Semantic Compositional Learning for Low-shot Scene Graph Generation

Tao He\textsuperscript{1}, Lianli Gao\textsuperscript{2}, Jingkuan Song\textsuperscript{2}, Jianfei Cai\textsuperscript{1}, Yuan-Fang Li\textsuperscript{1} \textsuperscript{*}

\textsuperscript{1}Faculty of Information Technology, Monash University
\textsuperscript{2}Center for Future Media, University of Electronic Science and Technology of China
{tao.he,yufang.li}@monash.edu, lianli.gao@uestc.edu.cn, jingkuan.song@gmail.com

Abstract

Scene graphs provide valuable information to many downstream tasks. Many scene graph generation (SGG) models solely use the limited annotated relation triples for training, leading to their underperformance on low-shot (few and zero) scenarios, especially on the rare predicates. To address this problem, we propose a novel semantic compositional learning strategy that makes it possible to construct additional, realistic relation triples with objects from different images. Specifically, our strategy decomposes a relation triple by identifying and removing the unessential component, and composes a new relation triple by fusing with a semantically or visually similar object from a visual components dictionary, whilst ensuring the realistic-ity of the newly composed triple. Notably, our strategy is generic, and can be combined with existing SGG models to significantly improve their performance. We performed comprehensive evaluation on the benchmark dataset Visual Genome. For three recent SGG models, adding our strategy improves their performance by close to 50%, and all of them substantially exceed the current state-of-the-art.

1. Introduction

Scene Graph Generation (SGG) is a fundamental task for many downstream applications in computer vision, including visual captioning \cite{zhu2019captioning,zhu2019visual} and visual question answering \cite{zhu2019vgqa}, as SGG can provide those tasks with a specific relation graph of an image, which makes it possible for the downstream models to conduct high-level graph reasoning over the generated scene graph and further capture more comprehensive information. Generally, a scene graph (SG) can be represented as a set of relation triples, i.e. \(<\text{subject, predicate, object}>\), or simply \(<s, p, o>\).

It has been observed that in real-world datasets, some predicates, e.g. flying in, are extremely rare. Moreover, the number of possible relations grows as the product of the number of subjects, objects and relations. With hundreds of subjects/objects in large-scale datasets, there can be millions of possible relations. These factors make the scene graph generation more challenging. Specifically, as shown in a recent study \cite{zhu2021motifs}, in the widely-used Visual Genome dataset \cite{krishna2017visual}, there are 150 object categories and 50 distinct relations. Thus, the total number of combinations of the relation triples is more than one million, while the total number of training images is only about 50,000. This fact leads to the arduous challenge that the majority of relation triples may only have a few samples, although the corresponding predicate has many samples. For example, in Visual Genome, the predicate sitting on belongs to a frequent predicate with more than 3,000 samples. However, if we focus on the number of samples on each relation triple, it can be observed that the majority of the samples belong to \(<\text{man, sitting on, chair}>\), but only several exemplars are for \(<\text{man, sitting on, car}>\). Therefore, even though a model may classify the former well, it may fail on the rare relation triples, e.g. \(<\text{man, sitting on, car}>\).

Furthermore, we also observe that given a relation triple,
Scene Graph Generation (SGG) is a task to detect visual relationships, which requires the model to localize and recognize them. The main challenge lies in the variability of combinations of subjects and objects, leading to the data-starvation issue for the majority of relationships. In the past, most of SGG methods predominantly focused on refinement of object and relation features early on \cite{23, 15, 24, 16, 8, 6, 9}, while consistently ignoring an important property of the Visual Genome (VG) dataset, that its relation distribution is long-tail. This biased distribution was first revealed in Motifs \cite{29}, which statistically showed the frequency of each relation. Interestingly, it was observed that when only using the statistical frequency information as prior knowledge to classify the relations, that is, without considering any visual feature, Motifs achieves competitive results, of only about 1.5% lower than state-of-the-art methods \cite{23, 15}. Inspired by Motifs, many recent works \cite{27, 20, 19, 4} invested heavily into solving the data bias or long tail problems in SG datasets. However, those methods still fail when facing those highly rare or unseen relation triples.

