compensation for cross entropy loss. EEM consists of a semantic extractor, experience estimator, and distribution label for supervision. Since entity class and position both have an influence on judging relation, This module utilizes both classes and position information to learn precise predicate distribution. First, we embed entity label i, j and position embedding $p_{ij}$ to high dimension representation space, then, the experience estimator predicts the predicate distribution $d_{ij}$ between subject i and object j:

$$p_{ij} = \phi_p[x_i, y_i, w_i, h_i, x_j, y_j, w_j, h_j], \tag{12}$$

$$d_{ij} = \varphi\left[\phi_s(f_{se}^{sub}(c_i))\phi_o(f_{se}^{obj}(c_j)), p_{ij}\right], \tag{13}$$

where $[\cdot, \cdot]$ refers to concatenation operation, $\phi_p, \phi_s, \phi_o, \varphi$ are fully connected layers with RELU as an activation function. Finally, we merge relationship distribution $d_{ij}$ with the prediction from the baseline, which could be described as

$$\hat{d}_{i,j} = SGG(\hat{n}_{ij}, \hat{e}_{ij}) + d_{ij}, \tag{14}$$

where $\hat{d}_{i,j}$ is the updated prediction likelihood. $\hat{n}_{ij}$ and $\hat{e}_{ij}$ are enhanced entity and relation feature obtained by LAM and CAM (Sections 3.2, 3.3), respectively.

**Distribution Label Generation:** Dataset annotations inherently reflect human commonsense, so we manage to generate accurate distribution labels from the dataset. For subject i, object j, we obtain $m_i^{sub}$ $m_j^{obj}$ as "subject-predicate" and "predicate-object" marginal distributions. Since $m_i^{sub}, m_j^{obj}$ are independent distribution, we calculate joint possibility $p_{ij}^{joint}$ as

$$p_{ij}^{joint} = m_i^{sub} \odot m_j^{obj}, \tag{15}$$

where $\odot$ denotes the element-wise product. Though EEM is like FREQ [17] generates predicate distribution from statistics, FREQ directly counts triplets $\langle subject, predicate, object\rangle$ occurrence. However, some triplet samples are scarce in training samples, so it is hard to establish an informative distribution prior. In contrast, EEM uses joint possibility as labels to infer the predicate distribution. Figure 4 is a typical example of generated joint possibility $p_{ij}^{joint}$ of a triplet that not occurred in the training set. In this circumstance, FREQ could not work due to zero sample number,

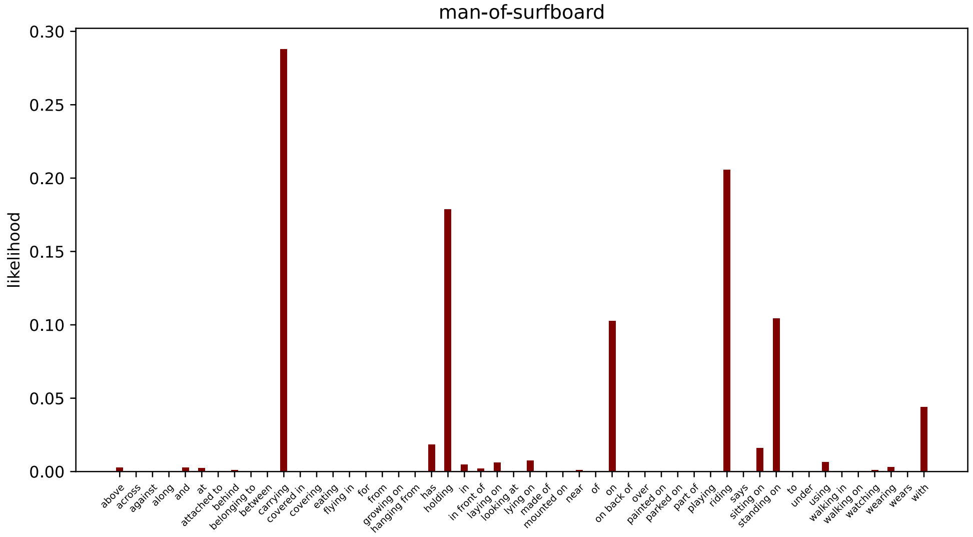


**Fig. 4** Joint possibility for the subject *man* and object *surfboard*. The ground truth triplet ⟨man,of,surfboard⟩ is sampled from zero-shot split (where triplets are only existed in the evaluation set but not occurred in the training set)

whereas, we can see that the top 5 highest likelihoods are reasonable and include many possible scenarios. In contrast, the predicate "of" is ambiguous. A better predicate could be one of top likelihoods from joint possibility.

