arXiv:2308.06712v3 [cs.CV] 17 Jan 2026

Compositional Feature Augmentation for Unbiased Scene Graph Generation

Lin Li[1,2], Guikun Chen[1], Jun Xiao[1], Yi Yang[1], Chunping Wang[3], Long Chen[2†]
[1]Zhejiang University [2]The Hong Kong University of Science and Technology [3]FinVolution
{mukti, guikun.chen, junx, yangyics}@zju.edu.cn, wangchunping02@xinye.com, longchen@ust.hk
https://github.com/HKUST-LongGroup/CFA

Abstract

Scene Graph Generation (SGG) aims to detect all the visual relation triplets <sub, pred, obj> in a given image. With the emergence of various advanced techniques for better utilizing both the intrinsic and extrinsic information in each relation triplet, SGG has achieved great progress over the recent years. However, due to the ubiquitous long-tailed predicate distributions, today's SGG models are still easily biased to the head predicates. Currently, the most prevalent debiasing solutions for SGG are re-balancing methods, e.g., *changing the distributions of original training samples. In this paper, we argue that all existing re-balancing strategies fail to increase the diversity of the relation triplet features of each predicate, which is critical for robust SGG. To this end, we propose a novel Compositional Feature Augmentation (**CFA**) strategy, which is the first unbiased SGG work to mitigate the bias issue from the perspective of increasing the diversity of triplet features. Specifically, we first decompose each relation triplet feature into two components: intrinsic feature and extrinsic feature, which correspond to the intrinsic characteristics and extrinsic contexts of a relation triplet, respectively. Then, we design two different feature augmentation modules to enrich the feature diversity of original relation triplets by replacing or mixing up either their intrinsic or extrinsic features from other samples. Due to its model-agnostic nature, CFA can be seamlessly incorporated into various SGG frameworks. Extensive ablations have shown that CFA achieves a new state-of-the-art performance on the trade-off between different metrics.*

1. Introduction

As one of the fundamental comprehensive visual scene understanding tasks, Scene Graph Generation (**SGG**) has attracted unprecedented interest from our community and has made great progress in recent years [29, 47, 44, 2, 4, 35, 24, 26, 21, 43, 23, 32]. Specifically, SGG aims to transform an image into a visually-grounded graph representation (*i.e.*, scene graph) where each node represents an object instance with a bounding box and each directed edge represents the corresponding predicate between the two objects. Thus, each scene graph can also be formulated as a set of visual relation triplets (*i.e.*, <sub, pred, obj>). Since such structural representations can provide strong explainable potentials, SGG has been widely-used in various downstream tasks, such as visual question answering, image retrieval, and captioning.

In general, due to the extremely diverse visual appearance of different visual relation triplets, recent SGG meth-

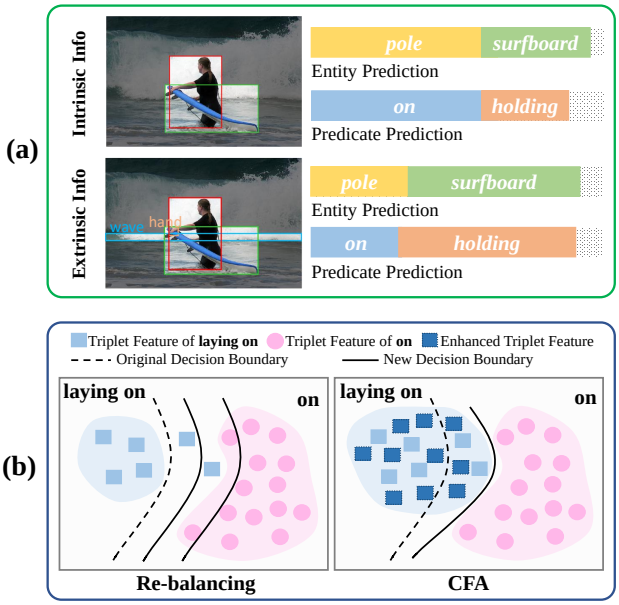


Figure 1: **(a)** The intrinsic and extrinsic information for SGG. The entity prediction is for the green box, and the predicate prediction is for the relation between the red and green boxes. **(b)** Illustration of the diversity of feature space and decision boundary between on and laying on before and after using re-balancing and CFA. Each sample denotes the corresponding visual triplet features.

† Corresponding author. Work was done when Lin Li visited HKUST.

ods all consider both ***intrinsic*** and ***extrinsic*** information for entity and predicate classification [44, 36, 30]. By "intrinsic information", we mean these intrinsic characteristics of the subjects and objects, such as their visual, semantic, and spatial features. For example in Figure 1(a), with the help of these intrinsic features, we can easily infer all possible entity categories (*e.g.*, `pole` or `surfboard`) and predicate categories (*e.g.*, `on` or `holding`) of each triplet. However, sometimes it is still hard to confirm the exact correct predictions with only intrinsic information, especially for tiny objects. Thus, it is also essential to consider other "extrinsic information" in the same image, such as the context features from neighbor objects. As shown in Figure 1(a), after encoding the features of surrounding objects (*e.g.*, `wave` and `hand`), we can easily infer that the categories of the entity and predicate should be `surfboard` and `holding`.

Although numerous advanced techniques have been proposed to effectively leverage both intrinsic and extrinsic information, today's SGG methods still fail to predict some informative predicates due to the ubiquitous long-tailed predicate distribution in prevalent SGG datasets [20]. Such a distribution is characterized by few categories with vast samples (head[1]) and many categories with rare samples (tail). Since the discrepancy of feature diversity and sample size among different categories, the learned decision boundary becomes improper (*c.f*. Figure 1(b)), *i.e.*, their predictions are biased towards the head predicates (*e.g.*, `on`) and they are error-prone for the tail ones (*e.g.*, `laying on`).

To overcome the bias issue, the most prevalent unbiased SGG solutions are re-balancing strategies, *e.g.*, sample resampling [26, 49] and loss re-weighting [41, 1, 27, 31, 17]. They alleviate the negative impact of long-tailed distribution by increasing samples or loss weights of tail classes. Then, the decision boundaries are adjusted to reduce the bias introduced by imbalanced distributions. However, we argue that all the existing re-balancing strategies fail to increase the diversity of relation triplet features[2] of each predicate, *i.e.*, they only change the frequencies or contributions of existing relation triplet features (*c.f*. Figure 1(b)). Since these tail categories are under-represented, it is still hard to infer the complete data distribution, *i.e.*, making it challenging to find the optimal direction to adjust the decision boundaries [6, 38]. For example in Figure 1(b), the feature space of `laying on` is so sparse that the decision boundary can be adjusted within a large range. The performance of this naive adjustment without "complete" distribution is always sensitive to hyperparameters, *i.e.*, excessively increasing the sample number or loss weight of tail predicates may cause some head predicate samples to be incorrectly predicted as tail classes (right solid line), and vice versa (left solid line).

[1]We directly use "tail", "body", and "head" categories to represent the predicate categories in the tail, body, and head parts of the number distributions of different predicates in SGG datasets, respectively.