Compositional Learning has received tremendous attention in many visual tasks \cite{3, 11, 31}. The core idea of compositional learning is to use the limited samples to compose additional exemplars that maintain the main semantic meanings. Christopher et.al. \cite{3} proposed a compositional generative model, named as Multi-Object Network (MONet), to decompose scenes into abstract building blocks by several Variational Auto-Encoder networks. For the multi-label
classification problem, Alfassy et al. [11] proposed a feature compositional network to generate new multi-labels by simulating set operations. Kato, Li and Gupta [11] proposed a novel framework which incorporates an external knowledge graph and then uses graph convolutional networks to compose classifiers for objects pairs. Hou et al. [10] proposed a compositional learning strategy for the task of Human-Object Interaction (HOI) detection by composing new interaction samples in the feature space. Peyre et al. [17] used analogies to learn the compositional representations for subjects, predicates, and objects and during testing stage, they use the nearest neighbour search to retrieve the similar triples.

All of the three [11, 10, 17] works attempt to solve the zero-shot learning problem in HOI detection by composing unseen interactions. Our idea is different from them in two important aspects. Firstly, the subject in interaction triples of Human-Object Interaction is fixed, which leads to the fact that existing methods do not consider the impacts of the subject on the verbs (i.e. predicates) and only need to change the object. Therefore, the composition for scene graph relation is much more challenging that interactions in HOI. Secondly, existing works only consider the inter-class composition and consistently ignore the intra-class composition, leaving great potential improvement room.

3. Methodology

Figure 2 illustrates our overall framework, where the top half generally depicts convention SGG models and the bottom half shows our proposed compositional strategy. Our strategy can be divided into two stages: (1) anchor relation selection and decomposition, and (2) relation composition. Specifically, the first stage aims at selecting anchor relation triples, i.e. those that can be decomposed, and then decomposing them into two parts: essential and unessential. Then, we utilize a similarity measurement to find the best matched candidate from a visual components dictionary generated by an object detection network. In the second stage, we use the anchor relation and those candidates to compose new relations with inter- and intra-class instances.

3.1. Revising Scene Graph Generation Models

Generally, we could characterize most of SGG models [29, 30, 23, 20, 14] in three main steps, as shown the top part of Figure 2. The first step is to use an off-the-shelf pre-trained object detection network, e.g. Faster-RCNN [18], to extract spatial, semantic and visual object features. Next, they deploy a context learning module to refine those features, e.g. surrounding context [29] or edge direction context [14]. Finally, they construct subject-object pairs from the same image and deploy a predicate classifier to classify those object pairs. Formally, we could denote the three steps with the below formula:

\[ \mathbf{p} = \text{cls}(\mathcal{C}(\mathcal{F}(\mathcal{D}))) \]  

where \( \mathcal{D}(\cdot) \) is the object detection network, aiming at detecting visual components in the image I, i.e object proposals. \( \mathcal{F}(\cdot) \) is the feature refinement module, such as \( \mathcal{C}(\cdot) \) is a subject-object pairs construction function aiming at fusing two object features into a relation representation. \( \text{cls}(\cdot) \) is a final relation classifier and \( \mathbf{p} \) is the predicted relation score.

As mentioned before, the majority of existing works put their efforts in the modification of the feature refinement module \( \mathcal{F} \). However, a common and important issue of those works is that they only use the detected objects in the same image to compose the relation triples but ignore the fact that even two objects from different images can also compose a relation triple. Based on this observation, chiefly, we could express our idea as the formula below:

\[ \mathbf{p} = \text{cls}(\mathcal{C}'(\mathcal{F}(\mathcal{S}(\mathcal{V})))) \]  

where \( \mathcal{V} \) denotes visual components dictionary of training set detected by \( \mathcal{D} \); \( \mathcal{S} \) is a selection function to choose suitable anchor relation triples and visual components from all candidates; and \( \mathcal{C}'(\cdot) \) is a composition function but different from \( \mathcal{C}(\cdot) \) in Eq. 1. In the following subsections, we will describe each of them.