Considering that introducing position information makes accurate predicate possible, we design a fusion function for mitigating joint possibility and true predicate label. The distribution label $l_{ij}$ can be denoted as

$$l_{ij} = \mu \times p_{ij}^{joint} + (1 - \mu) \times \text{onehot}[r_{ij}], \tag{16}$$

where $\mu$ is a factor regulating the proportion of marginal frequency. $r_{ij}$ refers to the true predicate label between subject $i$ and object $j$.

**Objective Function:** we choose MSE loss between distribution label $l_{i,j}$ and predicted distribution $d_{i,j}$, which is denoted as

$$\text{MSE} = \frac{1}{K} \sum_{i=1}^{K} (d_i, l_i)^2, \tag{17}$$

where K is the number of relation categories in dataset.

### 3.5 Baseline model

Any off-the-shelf two-stage scene graph generation model can be used as a baseline. It could be either a sequential model or a graph neural network, as long as it needs entity and relation features for prediction.

## 4 Experiments

In this section, we first introduce the experiment settings in our experiments. Then, we test our framework's effectiveness on several SGG models and conduct experiments to analyze the compatibility between our method and the state-of-art unbiased strategies. Finally, detailed analyses and qualitative studies are presented to further verify LANDMARK's superiority from different perspectives.

### 4.1 Experiment settings

**Datasets** We employ the widely adopted Visual Genome [43] dataset's split VG150 [44] to train and evaluate our framework. The VG150 dataset contains the most frequent 150 object categories and 50 predicate categories in VG. It consists of more than 108k images, with 70% of images held out for training and 30% for testing. Among the training set, 5000 images are used for evaluation.

**Tasks.** We consider three conventional sub-tasks of scene graph generation to evaluate our framework. 1) *Predicate Classification (PredCls)* predicts relationships between each object pair given their ground-truth bounding boxes and classes. 2) *Scene Graph Classification (SGCls)* predicts the object classes and their relationships given the ground-truth bounding boxes of objects. 3) *Scene Graph Detection (SGDet)* needs to detect object classes and bounding boxes, then predict their relationships.

**Evaluation Metrics** we report widely accepted Recall@k [34] and mean Recall@k [32] for evaluating the model's performance. Recall@k computes the fraction of times the correct relationship is predicted in the top k predictions of one image, the mean Recall is used to evaluate unbiased performance. However, both metrics are calculated based on the image level. In order to evaluate EEM, we need a metric to measure on prediction level. Therefore, the TOP-N Recall@K is proposed to allow top N scored predicates in one

**Table 1** Comparison between baseline models and LANDMARK under three sub-tasks on the VG dataset

| Model | Method | PredCls | | SGCls | | SGDet | | Params(M)/GFLOPs |
|---|---|---|---|---|---|---|---|---|
| | | mR@50/100 | R@50/100 | mR@50/100 | R@50/100 | mR@50/100 | R@50/100 | |
| IMP [44] | baseline | 16.88/18.02 | 66.8/68.25 | 7.57/8.08 | 38.9/40.17 | 6.00/7.30 | 27.24/34.24 | 336.3/206.4 |
| | LANDMARK | **19.54/21.06** | 64.89/66.61 | **9.89/10.49** | 33.63/34.89 | **6.47/8.00** | 24.34/29.05 | 356.5/208.8 |
| Transformer [26] | baseline | 19.13/20.3 | 65.59/67.21 | 10.25/10.72 | 39.3/40.5 | 7.99/9.68 | 26.91/31.26 | 330.6/205.6 |
| | LANDMARK | **22.19/23.87** | 65.48/66.92 | **11.94/12.84** | 36.62/37.73 | **8.23/10.43** | 28.76/32.70 | 348.5/207.8 |
| G-RCNN [30] | baseline | 16.46/17.28 | 66.27/67.9 | 9.67/10.17 | 39.3/40.5 | 4.89/5.95 | 30.37/34.47 | 366.1/207.1 |
| | LANDMARK | **19.24/20.51** | 65.76/67.50 | **11.14/11.62** | 42.80/43.55 | **5.52/7.21** | 24.61/29.69 | 386.4/209.6 |
| Motifs [17] | baseline | 18.79/19.69 | 66.17/67.72 | 9.39/9.97 | 41.57/42.7 | 6.24/7.54 | 29.82/33.65 | 367.1/211.5 |
| | LANDMARK | **22.37/23.79** | 59.34/61.16 | **14.36/15.25** | 33.57/35.02 | **7.87/10.56** | 24.65/28.87 | 389.8/214.4 |
| BGNN* [9] | baseline | 17.26/18.29 | 66.14/67.65 | 10.30/10.83 | 39.94/41.17 | 6.46/8.22 | 30.90/35.36 | 341.9/205.0 |
| | LANDMARK | **20.35/22.63** | 55.10/57.16 | **11.64/12.79** | 33.3/34.89 | **7.34/8.97** | 24.34/29.05 | 360.1/207.2 |