[2]We use the "relation triplet feature" to represent the combination of both **intrinsic** and **extrinsic** features of each visual relation triplet.



(a) Intrinsic-CFA **(b) Extrinsic-CFA**

Figure 2: **(a) Intrinsic-CFA**: Replacing the entity feature of tail predicate triplet `dog-laying on-bed` from `dog` to `cat` to enhance the intrinsic feature. **(b) Extrnisc-CFA**: Mixing up the feature of tail predicate triplet `pillow-laying on-bed` into the context of `pillow-on-bed` to enhance the extrinsic feature.

In this paper, we propose a novel Compositional Feature Augmentation (CFA) strategy for unbiased SGG, which tries to solve the bias issue by enhancing the diversity of relation triplet features, especially for tail predicates. Specifically, CFA consists of two components: an intrinsic feature augmentation (*intrinsic-CFA*) and an extrinsic feature augmentation (*extrinsic-CFA*), which enhance the intrinsic and extrinsic features[2], respectively. As shown in Figure 1(b), by increasing the feature diversity of the tail predicates (*e.g.*, `laying on`), SGG models can easily learn the proper decision boundaries (vs. re-balancing strategies).

For intrinsic-CFA, we replace the entity features (*e.g.*, subject or object) of a tail predicate triplet with other "suitable" entity features. To determine the suitable entity categories for the augmentation, we propose a new hierarchical clustering method to find the correlations between different entity categories, and then we regard the entity features from the same cluster are suitable. Specifically, we calculate the category correlation by pattern, context, and semantic similarities. For example in Figure 2(a), the entity categories `cat` and `dog` are in the same cluster, *i.e.*, we can augment the intrinsic feature by replacing the `dog` entity feature with a `cat` entity feature. **For extrinsic-CFA**, we take advantage of the context or interactions of other triplets (*i.e.*, context triplets) and enhance the features of tail predicate triplets by these context triplets. Specifically, given a context triplet randomly selected from an image, we first select a reasonable tail predicate triplet as the target by limiting the categories and relative position of two objects. Then, to minimize the impact on the prediction of other triplets in the original image and make use of the extrinsic features of

the context triplet, we use a *mixup operation* to fuse the features of targeted tail predicate triplet into the context triplet. For example in Figure 2(b), triplet `pillow-laying on-bed` is mixed up into the image of `pillow-on-bed`.

We evaluate CFA on two most prevalent and challenging SGG datasets: Visual Genome (**VG**) [20] and **GQA** [16]. Since CFA is a model-agnostic debiasing strategy, it can be seamlessly incorporated into various SGG architectures[3] and consistently improve their performance. Unsurprisingly, CFA can achieve a new state-of-the-art performance on the trade-off between different metrics. Extensive ablations and results on multiple SGG tasks and backbones have shown the generalization ability and effectiveness of CFA.

In summary, we make three contributions in this paper:

1. We reveal the issue of existing re-balancing methods, *i.e.*, the lack of triplet feature diversity of tail categories. To this end, we are the first to tackle unbiased SGG from the perspective of increasing the diversity of triplet features.
2. We propose the model-agnostic CFA for unbiased SGG, which is an efficient and novel compositional learning framework that spans the feature space of the tail categories by two independent plug-and-play modules.
3. Extensive results show the effectiveness of CFA, *i.e.*, it achieves a new SOTA performance on SGG benchmarks.

2. Related Work

Unbiased Scene Graph Generation. Biased predictions prevent further use of scene graphs in real-world applications. Recent unbiased SGG works can be roughly divided into three main categories: 1) *Re-balancing*: It alleviates the negative impact of long-tailed predicate distribution by re-weighting or re-sampling [41, 27, 31, 26, 8]. 2) *Unbiased Inference*: It makes unbiased predictions based on biased models [35, 42]. 3) *Noisy Label Learning*: It reformulates SGG as a noisy label learning problem and corrects these noisy samples [21, 22]. In this work, we point out the drawbacks of existing re-balancing methods and study unbiased SGG from the new perspective of feature augmentation.

Feature Augmentation. Data augmentation is a prevalent training trick to improve models' performance. Conventional data augmentation methods [9, 48] usually synthesize new samples in a hand-crafted manner. Compared to these image-level data augmentation methods, feature augmentation is another efficient way to improve models' generalizability by directly synthesizing samples in the feature space [6, 25]. Compared to existing data augmentation applications, SGG is a sophisticated task that involves intrinsic and extrinsic features. In this work, we propose CFA to enrich the diversity of relation triplet features for debiasing.

[3]Following the mainstream and concurrent unbiased SGG works, we also only focus on two-stage frameworks. As for one-stage models, a similar idea can be applied at the image-level, and we leave it for future works.

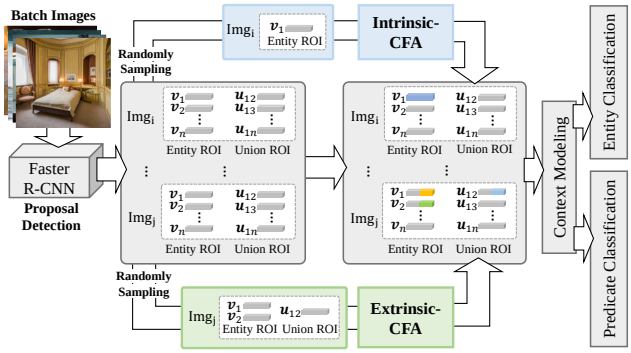


Figure 3: The illustration of unbiased SGG framework with CFA.

Compositional Learning (CL). CL has been successfully applied to various computer vision tasks. As for visual scene understanding, some Human Object Interaction detection works [14, 18, 15] compose new interaction samples that significantly benefit both low-shot and zero-shot settings. To the best of our knowledge, only two unpublished SGG work [13, 19] also uses CL. Compared to [13], we have several key differences: 1) They aim to generate new representations which are close to the original triplet. Instead, we try to increase the diversity of triplet features. 2) They only change the entities and ignore the extrinsic features. 3) Their augmentation strategies are mainly based on the spatial locations or IoU of the entities. The second work [19] primarily addresses classification tasks. Our work differs from it in two aspects: 1) CFA increases the diversity of tail predicate features by leveraging both intrinsic and extrinsic information. 2) we aim to improve robustness and performance under long-tailed predicate distributions.

3. Approach

Given an image I, a scene graph is formally represented as $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N}$ and $\mathcal{E}$ denote the set of all objects and their pairwise visual relations, respectively. Specifically, the i-th object in $\mathcal{N}$ consists of a bounding box (bbox) $\boldsymbol{b}_i \in \mathcal{B}$ and its entity category $o_i \in \mathcal{O}$. A relation $r_{ij} \in \mathcal{R}$ denotes the predicate category between i-th object and j-th object. $\boldsymbol{b}_{ij}$ denotes the union box of bbox $\boldsymbol{b}_i$ and bbox $\boldsymbol{b}_j$. $\mathcal{B}$, $\mathcal{O}$, and $\mathcal{R}$ represent the set of all entity bboxes, entity categories, and predicate categories, respectively.