3.2. Visual Components Detection

The first step is to generate all the visual components of the dataset. Following most SGG models [29, 5], we also use the pre-trained Faster-RCNN [18] as the backbone object detection network. Subsequently, a region proposal network (RPN) is deployed to generate a set of \( n \) bounding boxes \( \mathcal{B}_i = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \) for each image \( i \), where \( \mathbf{b}_j = [x^j_i, y^j_i, x^j_b, y^j_b] \) denotes the top-left coordinate and \( (x^j_b, y^j_b) \) is the bottom-right coordinate. The corresponding object visual features are represented as \( \mathcal{O}_i = \{\mathbf{f}_1^i, \mathbf{f}_2^i, \ldots, \mathbf{f}_n^i\} \), aligned by a RoIAlign module [7], where \( \mathbf{f}_j^i \in \mathbb{R}^{4096} \) for each \( j \). Additionally, we extract the localisation feature of the object proposals and transform each bounding box \( \mathbf{b}_j \in \mathbb{R}^4 \) into a 128-dimensional informative spatial feature vector, denoted as \( \mathbf{f}^{s}_j \in \mathbb{R}^{128} \). Previous work [29, 19] have shown that the language prior knowledge can serve as auxiliary information to enrich the object features. Therefore, we further extract the word embedding features of the object label \( l_j \) as \( \mathbf{f}^{w}_j \in \mathbb{R}^{200} \). Finally, we concatenate the three types of features into a comprehensive vector as the visual component, i.e. \( \mathbf{f}_j = [\mathbf{f}^{s}_j; \mathbf{f}^{w}_j; \mathbf{f}^{w}_j] \in \mathbb{R}^{4424} \). Thus, the visual component dictionary \( \mathcal{V} \) is denoted as \( \{\mathbf{f}^{v}_j\}_{j=1}^m \cdot \{\mathbf{f}^{v}_j\}_{j=1}^n \cdot \{\mathbf{f}^{v}_j\}_{j=1}^m \), where \( m \) is the number of images.
3.3. Anchor Relation Selection and Decomposition

Our overall goal is to decompose a real relation and compose a new relation by fusing with another visual component. Theoretically, we could formulate the goal as the below:

\[
\begin{align*}
  f(r') &\approx f(r), \text{ where:} \\
  f(r') &= f(r) \ominus f(u) \oplus f(u')
\end{align*}
\]

(3)

where \( f \) is a feature learning function; and \( \ominus \) denotes the decomposition operation whereas \( \oplus \) is the composition operation. \( f(r) \ominus f(u) \oplus f(u') \) is the composed relation triple, \( u \) is the unessential object in the anchor relation \( r \) and \( u' \) is the other visual component from \( \mathcal{V} \). The selection of \( u' \) is described in the next section.

Anchor Relation Selection. As analysed before, we observe that in some relation triples, the subject and object have different impacts on their predicate, and removing the unessential one does not make a difference for their relationship. We name those relation triples that satisfy this property as Anchor Relations. Not all relation triples are anchor relations. Intuitively, if two objects in a relation triple elaborate, they should have less shared information, because much shared information means that both objects have a large impact on their relationships and their interaction is relatively tight, which makes it challenging to decompose or disentangle them. That is, \( f(r) \ominus f(u) \) could be too far away from \( f(r) \) and consequently Eq. 3 becomes hard to satisfy. On the other hand, the shared information can be an important clue for some predicates, but the decomposition operation could break this clue. For the selection of anchor relations, we need to choose those that the subject and the object are less entangled Therefore, if boxes of a subject and an object in a relation have a smaller intersected region, i.e., a smaller IoU score, they have less common information.

Anchor Relation Decomposition. This operation aims at determine, given an anchor relation \( r \), which of the subject or the object, denoted as \( u \) in Eq. 3 should be decomposed from the anchor relation. Naturally, the smaller object is usually inconsequential in most cases, because its features have less contribution to their union feature. Furthermore, the variety of small objects, such as bowl and fork, are much smaller than the big objects, e.g. man. Therefore, we choose the smaller object as the decomposed element.

In summary, given a triple \(<s, r, o>\), we formulate the selection and decomposition rules as follows:

\[
S(s, o) = \begin{cases} 
  s & \text{if } \text{IoU}(b_s, b_o) < \delta & \text{& } A(b_s) < A(b_o) \\
  o & \text{if } \text{IoU}(b_s, b_o) < \delta & \text{& } A(b_s) > A(b_o) \\
  -1 & \text{otherwise} 
\end{cases}
\]

(4)

where \( s \) and \( o \) denote the subject and object of a ground-truth relation triple; \(-1\) means the relation is not an anchor relation, otherwise \( S(.) \) returns the unessential element. \( A(b_s) \) is an area calculation function for a bounding box \( b_s \); \( \text{IoU}(b_s, b_o) \) returns the IoU score of two bounding boxes \( b_s \) and \( b_o \); and \( \delta \) is a threshold for the IoU score. In our experiments, we empirically set it to 0.3.