* denotes that the default unbiased strategy in the original paper is not applied. All baselines are reimplemented on our codebase.
The best methods under mR metric are marked in bold

**Table 2** Compatibility test of unbiased methods and LANDMARK under three sub-tasks on the VG dataset

| Method | PredCls | | SGCls | | SGDet | |
|---|---|---|---|---|---|---|
| | mR@50/100 | R@50/100 | mR@50/100 | R@50/100 | mR@50/100 | R@50/100 |
| $\text{Motifs}_{\text{TDE}}$ | 22.87/25.90 | 35.28/41.11 | 13.76/15.36 | 23.88/27.42 | 7.87/9.55 | 9.83/13.06 |
| $\text{Motifs}_{\text{BLS}}$ | 30.64/32.76 | 54.56/56.27 | 20.21/21.09 | 34.11/35.00 | 13.20/15.97 | 25.27/28.88 |
| $\text{Motifs}_{\text{BLS+TDE}}$ | 26.19/30.84 | 17.01/18.89 | 17.31/19.28 | 19.63/21.43 | 9.18/11.89 | 8.34/10.27 |
| $\text{Motifs}_{\text{TDE+LANDMARK}}$ | 24.59/29.27 | 39.24/45.25 | 14.43/17.03 | 28.70/32.28 | 8.98/11.35 | 9.92/12.85 |
| $\text{Motifs}_{\text{BLS+LANDMARK}}$ | **33.62/35.69** | 55.29/56.80 | **21.43/22.05** | 36.02/36.85 | **13.86/16.31** | 25.10/29.12 |
| $\text{BGNN}_{\text{TDE}}$ | 19.23/21.59 | 32.75/36.45 | 13.09/14.18 | 28.66/31.20 | 8.65/11.01 | 20.36/26.00 |
| $\text{BGNN}^{*}_{\text{BLS}}$ | 30.64/32.76 | 55.39/56.8 | 16.69/17.83 | 35.70/36.94 | 13.17/15.56 | 23.54/27.56 |
| $\text{BGNN}_{\text{BLS+TDE}}$ | 31.88/34.89 | 30.37/33.33 | 16.97/18.70 | 21.62/23.52 | 11.60/14.32 | 19.45/24.11 |
| $\text{BGNN}_{\text{TDE+LANDMARK}}$ | 20.32/22.78 | 50.21/53.85 | 14.56/15.58 | 35.31/37.38 | 9.68/12.41 | 25.72/27.98 |
| $\text{BGNN}_{\text{BLS+LANDMARK}}$ | **34.41/36.43** | 53.99/55.42 | **18.27/19.12** | 34.55/35.80 | **14.08/18.34** | 24.87/29.41 |

* means this Baseline model and unbiased method are proposed in the same paper. All methods are reimplemented on our codebase. The best methods under mR metric are marked in bold.

prediction as candidates, then, $N \times K$ number of candidates are used for calculating Recall, i.e.,

$$\text{TOP-N Recall@K} = \frac{correct(\{\text{ candidates}\}_{N\times K})}{N_{gt}},$$

when N=1, TOP-N Recall@K is equal to Recall@K.

**Implementation Details** We use pretrained Faster R-CNN [45] with backbone ResNeXt-101-FPN [46] as object detector. ROIAlign's [46] resolution is 7. We froze its weight during training. We use pretrained GloVe [47] weight as initial $w_s, w_o, w_e$ in semantic extractor. For EEM, we use 3-layers MLP $\Phi_s$, $\Phi_o$, $\Phi_p$ with 1024 neurons. $\varphi$ is a 2-layers MLP with hidden dimension 4096. $\mu$ in Eq. 16 is set to 0.3 for unbiased methods, and 0.7 for baseline models. For Eq. 4, we choose two $3 \times 3$ convolution layers to generate a 256-dim attention vector. For LCM, We choose the context encoder with 4 layers and 8 heads, entity dimension $d = 512$. For training, approximately 10000 iterations are enough for each baseline. The basic learning rate is 0.01 and the batch size is 16. We choose the SGD optimizer for optimization.