In this section, we first revisit the two-stage SGG baselines in Sec. 3.1. Then, we detailedly introduce our CFA in Sec. 3.2, including intrinsic-/extrinsic-CFA (*c.f.* Figure 3). Finally, we introduce the training objectives in Sec. 3.3.

3.1. Revisiting the Two-Stage SGG Baselines

Since the mainstream SGG methods are two-stage models, we review the two-stage SGG framework[3] here [44, 36].

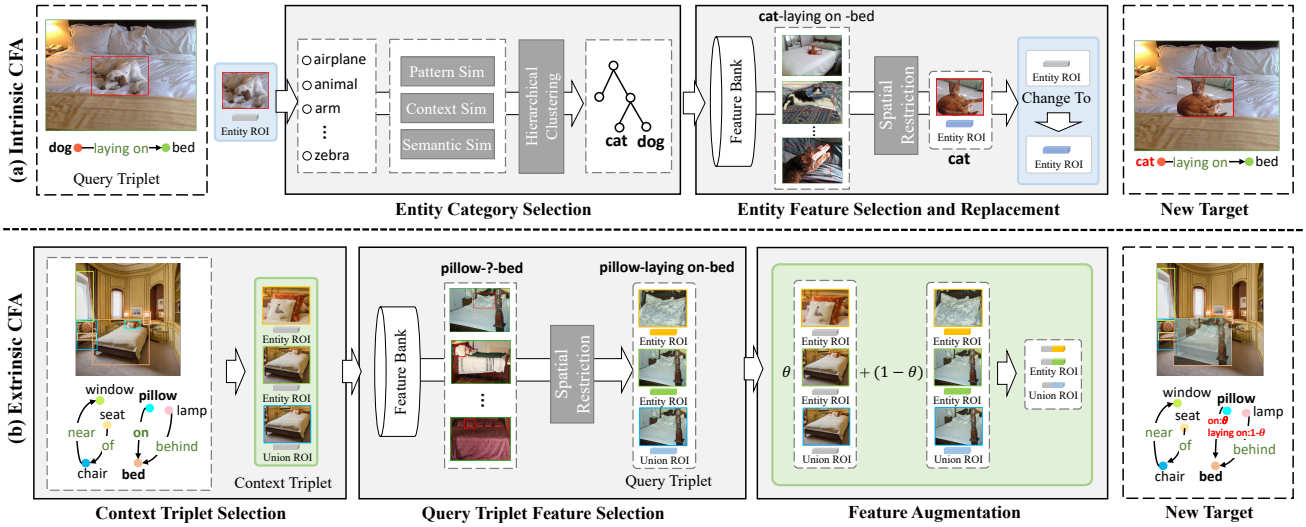


Figure 4: The pipeline of Intrinsic CFA (a) and Extrinsic CFA (b). The blue and green boxes represent operations on the query triplet features in Intrinsic-CFA and context triplet features in Extrinsic-CFA, respectively.

A typical two-stage SGG model involves three steps: proposal generation, entity classification, and predicate classification. Thus, the SGG task $P(\mathcal{G}|I)$ is decomposed into:

$$P(\mathcal{G}|I) = P(\mathcal{B}|I)P(\mathcal{O}|\mathcal{B},I)P(R|\mathcal{O},\mathcal{B},I). \quad (1)$$

Proposal Generation $P(\mathcal{B}|I)$**.** This step aims to generate all bbox proposals $\mathcal{B}$. Given an image I, they first utilize an off-the-shelf object detector (*e.g.*, Faster R-CNN [34]) to detect all the proposals $\mathcal{B}$ and their visual features $\{\boldsymbol{v}_i\}$.

Entity Classification $P(\mathcal{O}|\mathcal{B},I)$**.** This step mainly predicts the entity category of each $\boldsymbol{b}_i \in \mathcal{B}$. Given a visual feature $\boldsymbol{v}_i$ and proposal $\boldsymbol{b}_i$, they use an object context encoder $\texttt{Enc}_{obj}$ to extract the contextual entity representation $\boldsymbol{f}_i$:

$$\boldsymbol{f}_i = \texttt{Enc}_{obj}(\boldsymbol{v}_i \oplus \boldsymbol{b}_i), \quad (2)$$

where $\oplus$ denotes concatenation. Then, they use an object classifier $\texttt{Cls}_{obj}$ to predict their entity categories:

$$\widehat{o}_i = \texttt{Cls}_{obj}(\boldsymbol{f}_i). \quad (3)$$

Predicate Classification $P(\mathcal{R}|\mathcal{O},\mathcal{B},I)$**.** This step predicts the predicate categories of every two proposals in $\mathcal{B}$ along with their entity categories. First, they use a relation context encoder $\texttt{Enc}_{rel}$ to extract the refined entity feature $\tilde{\boldsymbol{f}}_i$:

$$\tilde{\boldsymbol{f}}_i = \texttt{Enc}_{rel}(\boldsymbol{v}_i \oplus \boldsymbol{f}_i \oplus \boldsymbol{w}_i), \quad (4)$$

where $\boldsymbol{w}_i$ is the GloVe embedding [33] of predicted $\widehat{o}_i$. It is worth noting that both $\texttt{Enc}_{cls}$ and $\texttt{Enc}_{rel}$ often adopt a sequence model (*e.g.*, Bi-LSTM [44], Tree-LSTM [36], or Transformer [42]) to better capture context. After relation feature encoding, they use a relation classifier $\texttt{Cls}_{rel}$ to predict the relation $\widehat{r}_{ij}$ between any two subject-object pairs:

$$\widehat{r}_{ij} = \texttt{Cls}_{rel}([\tilde{\boldsymbol{f}}_i \oplus \tilde{\boldsymbol{f}}_j] \circ \boldsymbol{u}_{ij}), \quad (5)$$

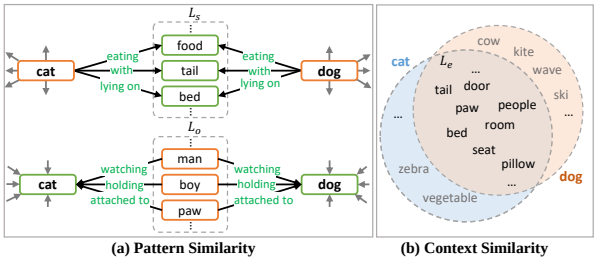
where $\circ$ denotes element-wise product, and $\boldsymbol{u}_{ij}$ denotes the visual feature of the union box $\boldsymbol{b}_{ij}$.

3.2. CFA: Compositional Feature Augmentation

In this paper, we treat a relation triplet feature[2] in SGG as a combination of two components: intrinsic feature and extrinsic feature, which refer to the intrinsic information (*i.e.*, the subject and object itself) and extrinsic information (*i.e.*, the contextual objects and stuff), respectively. Correspondingly, our CFA consists of intrinsic-CFA and extrinsic-CFA to augment these two types of features. For ease of presentation, we call the "targeted tail predicate triplet whose feature to be enhanced" as "**query triplet**", and the "relation triplet whose image is used to provide context" as "**context triplet**". Besides, to facilitate the feature augmentation, we store all the visual features of all tail predicate triplets (*i.e.*, visual features $\boldsymbol{v}$ of two entities and their union feature $\boldsymbol{u}$) in a **feature bank** before training. Besides, we adopt the repeat factor $\eta = \max(1, \eta_r)$ [12, 26] to sample images to provide enough context triplet and query triplet for augmentation, where $\eta_r = \sqrt{\lambda / f_r}$, f_r is the frequency of predicate category r on the entire dataset, and λ is hyperparameter. The unbiased SGG pipeline with CFA is shown in Figure 3.