3.4. Relation Composition

In this stage, we aim at composing new relation triples based on anchor relations, where the key challenge lies in how to choose the best appropriate visual component, i.e. \( u' \) in Eq. 3 to fuse with the anchor relation so that the composed relation is semantically and visually close to the anchor relation. As illustrated in Eq. 3, our aim is to force the representation of the composed relation, i.e. \( f(r') \) in Eq. 3 to be close to the anchor relation \( f(r) \).

More concretely, we devise two types of composition strategies, intra- and inter-class compositions, according to
whether or not the object labels of $u$ and $u'$ are same. For the intra-class composition, we require both of $u$ and $u'$ belong to the same object category, which can solve the problem of low-shot relation classification problem, while the inter-class composition could compose novel relation triples not present in the training dataset, which helps the model tackle the zero-shot relation classification problem.

**Intra-Class Composition.** It is intuitive that the distance of two objects’ visual features is a good way to measure two same-category objects’ similarity, but the critical drawback lies in time efficiency. As object features are 4096-dimensional vectors, it is computationally infeasible to calculate the distance for every $u'$ across the entire set of visual components. As an alternative, we design a simple but effective metric on the shapes of two bounding boxes, as we deem that the shape of objects can be an important cue to distinguish them from each other. The main reasons are twofold. First, for the majority of static objects, such as dishes and bottles, their shape and pose does not change a lot, and thus they are visually similar if they have similar shapes. Second, for those dynamic object, e.g., persons or animals, to some extend, their shape information can imply their visual appearance features. It could be argued that the shape information may not be quite precise at representing these dynamic objects. However, since these objects always play an important role in the predicate, we usually do not decompose them, i.e. $u$ is less likely to be a dynamic object. Figure 3 gives some examples $u'$ retrieved by the shape of $u$.

Formally, given two bounding boxes $b_u$, $b_v$ of $u$ and $u'$ respectively, we first normalize their top left corners to $(0, 0)$, as below:

$$b_u = [0, 0, x_u^u - x_v^u, y_u^u - y_v^u]$$

$$b_v = [0, 0, x_v^u - x_v^v, y_v^u - y_v^v]$$

Then, we use the overlapping of two boxes to measure their similarity:

$$\text{InS}(b_u', b_v') = \min(x_u'^v, x_v'^v) \cdot \min(y_u'^v, y_v'^v)$$

$$\mathcal{M}(b_u', b_v') = \frac{\text{InS}(b_u', b_v')}{x_u'^v \cdot y_u'^v + x_v'^v \cdot y_v'^v - \text{InS}(b_u, b_v)}$$

where $x_u'^v$ and $x_v'^v$ are the normalized coordinates; InS calculates the overlap of two boxes; and $\mathcal{M}$ calculates two boxes’ shape similarity.

**Inter-Class Composition.** Different from the above, this composition operator does not require $u$ and $v$ have the same object category. Thus, we could compose unseen relation combinations, e.g. $<$man, sitting on, train$>$ could derive from the anchor relation $<$man, sitting on, car$>$. The key issue in this module is that we need to guarantee the composed relation is reasonable, for example, $<$man, sitting on, eye$>$ is highly unrealistic. To solve this issue, we first use the language prior, i.e. the word embedding of object category of $u$ to retrieve the most semantically similar object categories. For example, if $u$ belongs to car, the retrieved object categories are likely to be train, plane or bike, as all of them are vehicles, therefore their language prior features are close. When we obtain the feasible object categories, we use the same similarity measurement in intra-class composition to obtain the best appropriate visual components.

### 3.5. Training and Inference

**Training.** We use the Cross Entropy (CE) loss to optimize Eq. 2, and Kullback-Leibler (KL) loss to optimize Eq. 3 to force the distribution of composed relations not to be far from their anchor relations. It is worth noting that $F$ in Eq. 3 can be any feature refinement module, such as Motifs [29], TransE [30], or GPS-Net [13], etc.