## 4.2 Comparison with baseline models

Table 1 shows mRecall & Recall of 5 baseline models with or without our framework LANDMARK. Baseline models include GCN-based models like G-RCNN [30], and BGNN [9], context modeling networks like IMP [44], Transformer [26], and Motifs [17]. It is worth mentioning that we do not deploy any unbiased strategies on these models. We observe that incorporating our proposed LANDMARK leads to a consistent mRecall improvement in all three tasks for all baseline models, which demonstrates the robustness of our approach. For mR@100, our model average improvements are 3.66%, 2.64%, and 1.37% on three tasks. The improvements might be attributed to the fact that multi-semantic language representations indeed facilitate visual representations. It is not surprising that improvement consecutively

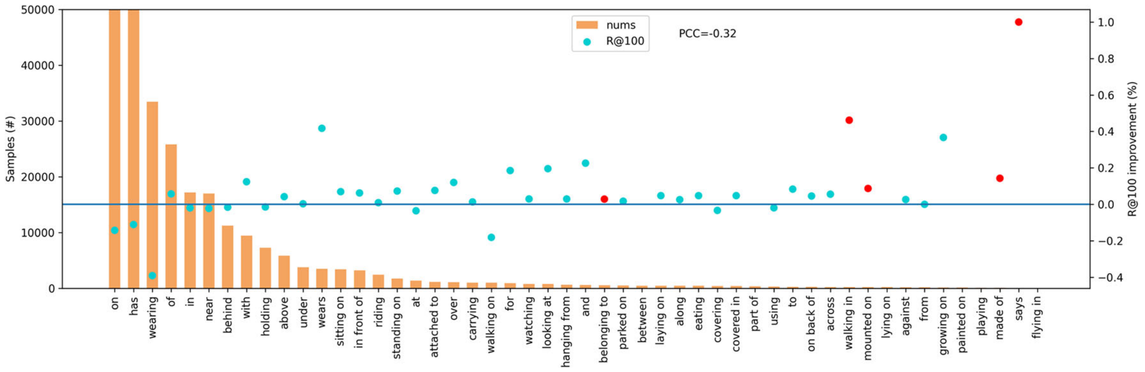


**Fig. 5** The number of data samples (bars) and Recall@100 improvements (dots) of LANDMARK over BGNN on PredCls task. The red dots indicate that BGNN Recall@100 is zero. PCC is the abbreviation of Pearson Correlation Coefficient

**Table 3** Ablation studies of three modules on BGNN+BLS

| Module | | | PredCls | SGCls | SGDet |
|---|---|---|---|---|---|
| EEM | LAM | LCM | mR@50/100 | mR@50/100 | mR@50/100 |
| | | | 30.64/32.76 | 16.69/17.83 | 13.17/15.56 |
| ✓ | | | 31.75/33.00 | 17.27/18.18 | 13.54/15.92 |
| ✓ | ✓ | | 32.06/33.36 | 17.75/18.89 | 13.84/16.42 |
| | ✓ | ✓ | 34.10/36.07 | 18.07/19.01 | 13.89/17.00 |
| ✓ | ✓ | ✓ | **34.41/36.43** | **18.27/19.12** | **14.08/18.34** |

The best methods under mR metric are marked in bold

shrinks in three tasks, due to inaccurate class and position predicted by pretrained object detector. Besides, Recall shows drops to different extents, which is a common characteristic of an unbiased method.

We also measure the total parameters (M) and GFLOPs in Table 1. There are ∼20M parameters increasing and ∼2.5GFLOPs additional computation cost when adding LANDMARK. The relative proportions are about 6% and 1.2%, respectively. It is attributed to lightweight module design and adoption of low-dimensional inputs (i.e., language rather than image inputs.)

### 4.3 Compatibility with unbiased methods

A tricky problem of unbiased SGG strategies is that most of them have demanding working conditions, for example, the baseline model's performance or sample distribution. Hence, we test our framework's compatibility with other unbiased methods by stacking two methods together, the mRecall & Recall are listed in Table 2. The listed strategies belong to different types, e.g., data resampling: BLS [9], logit manipulation: TDE [15], and feature refinement: LANDMARK (ours). According to this table, there are several findings:

- Applying LANDMARK with other methods is effective. For instance, BLS with LANDMARK on BGNN gets new SOTA performance. The reason has two: 1) LANDMARK has a distinctive semantic feature enhancement strategy, which does not conflict with other methods. 2) Most of the unbiased methods are designed to obtain priors from biased prerequisites (e.g., baseline, long-tailed dataset), whereas LANDMARK is model-agnostic, that is, baseline does not affect LANDMARK's inference.
- Only a tiny improvement, or even decrease in mR@K occurred when using BLS and TDE together. For example, Motif+BLS+TDE results in an obvious decrease in mR@k. This suggests that these methods are sensitive to changes in external circumstances (e.g., sampling distribution, baseline model capability).
- Our network does not sacrifice Recall a lot. BLS+LANDMARK on Motifs has a higher Recall than the one without ours. By contrast, using BLS+TDE remarkably impair the Recall performance. We speculate that existing unbiased methods do not explore real discrepancies between predicates, but only increase the likelihood of tail predicates.