3.2.1 Intrinsic-CFA

Intrinsic-CFA enhances the feature of query triplet by replacing its visual features of entities (*i.e.*, subject and ob-



(a) Pattern Similarity (b) Context Similarity

Figure 5: Illustration of the pattern and context similarity between entity categories `cat` and `dog`. a) White boxes are the behavior patterns common to two categories. b) Blue and red circles are entity categories that co-occur with `cat` and `dog`, respectively.

ject). It consists of two steps: *entity category selection* and *entity feature selection and replacement* (*c.f*. Figure 4(a)). During training, we randomly select a query triplet from a batch of images, and randomly select one of its entity features as input. Then we put it into the Intrinsic-CFA module (*c.f*. Figure 3). Next, we detailedly introduce each step.

Entity Category Selection. This step is used for determining the category to which the entity feature of query triplet is replaced. Firstly, we propose a novel **hierarchical clustering** strategy to mine potentially fungible entity categories. Then we randomly select an entity category from the same cluster for the next entity feature selection. For the query triplet `dog-laying on-bed` in Figure 4, the category `cat` is selected from the same cluster as `dog`. The categories in the same cluster are common in the behavior patterns (*e.g*., they can be ridden), contexts (*e.g*., they can appear in a scene with a street) as well as semantics. Accordingly, three kinds of similarity are used to measure the common characters of two entity categories: pattern similarity, context similarity, and semantic similarity.

(i) Pattern Similarity. It measures the overlap of the behavior patterns of two entity categories [39]. Figure 5(a) visualizes the common pattern between `cat` and `dog` (*e.g*., they all can be attached by the `paw`). Pattern similarity between two entity categories c_i and c_j is defined as:

$$Sim_p(c_i, c_j) = \frac{|L_s|}{d_{out}(c_i) + d_{out}(c_j) - |L_s|} + \frac{|L_o|}{d_{in}(c_i) + d_{in}(c_j) - |L_o|}, \tag{6}$$

where $|L_s|$ ($|L_o|$) is the number of common `pred-obj` (`sub-pred`) classes of the triplet whose `sub` (`obj`) class is c_i or c_j. $d_{in}(c_i)$ ($d_{out}(c_i)$) is the number of incoming (outgoing) edges of the entity with class c_i in the whole dataset.

(ii) Context Similarity. It measures the overlap of other entity categories in images of the two entity categories. Both `cat` and `dog` can appear in the same scene with entity categories, *e.g*., `room` in Figure 5(b). It is defined as:

$$Sim_c(c_i, c_j) = \frac{|L_e|}{d_{co}(c_i) + d_{co}(c_j) - |L_e|}, \tag{7}$$

where $|L_e|$ denotes the number of entity category intersections that appear in the same image with entity category c_i or with c_j. $d_{co}(c_i)$ is the number of entity instances in the whole dataset that appear in the same image as category c_i.

(iii) Semantic Similarity. It is the cosine similarity between two entity categories on semantic embedding [33] space:

$$Sim_s(c_i, c_j) = \frac{\boldsymbol{w}_i \cdot \boldsymbol{w}_j}{|\boldsymbol{w}_i||\boldsymbol{w}j|}, \tag{8}$$

The final similarity function $Sim(c_i, c_j)$ used in clustering is the weighted sum of above three similarities. Complete clustering algorithm and results are in the **appendix**.

Entity Feature Selection and Replacement. This step selects an entity feature from the feature bank based on the selected entity category and replaces it with the query triplet. Specifically, we first update a new entity category of query triplet (*e.g*., `cat-laying on-bed`). Then, we select all triplet features with the same category as query triplet from feature bank as candidates (*c.f*. Figure 4(a)). And we utilize spatial restriction to filter unreasonable triplet features.

Spatial Restriction. In real augmentation, even if the two entity categories appear to be interchangeable, the positions of the subject and object after replacement may not make sense. For example, in VG dataset (*c.f*. **appendix**), though category `hat` and `shoe` are in the same cluster, directly replacing `hat` with `shoe` would place the `shoe` on the `head` and make it contrary to common sense.

To further ensure the rationality of the replacement of the two entity features, we use the cosine similarity of the relative spatial position of subject-object of the query triplet and the selected triplet from the feature bank as the restriction:

$$Sim_d(\boldsymbol{p}_i, \boldsymbol{p}_j) = \frac{\boldsymbol{p}_i \cdot \boldsymbol{p}_j}{|\boldsymbol{p}_i||\boldsymbol{p}j|}, \tag{9}$$

where $\boldsymbol{p}_i$ is the spatial vector from the center of the subject bbox to the center of object bbox. When the Sim_d between $\boldsymbol{p}_i$ of the query triplet and $\boldsymbol{p}_j$ of the selected triplet from the feature bank is larger than the threshold σ, the entity feature of the selected triplet is reasonable for replacement.

During replacement, we randomly select one of the reasonable entity features and replace it into query triplet and change the original entity target to the new entity category.

3.2.2 Extrinsic-CFA

Extrinsic-CFA aims to enrich the feature of the query triplet through the extrinsic information (*i.e*., the context formed by all the entities of an image) of context triplets. Different from the Intrinsic-CFA, we first select context triplet during

training for computation efficiency. Extrinsic-CFA includes three steps: context triplet selection, query triplet feature selection, and feature augmentation (*c.f*. Figure 4(b)). Then, we will detailedly introduce each step of Extrinsic-CFA.

Context Triplet Selection. For each image, this step selects context triplets for extrinsic-CFA. Specifically, we randomly select context triplets with foreground or background predicate categories to provide more extrinsic information. *For foreground context triplets*, considering the large variation in the number of triplet samples for different predicate categories, we randomly sample the context triplets by using probability $p = (\eta - \eta_r)/\eta \times \gamma$ during training, where γ is hyperparameter. *For background context triplets*, we randomly select them with the same subject-object pair as the tail predicate triplet in the whole dataset.

Query Triplet Feature Selection. This step selects a query triplet with its features for the feature augmentation. Specifically, we select triplets with the same subject-object categories (*e.g*., `pillow-bed`) as the context triplet (*e.g*., `pillow-on-bed`) from feature bank as the candidate query triplets. Then, we implement the same spatial restriction as intrinsic-CFA to filter unreasonable triplets. Finally, we randomly select one of the reasonable triplets as the query triplet (*e.g*., `pillow-laying on-bed`), and its features are utilized for augmentation (*c.f*. Figure 4(b)).