In addition, our method is able to compose many extra relation triples whose labels depend on the anchor relation. Thus, to balance predicates in every training batch, we utilize a balanced sampling strategy to sample images. Specifically, we firstly randomly select $N$ predicates from all predicates for each training batch and then sample $K$ images for each sampled predicate. Therefore, the batch size is equal to $N \times K$. The benefit is that each predicate can be equally sampled, avoiding oversampling for frequent predicates and undersampling for the rare.

**Inference.** During testing, we discard the composition strategy, and use the standard SGG models, as defined in Eq. 1 to generate scene graphs.
4. Experiments

In this section, we evaluate our model on the task of relation retrieval (RR), as well as its two more challenging settings, namely few-shot and zero-shot relation retrieval (FR and ZR respectively). Furthermore, we discuss the effectiveness of each component in an ablation study. More experimental settings, results, discussions, a qualitative analysis, as well as source code can be found in the supplementary materials.

4.1. Dataset and Baselines

**Dataset.** We conduct our experiments on the most widely-used and challenging benchmark dataset Visual Genome (VG) [12]. Following the widely adopted split [23, 29][19][22], we choose the most frequent 150 object classes and 50 predicates to generate scene graphs, with 57,723 images for training and 26,443 images for testing. Additionally, 5,000 images make up the validation set to select the best model and to finetune model parameters.

**Baselines.** We apply our decomposition and composition operations (denoted Dec) to three open-source representative SGG models: Motifs [29], GPSN [14], and TransE [30], and compare performance with these models without DeC. Besides, we compare with a number of other recent SGG models: KERN [4], GLAT [28], GB-Net [27], and TDE [19]. For TDE, we choose TransE [30] as the base model and SUM [19] as the fusion function since it was shown that this combination obtains the best performance [19].

4.2. Implementation Details

Following recent works [29] [19], we use the pre-trained Faster-RCNN [13] as the object detection network and freeze its parameters for training our SGG models. For each batch, we set the number of predicates \(N = 5\) and number of images per predicate \(K = 1\). During training, we do not generate all visual components at once, given the large total number. Instead, we maintain a fixed-size visual component dictionary, whose size is set to 3,000. When the dictionary is full, we randomly choose components to be removed. For the relation classification, we set task-specific maximum number of relation triples for each image: 256 for PredCls and SGCls. For the task of SGDet, the maximum number of object proposals is set to 80 and the number of classification triples to 1,024. Additionally, the total number of training iterations is about 130,000 and the number of composed relation triples is about 600,000, including inter- and intra-class composition.

4.3. Evaluation Tasks and Metrics

We evaluate our model on the common task of relation retrieval (RR) and its few-shot (FR) and zero-shot (ZR) variants, which are increasingly difficult due to the reduction in training samples. Specifically, FR tests a model’s ability to learn from only a few training examples. We only sample a few samples (5 and 10) for each predicate to train the model. ZR [15] aims to evaluate a model’s generalizability on unseen relations. During the testing stage, we select new relation triples as evaluation triples, following TDE [19]. Each task includes three sub-tasks: (a) Predicate Classification (PredCls) with ground-truth object labels and boxes; (b) Scene Graph Classification (SGCls) with only the labels; and (c) Scene Graph Detection (SGDet) with neither of them.

In this paper, we report the unbiased metric mean Recall@K (mR@K) instead of the conventional one R@K, as it has been shown in the literature that R@K is biased and does not reflect a model’s true performance on tail relations [4] [19] [22]. Note that all experiments are under a constraint scheme [4].

4.4. Relation Retrieval (RR)

Table 1 shows the comparison results with the state-of-the-art methods. It can be clearly observed that when the three baseline models, TransE, GPSN and Motifs, are equipped with our proposed DeC strategy, their performance gains a significant improvement. For instance, for Motifs [29], with our proposed DeC added, it obtains approx. 2x improvement compared with the original Motifs across all three subtasks, and surpasses the best method TDE by nearly 50%. These results clearly demonstrate the effectiveness and generalisability of our DeC strategy.