### 4.4 Predicate analysis

Shown in Fig. 5, we present the number of data samples in VG dataset and Recall@100 improvements of LANDMARK. In 50 predicates, only 11 predicates predicted by baseline are superior to LANDMARK. Though some predicates have obvious decreases, e.g., "wearing". It is compensated by the increase of "wears". Besides, there are 5 hard-to-predict predicates (i.e. *belong to, walking in, mounted on, made of, says*) that are recalled by LANDMARK.

For exploring the correlation between Recall improvements and data distribution, Pearson Correlation Coefficient (PCC) is used. PCC= -0.32 shows a weak negative correlation between dataset bias and LANDMARK improvements. It should be noticed that some unbiased methods overfitting to tail classes so that Recall improvements show a strong negative correlation with the number of samples (e.g., TDE: PCC=-0.56). Therefore, LANDMARK is robust to data distribution.

**Table 4** Comparison of EEM and FREQ on baseline Motifs

| Module | PredCls<br>Top-1/5 R@100 | SGCls<br>Top-1/5 R@100 | SGDet<br>Top-1/5 R@100 |
|---|---|---|---|
| FREQ [17] | 14.72/**42.47** | 9.71/27.56 | 7.66/18.32 |
| EEM | **22.74/40.38** | **14.76/29.14** | **14.13/23.43** |

The best methods under the Top-N R metric are marked in bold

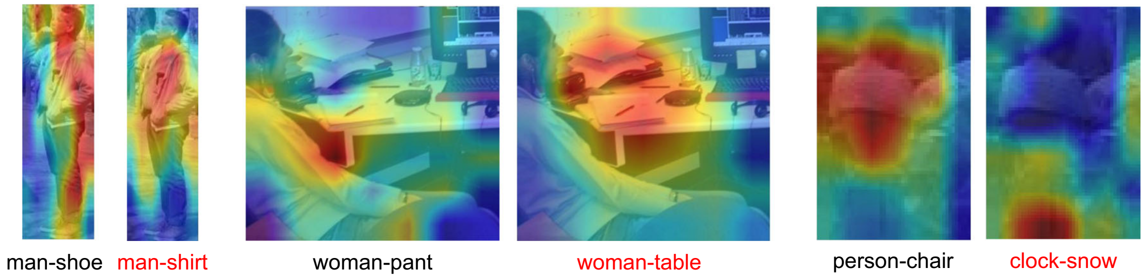


**Fig. 6** Visualization of LAM. We visualize the union area by giving ground truth subject-object pair and random generated pair (red words), which shows the connection between word pairs and particular visual features

## 4.5 Ablation studies

We investigate each LANDMARK component by incrementally adding EEM, LAM, and LCM to the BGNN+BLS in Table 3. The results indicate that: 1) each component is helpful for the whole framework, and no conflict between them. Improvements show that three modules extract distinctive semantics from the same label inputs. 2) EEM mainly improves PredCls more than other tasks, which might caused by inaccurate position and object label predictions in last two tasks. 3) LCM and LAM consistently promote performances of each task. Because priors from language context and correlation of word-visual patterns are relatively robust to misclassified but similar semantic object labels.

## 4.6 Analysis of experience estimation module

As mentioned before, Experience Estimation Module independently outputs predicate predictions like Frequency Baseline (FREQ) [17]. Therefore, we evaluate TOP-N Recall of EEM and FREQ trained on Motifs in Table 4. Except for Top-5 on PredCls task, all performances of EEM outperform FREQ, and the gap enlarged along with task difficulty increased. It is attributed to supervision from joint possibility that alleviates the deficiency of rare subject-object samples. Besides, position information introduced in EEM makes it accurate when inferencing the same object pair.

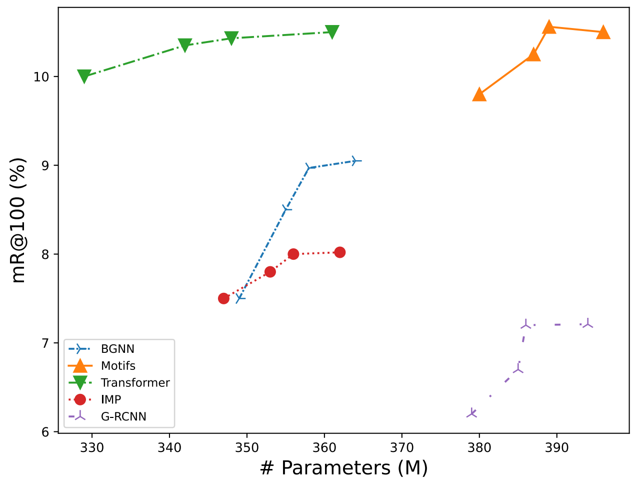


**Fig. 7** mR@100 of SGG models with different model sizes (1,3,4,6 layers of LCM) on SGDet task

## 4.7 Analysis of Language Attention Module

To validate LAM's effectiveness, we visualize heatmaps of relation representation $\hat{e}_{ij}$ (Eq. 5) generated by BGNN+BLS+LANDMARK with ground truth or random generated subject-object pair (with red words) in Fig. 6. Intuitively, we can notice that attention area is correlated with a given subject and object. For instance, in the leftmost two images, the attention area transfers from the foot to the middle of the man's body when the object changes from *shoe* to *shirt*. While given irrelevant words in the rightmost two images, e.g., *clock-snow*, this module seems to be interested in the top left and bottom area, which suggests LAM can associate words to related visual pattern without given coordinates.