Feature Augmentation. To reduce the impact on other triplets of the original image and enhance the extrinsic features of the tail predicate triplets, we perform mixup operation [46] between the selected query triplet (*e.g*., `pillow-laying on-bed`) and context triplet (*e.g*., `pillow-on-bed`), *c.f*. Figure 4(b). The mixup operation is written as:

$$\tilde{\boldsymbol{v}}_s = \theta \boldsymbol{v}_s + (1-\theta)\boldsymbol{v}'_s, \tag{10}$$

$$\tilde{\boldsymbol{v}}_o = \theta \boldsymbol{v}_o + (1-\theta)\boldsymbol{v}'_o, \tag{11}$$

$$\tilde{\boldsymbol{u}} = \theta \boldsymbol{u} + (1-\theta)\boldsymbol{u}', \tag{12}$$

where $\boldsymbol{v}'_s$ ($\boldsymbol{v}_s$) and $\boldsymbol{v}'_o$ ($\boldsymbol{v}_o$) denote the visual features of the subject and object of the query triplet (context triplet), respectively. The $\boldsymbol{u}'$ ($\boldsymbol{u}$) denotes the visual feature of the uion box of the query triplet (context triplet). Similarly, predicate target ground-truth r of the context triplet is mixed:

$$\tilde{r} = \theta r + (1-\theta)r', \tag{13}$$

where θ is utilized to control the degree of mixup operation.

After the mixup operation, the query triplet feature can be enhanced in context modeling (*c.f*. Figure 3) by leveraging the extrinsic features of the context triplet.

3.3. Training Objectives

Cross-Entropy Losses. The optimization objective of commonly used SGG models mainly includes the cross-entropy of entity and relation classification, the loss functions are:

$$L_{obj} = \sum\nolimits_i \texttt{XE}(\widehat{o}_i, o_i),\ L_{rel} = \sum\nolimits_{ij} \texttt{XE}(\widehat{r}_{ij}, r_{ij}), \tag{14}$$

where $\widehat{o}_i$ is the predicted entity category and o_i is the ground-truth entity category. $\widehat{r}_{ij}$ is the predicted predicate category and r_{ij} is the ground-truth predicate category.

Contrastive Loss. Due to the pattern changes between the entity features after mixup and original entity features, using only XE losses may result in performance drops in entity classification. To maintain the discriminative entity features after mixup, we further apply a contrastive loss [3]:

$$L_{cl} = -\log \frac{\exp(Sim_e(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2M} 1_{[k\neq i]} \exp(Sim_e(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \tag{15}$$

where $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ represent the output of the layer before predictor of original entity and the entity after mixup operation, M is the number of entities in which the mixup operation is performed. $Sim_e(\cdot,\cdot)$ is the cosine similarity.

Training. During training, the total loss includes the cross-entropy losses L_{obj}, L_{rel} and the extra contrastive loss L_{cl}:

$$L_{total} = L_{rel} + L_{obj} + \beta L_{cl}, \tag{16}$$

where β is used to regulate the magnitude of loss.

4. Experiments

4.1. Experimental Settings and Details

Tasks. We evaluated models in three tasks [40]: 1) *Predicate Classification* (**PredCls**): Predicting the predicate category given all ground-truth entity bboxes and categories. 2) *Scene Graph Classification* (**SGCls**): Predicting categories of the predicate and entity given all ground-truth entity bboxes. 3) *Scene Graph Generation* (**SGGen**): Detecting all entities and their pairwise predicates.

Metrics. We evaluated SGG models on three metrics: 1) *Recall@K* (**R@K**): It indicates the proportion of ground-truths that appear among the top-K confident predicted relation triplets. 2) *mean Recall@K* (**mR@K**): It is the average of R@K scores which are calculated for each predicate category separately. 3) **Mean**: It is the average of all R@K and mR@K scores. Since R@K favors head predicates while mR@K favors tail predicates, the Mean can better reflect the overall performance of all predicates [21].

Datasets and Implementation Details. We conducted all experiments on two datasets: **VG** [20] and **GQA** [16]. More details of datasets and implementation are in the appendix.

4.2. Comparison with State-of-the-Arts

Setting. Due to the model-agnostic nature, we equipped our CFA with three strong two-stage SGG baselines: Motifs [44], VCTree [36] and Transformer [37], and they

SGG Models	PredCls					SGCls					SGGen				
	mR@K		R@K		Mean	mR@K		R@K		Mean	mR@K		R@K		Mean
	50	100	50	100		50	100	50	100		50	100	50	100	
Motifs [44]$_{CVPR'18}$	16.5	17.8	65.5	67.2	41.8	8.7	9.3	39.0	39.7	24.2	5.5	6.8	32.1	36.9	20.3
VCTree [36]$_{CVPR'19}$	17.1	18.4	65.9	67.5	42.2	10.8	11.5	45.6	46.5	28.6	7.2	8.4	32.0	36.2	20.9
Transformer [37]$_{NIPS'17}$	17.9	19.6	63.6	65.7	41.7	9.9	10.5	38.1	39.2	24.4	7.4	8.8	30.0	34.3	20.1
BGNN [26]$_{CVPR'21}$	30.4	32.9	59.2	61.3	45.9	14.3	16.5	37.4	38.5	26.7	10.7	12.6	31.0	35.8	22.5
Motifs+PCPL [41]$_{ACMMM'20}$	24.3	26.1	54.7	56.5	40.4	12.0	12.7	35.3	36.1	24.0	10.7	12.6	27.8	31.7	20.7
Motifs+DLFE [5]$_{ACMMM'21}$	26.9	28.8	52.5	54.2	40.6	15.2	15.9	32.3	33.1	24.1	11.7	13.8	25.4	29.4	20.1
Motifs+BPL-SA [11]$_{ICCV'21}$	29.7	31.7	50.7	52.5	41.2	16.5	17.5	30.1	31.0	23.8	13.5	15.6	23.0	26.9	19.8
Motifs+NICE [21]$_{CVPR'22}$	29.9	32.3	55.1	57.2	43.6	16.6	17.9	33.1	34.0	25.4	12.2	14.4	27.8	31.8	21.6
Motifs+IETrans [45]$_{ECCV'22}$	30.9	33.6	54.7	56.7	44.0	16.8	17.9	32.5	33.4	25.2	12.4	14.9	26.4	30.6	21.1
Motifs+CFA (ours)	35.7	38.2	54.1	56.6	**46.2**	17.0	18.4	34.9	36.1	**26.6**	13.2	15.5	27.4	31.8	**22.0**
VCTree+PCPL [41]$_{ACMMM'20}$	22.8	24.5	56.9	58.7	40.7	15.2	16.1	40.6	41.7	28.4	10.8	12.6	26.6	30.3	20.1
VCTree+DLFE [5]$_{ACMMM'21}$	25.3	27.1	51.8	53.5	39.4	18.9	20.0	33.5	34.6	26.8	11.8	13.8	22.7	26.3	18.7
VCTree+BPL-SA [11]$_{ICCV'21}$	30.6	32.6	50.0	51.8	41.3	20.1	21.2	34.0	35.0	27.6	13.5	15.7	21.7	25.5	19.1
VCTree+NICE [21]$_{CVPR'22}$	30.7	33.0	55.0	56.9	43.9	19.9	21.3	37.8	39.0	29.5	11.9	14.1	27.0	30.8	21.0
VCTree+IETrans [45]$_{ECCV'22}$	30.3	33.9	53.0	55.0	43.1	16.5	18.1	32.9	33.8	25.3	11.5	14.0	25.4	29.3	20.1
VCTree+CFA (ours)	34.5	37.2	54.7	57.5	**46.0**	19.1	20.8	42.4	43.5	**31.5**	13.1	15.5	27.1	31.2	**21.7**
Transformer+IETrans [45]$_{ECCV'22}$	30.8	34.5	51.8	53.8	42.7	17.4	19.1	32.6	33.5	25.7	12.5	15.0	25.5	29.6	20.7
Transformer+CFA (ours)	30.1	33.7	59.2	61.5	**46.1**	15.7	17.2	36.3	37.3	**26.6**	12.3	14.6	27.7	32.1	**21.7**

Table 1: Performance (%) of the SOTA trade-off SGG models on VG [20]. "Mean" is the average of mR@50/100 and R@50/100.