For a more comprehensive comparison, we further test specific improvement on each predicate, as shown in Figure 3, where we report the results of R@100 of PredCls on two representative methods, Motifs and TDE, as the compared methods. The relationships on the x-axis are sorted by the numbers of instances in the training set, with the most frequent on the left. From this analysis, we can observe that while our model achieves competitive, albeit slightly worse performance on the head relations, it achieves significantly better performance on the tails relations, without 0 R@100 values on any relation. It can be seen that both Motifs and TDE achieve a near 0 R@100 value for all of the 15 least frequent relations (part of onwards). In contrast, our strategy achieves substantially better results, with R@100 scores of at least 40% for 10 of the 15 relations, and only one relation (part of) with a R@100 value below 20. Moreover, for the most challenging relation flying in with only 5 samples in the training set, our model achieves a 100% R@100 score, whereas neither Motifs and TDE recognises it (with a R@100 score of 0).

Interestingly, we can also find that the performance of our method moderately degrades in some head relations, e.g., has and on, compared with Motifs. We deem that...
the possible reason is that Motifs adds the frequency bias information in its prediction, and due to the extreme imbalanced distribution of the relations, in which the top-10 highly frequent relations take up almost 90% of samples, the added bias knowledge plays a positive role for these head relations. At the same time, this frequency bias is harmful to the tail relations for the same reason. To some extend, we can view it as overfitting on the head relations and underfitting on the tail relations. In contrast, our DeC strategy will supplement those rare relations and relationships and consequently make the model acquire much more balanced performance on all the relationships, thus effectively tackles the long-tail relation classification problem.

### 4.5. Few-shot Relation Retrieval (FR)

Few-shot learning for SGG is an important and realistic metric, as the tail relationships have very few samples. To evaluate the performance of FR, we randomly sample $S$ images for each predicate to train the models. More concretely, we select $S = [5, 10]$ as the number of images per predicate for training. We choose five representative baselines: Motifs, KERN and, GPSN, TransE, and TDE, and Table 2 shows the results of FR on the three subtasks, where we also apply our DeC strategy to Motifs, GPSN and TransE. It is worth noting that due to the balanced training set, the statistical bias knowledge proposed in Motifs [29] is discarded for all methods.

From the results it is easy to see the challenging nature of the FR task, where TDE achieves competitive results on the three tasks. When we add DeC to the three models, it can be clearly observed that the three methods gain results comparable to and better than TDE, even though their original versions are much worse than TDE. For instance, after plugging DeC to Motifs, we can find its performance is on average about 2 points higher than TDE in PredCls. Additionally, we can observe that both of the original Motifs and KERN struggle in FR, possibly because both of them highly depend on the biased prior knowledge, which becomes ineffective in the few-shot setup.

### 4.6. Zero-shot Relation Retrieval (ZR)

Table 3 shows the comparison results on the task of ZR. Following the settings in TDE [19], we select unseen relation triples of the test dataset as the evaluated samples. It is worth noting that the statistic prior knowledge utilised by Motifs and KERN does not cover unseen relations of the test set, thus is ineffectual for the classification of new triples. In contrast, when we apply our DeC strat-

| Method   | PredCls | SGCls | SGDct |
|----------|---------|-------|-------|
|          | mR@20   | mR@50 | mR@100 | mR@20   | mR@50 | mR@100 | mR@20   | mR@50 | mR@100 |
| KERN     | 13.5    | 17.9  | 19.1  | 8.1    | 9.8    | 10.2   | 5.1    | 6.2    | 7.4    |
| GLAT     | 17.8    | 19.3  |       | 9.9    | 10.4   |        | -      | -      | -      |
| GB-Net†  | 16.3    | 22.1  | 24.0  | 11.4   | 12.7   | 13.4   | 6.6    | 7.1    | 8.5    |
| TDE†     | 17.3    | 25.1  | 28.2  | 9.3    | 13.0   | 14.7   | 6.2    | 8.5    | 10.7   |
| TransE   | 13.6    | 17.6  | 19.4  | 6.7    | 8.1    | 8.9    | 5.5    | 7.0    | 8.2    |
| TransE+DeC| 23.2   | 31.4  | 33.1  | 12.8   | 13.9   | 16.5   | 8.7    | 10.6   | 12.7   |
| GPSN     | 15.4    | 20.2  | 22.8  | 10.2   | 11.2   | 12.6   | 7.2    | 8.4    | 9.8    |
| GPSN+DeC| 26.0    | 35.9  | 38.4  | 14.3   | 17.4   | 18.5   | 10.4   | 11.2   | 15.2   |
| Motifs   | 10.2    | 13.5  | 14.8  | 6.8    | 7.3    | 7.8    | 4.5    | 5.7    | 6.8    |
| Motifs+DeC| 26.2   | 35.7  | 38.6  | 15.0   | 18.4   | 19.1   | 10.1   | 13.2   | 15.6   |