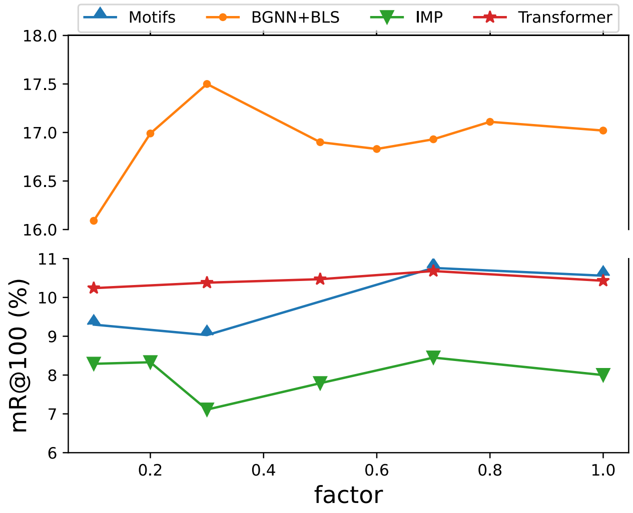


**Fig. 8** mR@100 performance of SGG models with different $\mu$ factor on SGDet task

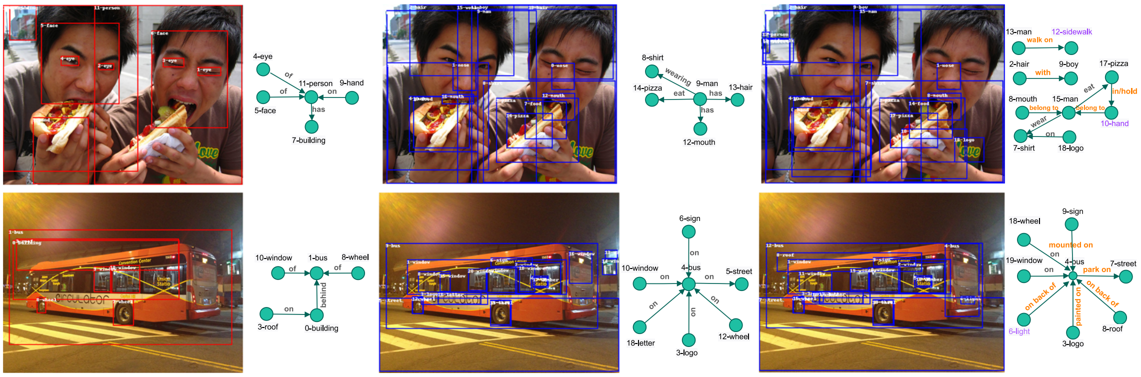


**Fig. 9** Visualization Results: In three columns, we present scene graphs generated by annotations, BGNN with BLS (unbiased model), and LANDMARK, respectively. Relations and entities that neither occur in annotations nor baseline are marked with orange and purple, respectively

### 4.8 Analysis of Language Context Module

We test mRecall@100 performance of the different numbers of LCM transformer layers in Fig. 7. For each model, we record 1,3,4,6-layer LCM's performance and corresponding parameters. The figure shows mRecall has a noticeable boost with the number of layers increasing from 1 to 4. However, the 6-layer structure could not bring sufficient performance improvement in consideration of parameters increase. Therefore, 4 layers are adopted for LCM.

### 4.9 Evaluation of $\mu$ factor

Figure 8 shows 4 model's mRecall@100 on SGDet task with different factor $\mu$ in Eq. 16. We find that for baseline models, factor=0.7 is preferable, and for unbiased methods, 0.3 is better. This indicates that EEM mainly learns diversified predicates.

### 4.10 Qualitative studies

We visualize scene graph generation result from annotations, BGNN with BLS, and BGNN+BLS+LANDMARK on PredCls task in Fig. 9. Intuitively, annotations and BGNN+BLS tend to predict relationships between "less informative pairs" (e.g., *person-eye*, *roof-building* and *light-bus*). However, LANDMARK can further detect relationships between *man-sidewalk* or *roof-bus*. Besides, LANDMARK focuses on high-level semantic and positional relationships. For instance, "hand hold pizza" and "light on back of bus" prove that our framework successfully learns the position relationships between objects.