SGG Models	PredCls			SGCls			SGGen		
	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100
Motifs+TDE [35]$_{CVPR'20}$	18.5	25.5	29.1	9.8	13.1	14.9	5.8	8.2	9.8
Motifs+CogTree [42]$_{IJCAI'21}$	20.9	26.4	29.0	12.1	14.9	16.1	7.9	10.4	11.8
Motifs+RTPB [1]$_{AAAI'22}$	28.8	35.3	37.7	16.3	20.0	21.0	9.7	13.1	15.5
Motifs+PPDL [27]$_{CVPR'22}$	27.9	32.2	33.3	15.8	17.5	18.2	9.2	11.4	13.5
Motifs+GCL [10]$_{CVPR'22}$	30.5	36.1	38.2	**18.0**	20.8	21.8	**12.9**	**16.8**	**19.3**
Motif+HML [7]$_{ECCV'22}$	30.1	36.3	38.7	17.1	20.8	22.1	10.8	14.6	17.3
Motifs+CFA‡ (ours)	**31.5**	**39.9**	**43.0**	17.3	**20.9**	**22.4**	11.2	15.3	18.1
VCTree+TDE [35]$_{CVPR'20}$	18.4	25.4	28.7	8.9	12.2	14.0	6.9	9.3	11.1
VCTree+CogTree [42]$_{IJCAI'21}$	22.0	27.6	29.7	15.4	18.8	19.9	7.8	10.4	12.1
VCTree+RTPB [1]$_{AAAI'22}$	27.3	33.4	35.6	20.6	24.5	25.8	9.6	12.8	15.1
VCTree+PPDL [27]$_{CVPR'22}$	29.7	33.3	33.8	20.3	21.8	22.4	9.1	11.3	13.3
VCTree+GCL [10]$_{CVPR'22}$	31.4	37.1	39.1	19.5	22.5	23.5	**11.9**	**15.2**	17.5
VCTree+HML [7]$_{ECCV'22}$	31.0	36.9	39.2	20.5	25.0	26.8	10.1	13.7	16.3
VCTree+CFA‡ (ours)	**31.6**	**39.2**	**42.5**	**21.5**	**26.3**	**28.3**	10.8	15.1	**17.9**
Transformer+CogTree [42]$_{IJCAI'21}$	22.9	28.4	31.0	13.0	15.7	16.7	7.9	11.1	12.7
Transformer+HML [7]$_{ECCV'22}$	27.4	33.3	35.9	15.7	19.1	20.4	**11.4**	15.0	17.7
Transformer+CFA‡ (ours)	**31.2**	**38.6**	**41.5**	**17.2**	**20.9**	**22.7**	10.6	**15.0**	**17.9**

Table 2: Performance (%) of the SOTA tail-focused SGG models on VG [20]. ‡ means using the component prior knowledge.

Models	PredCls	SGCls	SGGen
	mR@50/100	mR@50/100	mR@50/100
Motifs [44]	13.9 / 14.7	7.2 / 7.5	5.5 / 6.6
+CFA	**31.7 / 33.8**	**14.2 / 15.2**	**11.6 / 13.2**
VCTree [36]	14.4 / 15.3	6.1 / 6.6	5.8 / 6.0
+CFA	**33.4 / 35.1**	**14.1 / 15.0**	**10.8 / 12.6**
Transformer [37]	15.2 /16.1	7.5 / 7.9	6.9 / 7.8
+CFA	**27.8 / 29.4**	**16.2 / 16.9**	**13.4 / 15.3**

Table 3: Performance (%) of the SGG models on GQA [16].

are denoted as **Motifs+CFA**, **VCTree+CFA**, and **Transformer+CFA**, respectively. In addition, due to the limited diversity of tail predicate components [28], it has a high correlation with the category of subject & object. Thus, we further equip our three models with the component prior knowledge collected from the dataset to further improve the performance of the tail predicates. And they are denoted as **Motifs+CFA**‡, **VCTree+CFA**‡, and **Transformer+CFA**‡ (More details about the priors are discussed in appendix).

Baselines. We compared our methods with the SOTA models in the VG dataset (Table 1, Table 2) and the GQA dataset (Table 3). Specifically, these models can be divided into three groups: 1) **Model-specific designs**: Motifs, VCTree, Transformer, and BGNN [26]. 2) **Model-agnostic trade-off** methods: consider the performance of all predicates comprehensively (*i.e.*, higher Mean), *e.g.*, PCPL [41], DLFE [5], BP-LSA [11], NICE [21], and IETrans [45]. 3) **Model-agnostic tail-focused** methods: improve tail predi-

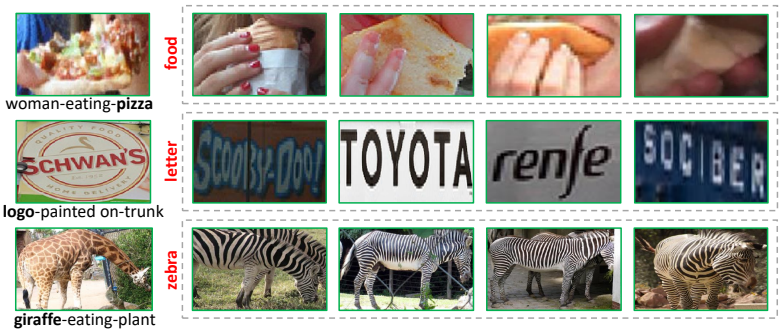


Figure 6: The examples of reasonable entities for the query triplet in intrinsic CFA. The lefts are original entities of the query triplets. The rights are some alternate entity categories (red) and their samples (gray boxes) for each query triplet to replace.