Table 1. Comparison with the state-of-the-art SGG models in terms of mean Recall@K (mR@K). † denotes the results reproduced by our implementation based on released source code, and DeC denotes our proposed decomposition and composition method.
strategy to choose
composition; (4) +intra, denoting that the baseline model with the intra-class
composition shows superiority to inter-class, because the
majority of relations in the test set consists of common re-
lations, and only a small subset of them are unseen by the
training model. Thus, the intra-class composition plays a
more essential role on the RR task. As for the task of
ZR, since all relations have only few samples, the intra-
class strategy can compose many extra same relations which
leads to a significant improvement. Last, for the ZR task
that simply evaluates the unseen relations, the inter-class
composition exactly creates novel relation samples, and
shows a better effectiveness than the intra-class composition.
Addition, when we change the similarity measurement
in Eq. 3 instead of our proposed strategy, we can clearly observe consider-
able improvements, especially on GPSN, which enjoys the
largest performance lift. Even though on the sub-task of
PredCls, TDE achieves the best performance in terms of
R@100, TDE shows deteriorated performance on the other	
sub-tasks.

4.7. Ablation Study

We further study the effectiveness of each module in our
framework for the three tasks: RR, FR and ZR. Specifi-
cally, we divide our framework into five variant: (1) baseline
without any our proposed strategy; (2) +inter, denoting
the baseline model with the inter-class composition; (3)
+intra, denoting that the baseline model with the intra-class
composition; (4) +Dec′, denoting the baseline with a ran-
dom strategy to choose \(v\) for \(u\) in Eq. 3 instead of our pro-
posed similarity measurement; and (5) +Dec, which is our
full model. We choose Motifs as the baseline model, and
the results are shown in Table 4. Note that on the task of
FR, we set the \(S=10\) for each predicate.

From the results, we could obtain the following obser-
vations. On the three tasks, both the inter- and intra-class compositions make positive contributions to model performance, but they have different task-specific effectiveness. More concretely, on the main task of RR, the intra-class composition shows superiority to inter-class, because the majority of relations in the test set consists of common relations, and only a small subset of them are unseen by the training model. Thus, the intra-class composition plays a more essential role on the RR task. As for the task of FR, since all relations have only few samples, the intra-class strategy can compose many extra same relations which leads to a significant improvement. Last, for the ZR task

Table 2. The results of few-shot relation retrieval (FR) on VG.

| Method   | SGDet  | SGCls  | PredCls  |
|---------|--------|--------|----------|
|         | mR@50/100 | mR@50/100 | mR@50/100 |
| Kern [4] | 1.8 / 3.2 | 3.0 / 4.5 | 7.1 / 8.4 |
| TDE [19] | 2.1 / 3.3 | 4.2 / 5.2 | 9.1 / 11.7 |
| TransE+Dec [30] | 0.7 / 1.2 | 2.2 / 3.4 | 4.6 / 6.5 |
| GPSN+Dec [14] | 1.0 / 2.1 | 2.8 / 4.4 | 8.3 / 10.3 |
| Motifs+Dec [29] | 1.4 / 2.7 | 2.7 / 4.0 | 7.6 / 9.5 |

Table 3. The results of zero-shot relation retrieval (ZR) on VG.

| Method   | SGDet  | SGCls  | PredCls  |
|---------|--------|--------|----------|
|         | R@50/100 | R@50/100 | R@50/100 |
| Kern [4] | 0.7 / 1.3 | 2.6 / 3.4 | 11.4 / 15.1 |
| TDE [19] | 1.9 / 2.7 | 2.9 / 3.6 | 13.2 / 17.1 |
| GPSN+Dec [14] | 0.5 / 1.0 | 2.0 / 2.8 | 8.5 / 9.3 |
| TransE [4] | 1.4 / 3.3 | 1.9 / 4.3 | 10.8 / 12.5 |
| Motifs+Dec [29] | 0.1 / 0.5 | 2.1 / 3.0 | 10.2 / 14.2 |

Table 4. The ablation study on the three tasks: RR, FR and ZR.