## 5 Conclusion

In this paper, we first point out inadequate language modality utilization in precious SGG methods. Motivated by language's polysemy, we purpose a representation enhancement framework (LANDMARK) for the SGG task, featured by multi-semantic extraction from object labels. This plug-in network explores word-vision correlated patterns and language context from word embedding and learns predicate distribution from subject-object pair with the position. Compared with other unbiased methods, our framework is a new approach from the representation refinement perspective. Experiment and analysis show consistent improvement in Baseline models and great compatibility with other unbiased methods.

**Acknowledgements** This research was in part supported by the National Natural Science Foundation of China (Grant No. 62273093, 61921004), and in part by ZhiShan Scholar Program of Southeast University. This research is also supported by the Big Data Computing Center of Southeast University.

**Code availability** The code are openly available in [PySGG-cxg] at https://github.com/rafa-cxg/PySGG-cxg.

## Declarations

**Competing interests** No potential conflict of interest was reported by the authors.

## References

1. Gu J, Joty S, Cai J, Zhao H, Yang X, Wang G (2019) Unpaired image captioning via scene graph alignments. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp 10323–10332
2. Xu N, Liu AA, Liu J, Nie W, Su Y (2019) Scene graph captioner: Image captioning based on structural visual representation. J Vis Commun Image Represent 58:477–485
3. Shi J, Zhang H, Li J (2019) Explainable and explicit visual reasoning over scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 8376–8384

4. Qian T, Chen J, Chen S, Wu B, Jiang YG (2022) Scene graph refinement network for visual question answering. IEEE Trans Multimedia 1. https://doi.org/10.1109/TMM.2022.3169065
5. Teng Y, Wang L, Li Z, Wu G (2021) Target adaptive context aggregation for video scene graph generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp 13688–13697
6. Han Y, Zhuo T, Zhang P, Huang W, Zha Y, Zhang Y, Kankanhalli M (2022) One-shot video graph generation for explainable action reasoning. Neurocomputing 488:212–225
7. Woźniak M, Wieczorek M, Siłka J (2022) Deep neural network with transfer learning in remote object detection from drone (DroneCom '22). Association for Computing Machinery, New York. pp 121–126. https://doi.org/10.1145/3555661.3560875
8. Siłka W, Wieczorek M, Siłka J, Woźniak M (2023) Malaria detection using advanced deep learning architecture. Sensors 23(3). https://doi.org/10.3390/s23031501
9. Li R, Zhang S, Wan B, He X (2021) 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
10. Dong X, Gan T, Song X, Wu J, Cheng Y, Nie L (2022) Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. arXiv preprint arXiv:2203.09811
11. Han X, Dong X, Song X, Gan T, Zhan Y, Yan Y, Nie L (2022) Divide-and-conquer predictor for unbiased scene graph generation. IEEE Trans Circ Syst Vid Technol 32(12):8611–8622
12. Yu J, Chai Y, Wang Y, Hu Y, Wu Q (2020) Cogtree: Cognition tree loss for unbiased scene graph generation. arXiv preprint arXiv:2009.07526
13. Tao L, Mi L, Li N, Cheng X, Hu Y, Chen Z (2022) Predicate correlation learning for scene graph generation. IEEE Trans Image Process 31:4173–4185
14. Lyu X, Gao L, Guo Y, Zhao Z, Huang H, Shen HT, Song J (2022) Fine-grained predicates learning for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 19467–19475
15. Tang K, Niu Y, Huang J, Shi J, Zhang H (2020) Unbiased scene graph generation from biased training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 3716–3725
16. Chen C, Zhan Y, Yu B, Liu L, Luo Y, Du B (2022) Resistance training using prior bias: toward unbiased scene graph generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 36. pp 212–220
17. Zellers R, Yatskar M, Thomson S, Choi Y (2018) Neural motifs: Scene graph parsing with global context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 5831–5840
18. Gkanatsios N, Pitsikalis V, Koutras P, Maragos P (2019) Attention-translation-relation network for scalable scene graph generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops
19. Zhong Y, Shi J, Yang J, Xu C, Li Y (2021) Learning to generate scene graph from natural language supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp 1823–1834
20. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations
21. Zhou H, Yang Y, Luo T, Zhang J, Li S (2022) A unified deep sparse graph attention network for scene graph generation. Pattern Recog 123:108367
22. Lin X, Ding C, Zeng J, Tao D (2020) Gps-net: Graph property sensing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 3746–3753
23. Wang W, Wang R, Shan S, Chen X (2019) Exploring context and visual pattern of relationship for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
24. Woo S, Kim D, Cho D, Kweon IS (2018) Linknet: Relational embedding for scene graph. Advances in Neural Information Processing Systems 31:558–568
25. Li Y, Ouyang W, Zhou B, Shi J, Zhang C, Wang X (2018) Factorizable net: an efficient subgraph-based framework for scene graph generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp 335–351
26. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems. pp 5998–6008
27. Li R, Zhang S, He X (2022) Sgtr: End-to-end scene graph generation with transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp 19486–19496
28. Zhang A, Yao Y, Chen Q, Ji W, Liu Z, Sun M, Chua TS (2022) Fine-grained scene graph generation with data transfer. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII. Springer, pp 409–424
29. Suhail M, Mittal A, Siddiquie B, Broaddus C, Eledath J, Medioni G, Sigal L (2021) Energy-based learning for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 13936–13945
30. Yang J, Lu J, Lee S, Batra D, Parikh D (2018) Graph r-cnn for scene graph generation. In: Proceedings of the European conference on computer vision (ECCV). pp 670–685
31. Tian P, Mo H, Jiang L (2021) Scene graph generation by multi-level semantic tasks. Appl Intell 51(11):7781–7793
32. Chen T, Yu W, Chen R, Lin L (2019) Knowledge-embedded routing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 6163–6171
33. Han X, Song X, Dong X, Wei Y, Liu M, Nie L (2022) Dbiased-p: Dual-biased predicate predictor for unbiased scene graph generation. IEEE Trans Multimedia 1–11. https://doi.org/10.1109/TMM.2022.3190135
34. Tang K, Zhang H, Wu B, Luo W, Liu W (2019) Learning to compose dynamic tree structures for visual contexts. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 6619–6628
35. He T, Gao L, Song J, Li YF (2022) State-aware compositional learning toward unbiased training for scene graph generation. IEEE Trans Image Process 32:43–56
36. Wang Z, Xu X, Zhang Y, Yang Y, Shen HT (2022) Complex relation embedding for scene graph generation. IEEE Transactions on Neural Networks and Learning Systems 1–5. https://doi.org/10.1109/TNNLS.2022.3226871
37. Tursun O, Denman S, Sridharan S, Goan E, Fookes C (2022) An efficient framework for zero-shot sketch-based image retrieval. Pattern Recognition 126:108528
38. Wang J, Ge Y, Cai G, Yan R, Lin X, Shan Y, Qie X, Shou MZ (2022a) Object-aware video-language pre-training for retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 3313–3322
39. Wang AJ, Ge Y, Yan R, Ge Y, Lin X, Cai G, Wu J, Shan Y, Qie X, Shou MZ (2022b) All in one: Exploring unified video-language pre-training. arXiv preprint arXiv:2203.07303
40. Du Y, Wei F, Zhang Z, Shi M, Gao Y, Li G (2022) Learning to prompt for open-vocabulary object detection with vision-language