Component			PredCls		
IN	EX-fg	EX-bg	mR@50 / 100	R@50 / 100	Mean
			16.5 / 17.8	65.6 / 67.2	41.8
✓			19.3 / 21.2	64.7 / 66.6	43.0
	✓		25.6 / 27.8	63.0 / 64.8	45.3
		✓	23.9 / 26.3	63.3 / 65.7	44.8
✓	✓		27.2 / 29.3	61.9 / 64.3	45.7
✓		✓	27.5 / 30.0	61.7 / 64.1	45.8
	✓	✓	27.8 / 30.3	60.7 / 63.4	45.6
✓	✓	✓	**35.7 / 38.2**	54.1 / 56.6	**46.2**

Table 4: Ablation study on each component on VG [20]. IN: Replace the intrinsic features. Ex-fg: Mix up the extrinsic features of foreground triplets. Ex-bg: Mix up the extrinsic features of background triplets.

cates performance at the expense of excessively sacrificing the head (*i.e.*, higher mR@K and R@50 less than 50.0% on PredCls), *e.g.*, TDE [35], CogTree [42], RTPB [1], PPDL [27], GCL [10], and HML [7]. For a fair comparison, we compared with the methods in the last two groups.

Quantitative Results on VG. From the results of **trade-off** methods in Table 1, we can observe that: 1) Compared to the three strong baselines (*i.e.*, Motifs, VCTree and Transformer), CFA can significantly improve model performance on mR@K metric over all three settings. 2) CFA can achieve the best trade-off between R@K and mR@K, *i.e.*, highest Mean, and surpass the SOTA trade-off method NICE [21] in Mean metric under all settings. CFA shows minimal performance degradation on the head predicates (*c.f.* R@K) while maintaining the performance of the tail (*c.f.* mR@K), demonstrating the superiority of CFA considering all predicates. From the results of **tail-focused** methods in Table 2, we can observe that: after further implementing strategies to improve tail performance, CFA‡ can achieve the highest mR@K and exceed the SOTA tail-focused method HML [7] on mR@K metric under all settings. More experiment analysis is in the appendix.

Quantitative Results on GQA. From the results of Table 3, we can observe that: CFA can also greatly improve the mR@K of three strong baselines (*i.e.*, Motif, VCTree and Transformer) on the large dataset GQA, which proves the universality and effectiveness of our method.

4.3. Ablation Studies

Effectiveness of Each Component. We evaluated the importance of each component of CFA based on Motifs [44] under the PredCls setting. There are three components of CFA: replace the intrinsic features (**IN**), mix up the extrinsic features of the foreground triplets (**EX-fg**), and mix up the extrinsic features of the background triplets (**EX-bg**). As reported in Table 4, we have the following observations:

Similarity			PredCls		
Pattern	Context	Semantic	mR@50 / 100	R@50 / 100	Mean
✓			33.4 / 36.2	52.0 / 54.0	43.9
	✓		34.6 / 37.3	55.2 / 57.0	46.0
		✓	35.3 / 38.0	51.4 / 54.1	44.7
✓	✓		35.3 / 37.8	52.6 / 55.2	45.2
✓		✓	35.2 / 37.9	54.0 / 56.2	45.8
	✓	✓	34.5 / 36.9	55.2 / 57.5	46.0
✓	✓	✓	**35.7 / 38.2**	54.1 / 56.6	**46.2**

(a) Ablation study on each similarity in clustering of intrinsic CFA.

K	PredCls		
	mR@50 / 100	R@50 / 100	Mean
15	**35.7 / 38.2**	54.1 / 56.6	**46.2**
40	33.5 / 36.0	55.3 / 57.8	45.7
150	33.2 / 35.7	55.7 / 57.9	45.6

(b) Ablation study on the number of clusters K of intrinsic CFA.

θ	PredCls		
	mR@50 / 100	R@50 / 100	Mean
0.0	27.1 / 29.5	62.2 / 64.3	45.8
0.3	28.5 / 31.0	60.8 / 63.3	45.9
0.5	**35.7 / 38.2**	54.1 / 56.6	**46.2**
0.7	32.9 / 35.2	50.4 / 53.0	42.9
1.0	19.3 / 21.2	64.7 / 66.6	43.0

(c) Ablation study on mixup parameter θ of extrinsic CFA.

Table 5: Ablation studies on the different hyperparameters of each component of CFA. Motifs [44] is used in all ablation on VG [20].

1) Using only IN can slightly improve mR@K (*e.g.*, 2.8% $\sim$ 3.4% gains) and slightly hurts R@K (0.6% $\sim$ % 0.9 loss). The reason is that there are not enough tail predicate triplets, resulting in the feature diversity still being limited after replacing the intrinsic features. 2) Both Ex-fg and Ex-bg can significantly improve mR@K and keep competitive R@K compared to baseline (*e.g.*, 17.8% *vs.* 30.3% in mR@100, and 67.2% *vs.* 63.4% in R@100). 3) Combining all components allows for the best trade-off, *i.e.*, the highest Mean.

Similarity of Clustering in Intrinsic CFA. We analyzed the influence of three kinds of similarity (pattern, context, and semantic similarities) under the PredCls setting with baseline model Motifs [44]. From Table 5(a), we can ob-

serve that using only pattern similarity shows the worst performance, since it ignores the contextual and semantic information which gives crucial guidance for selection. For example, both `boat` and `car` can be ridden, but we cannot replace `boat` with `car` because `car` can't appear in the sea. Once the other two similarities are fused, the SGG performance becomes robust and the highest Mean is achieved.

Different Clusters K of Clustering in Intrinsic CFA. We performed $K \in \{15, 40, 150\}$ to evaluate the impact of the number of clusters under PredCls setting with Motifs [44]. All results are reported in Table 5(b). We find that K=15 works best in mR@K. The reason may be that when K=15, there is a larger selection range of replaceable entity categories and richer feature diversity of tail predicate triplets.

Mixup Parameter θ in Extrinsic CFA. As mentioned in Sec. 3.2.2, θ indicates the proportion of selected features in final augmented features. We investigated $\theta \in \{0.0, 0.3, 0.5, 0.7, 1.0\}$ under the PredCls setting with Motifs [44] in Table 5(c). When θ is too small, the query triplet has too much impact on the other triplets in the image of the context triplet, and when θ is too large, feature augmentation of the query triplet is not strong enough. To better trade-off the performance on different predicates, we set θ to 0.5.

Visualization of Reasonable Entities in Intrinsic CFA. Figure 6 shows some alternative entities for query triplets. The visual features of the candidate entities are quite different from the original entity, but are still reasonable for the query triplet, *e.g.*, the samples of `zebra` differ greatly in color and texture from the `giraffe`, but they can all eat plants. Replacing the `giraffe` with the `zebra` can provide new features to enrich the feature diversity of `eating`.

5. Conclusion and Future Work

In this paper, we revealed the drawbacks of existing rebalancing methods and discovered that the key challenge for unbiased SGG is to learn proper decision boundaries under the severe long-tailed predicate distribution. Thus, we proposed a model-agnostic CFA framework that can enrich the feature space of tail categories by augmenting both intrinsic and extrinsic features of the relation triplets. Comprehensive experiments on the challenging VG and GQA datasets showed that CFA significantly improves the performance of unbiased SGG. In the future, we would like to extend CFA to compose new triplets features by fusing features from open-vocabulary categories or images from other domains.

Acknowledgement. This work was supported by the National Key Research & Development Project of China (2021ZD0110700), the National Natural Science Foundation of China (U19B2043, 61976185), and the Fundamental Research Funds for the Central Universities(226-2023-00048). Long Chen was supported by HKUST Special Support for Young Faculty under Grant F0927.

References

[1] Chao Chen, Yibing Zhan, Baosheng Yu, Liu Liu, Yong Luo, and Bo Du. Resistance training using prior bias: toward unbiased scene graph generation. In *AAAI*, pages 212–220, 2022. 2, 7, 8

[2] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *ICCV*, pages 4613–4623, 2019. 1

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 6

[4] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, pages 6163–6171, 2019. 1

[5] Meng-Jiun Chiou, Henghui Ding, Hanshu Yan, Changhu Wang, Roger Zimmermann, and Jiashi Feng. Recovering the unbiased scene graphs from the biased ones. In *ACM MM*, 2021. 7

[6] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *ECCV*, pages 694–710, 2020. 2, 3

[7] Youming Deng, Yansheng Li, Yongjun Zhang, Xiang Xiang, Jian Wang, Jingdong Chen, and Jiayi Ma. Hierarchical memory learning for fine-grained scene graph generation. In *ECCV*, 2022. 7, 8

[8] Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. In *ICCV*, pages 15404–15413, 2021. 3

[9] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv*, 2017. 3

[10] Xingning Dong, Tian Gan, Xuemeng Song, Jianlong Wu, Yuan Cheng, and Liqiang Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *CVPR*, pages 19427–19436, 2022. 7, 8

[11] Yuyu Guo, Lianli Gao, Xuanhan Wang, Yuxuan Hu, Xing Xu, Xu Lu, Heng Tao Shen, and Jingkuan Song. From general to specific: Informative scene graph generation via balance adjustment. In *ICCV*, pages 16383–16392, 2021. 7

[12] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 4

[13] Tao He, Lianli Gao, Jingkuan Song, Jianfei Cai, and Yuan-Fang Li. Semantic compositional learning for low-shot scene graph generation. *arXiv*, 2021. 3

[14] Zhi Hou, Xiaojiang Peng, Yu Qiao, and Dacheng Tao. Visual compositional learning for human-object interaction detection. In *ECCV*, pages 584–600, 2020. 3

[15] Zhi Hou, Baosheng Yu, Yu Qiao, Xiaojiang Peng, and Dacheng Tao. Detecting human-object interaction via fabricated compositional learning. In *CVPR*, pages 14646–14655, 2021. 3

[16] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional

question answering. In *CVPR*, pages 6700–6709, 2019. 3, 6, 7

[17] Haeyong Kang and Chang D Yoo. Skew class-balanced re-weighting for unbiased scene graph generation. *Machine Learning and Knowledge Extraction*, 5(1):287–303, 2023. 2

[18] Keizo Kato, Yin Li, and Abhinav Gupta. Compositional learning for human object interaction. In *ECCV*, pages 234–251, 2018. 3

[19] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Generative compositional augmentations for scene graph prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15827–15837, 2021. 3

[20] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 2, 3, 6, 7, 8

[21] Lin Li, Long Chen, Yifeng Huang, Zhimeng Zhang, Songyang Zhang, and Jun Xiao. The devil is in the labels: Noisy label correction for robust scene graph generation. In *CVPR*, pages 18869–18878, 2022. 1, 3, 6, 7, 8

[22] Lin Li, Long Chen, Hanrong Shi, Hanwang Zhang, Yi Yang, Wei Liu, and Jun Xiao. Nicest: Noisy label correction and training for robust scene graph generation. *arXiv*, 2022. 3

[23] Lin Li, Jun Xiao, Guikun Chen, Jian Shao, Yueting Zhuang, and Long Chen. Zero-shot visual relation detection via composite visual cues from large language models. *arXiv preprint arXiv:2305.12476*, 2023. 1

[24] Lin Li, Jun Xiao, Hanrong Shi, Wenxiao Wang, Jian Shao, An-An Liu, Yi Yang, and Long Chen. Label semantic knowledge distillation for unbiased scene graph generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 1

[25] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *ICCV*, pages 8886–8895, 2021. 3

[26] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, pages 11109–11119, 2021. 1, 2, 3, 4, 7

[27] Wei Li, Haiwei Zhang, Qijie Bai, Guoqing Zhao, Ning Jiang, and Xiaojie Yuan. Ppdl: Predicate probability distribution based loss for unbiased scene graph generation. In *CVPR*, pages 19447–19456, 2022. 2, 3, 7, 8

[28] Xingchen Li, Long Chen, Jian Shao, Shaoning Xiao, Songyang Zhang, and Jun Xiao. Rethinking the evaluation of unbiased scene graph generation. In *BMVC*, 2022. 7

[29] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, pages 852–869, 2016. 1

[30] Yichao Lu, Himanshu Rai, Jason Chang, Boris Knyazev, Guangwei Yu, Shashank Shekhar, Graham W Taylor, and Maksims Volkovs. Context-aware scene graph generation with seq2seq transformers. In *ICCV*, pages 15931–15941, 2021. 2

[31] Xinyu Lyu, Lianli Gao, Yuyu Guo, Zhou Zhao, Hao Huang, Heng Tao Shen, and Jingkuan Song. Fine-grained predicates learning for scene graph generation. In *CVPR*, 2022. 2, 3

[32] Misaki Ohashi and Yusuke Matsui. Unbiased scene graph generation using predicate similarities. *arXiv preprint arXiv:2210.00920*, 2022. 1

[33] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 4, 5

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 4

[35] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, pages 3716–3725, 2020. 1, 3, 7, 8

[36] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, pages 6619–6628, 2019. 2, 3, 4, 6, 7

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. 6, 7

[38] Rahul Vigneswaran, Marc T Law, Vineeth N Balasubramanian, and Makarand Tapaswi. Feature generation for long-tail classification. In *ICVGIP*, pages 1–9, 2021. 2

[39] Meng Wei, Chun Yuan, Xiaoyu Yue, and Kuo Zhong. Hose-net: Higher order structure embedded network for scene graph generation. In *ACM MM*, pages 1846–1854, 2020. 5

[40] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 5410–5419, 2017. 6

[41] Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In *ACM MM*, pages 265–273, 2020. 2, 3, 7

[42] Jing Yu, Yuan Chai, Yue Hu, and Qi Wu. Cogtree: Cognition tree loss for unbiased scene graph generation. In *IJCAI*, 2021. 3, 4, 7, 8

[43] Qifan Yu, Juncheng Li, Yu Wu, Siliang Tang, Wei Ji, and Yueting Zhuang. Visually-prompted language model for fine-grained scene graph generation in an open world. *arXiv preprint arXiv:2303.13233*, 2023. 1

[44] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018. 1, 2, 3, 4, 6, 7, 8, 9

[45] Ao Zhang, Yuan Yao, Qianyu Chen, Wei Ji, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. Fine-grained scene graph generation with data transfer. In *ECCV*, 2022. 7

[46] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 6

[47] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, pages 5532–5540, 2017. 1

[48] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, pages 13001–13008, 2020. 3

[49] Liguang Zhou, Junjie Hu, Yuhongze Zhou, Tin Lun Lam, and Yangsheng Xu. Peer learning for unbiased scene graph generation. *arXiv preprint arXiv:2301.00146*, 2022. 2