model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 14084–14093

41. Zareian A, Karaman S, Chang SF (2020) Bridging knowledge graphs to generate scene graphs. In: European Conference on Computer Vision. Springer, pp 606–623
42. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
43. Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li LJ, Shamma DA et al (2017) Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision 123(1):32–73
44. Xu D, Zhu Y, Choy CB, Fei-Fei L (2017) Scene graph generation by iterative message passing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 5410–5419
45. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems 91–99
46. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp 2961–2969
47. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp 1532–1543

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Xiaoguang Chang** received his bachelor's degree from Chang'an University in 2020. He is currently a Ph.D. candidate in the School of Cyber Science and Engineering at Southeast University. His research interests include computer vision, scene graph generation, and vision-language learning.

**Shaowei Cai** received the B.S. degree from Southeast University in 2018. He is pursuing his master's degree in the School of Automation at Southeast University. His research interests are in the field of computer vision with a specialization in scene graph generation.

**Teng Wang** received her bachelor's degree from Shandong University in 2009, M.S degree from the University of Science and Technology of China in 2012, and Ph.D. degree in computer engineering from Iowa State University, Ames, USA, in 2016. She is currently an assistant professor at Southeast University. Her research interests include pattern recognition, computer vision, environment sensing, visual navigation, and GPS-denied UAS navigation.

**Changyin Sun** received his bachelor's degree from the Department of Mathematics, Sichuan University, Chengdu, in 1996, and M.S. and Ph.D. degrees in electrical engineering from Southeast University in 2001 and 2003, respectively. He was a Post-Doctoral Fellow with the Department of Computer Science of Hong Kong in 2004. Since 2007, he has been a Professor at the School of Automation, Southeast University. From 2022, he serves as a member of the Standing Committee of the Party Committee and Vice President of Anhui University, Dean of the Future College, and Dean of the School of Artificial Intelligence. His current research interests include pattern recognition, machine vision, the theory and design of intelligent control systems, and deep reinforcement learning.