Improving Predicate Representation in Scene Graph Generation by Self-Supervised Learning

So Hasegawa , Masayuki Hiromoto , Akira Nakagawa , and Yuhei Umeda

Fujitsu Limited, Japan {hasegawa.sou,hiromoto,anaka,umeda.yuhei}@fujitsu.com

Abstract

Scene graph generation (SGG) aims to understand sophisticated visual information by detecting triplets of subject, object, and their relationship (predicate). Since the predicate labels are heavily imbalanced, existing supervised methods struggle to improve accuracy for the rare predicates due to insufficient labeled data. In this paper, we propose SePiR, a novel self-supervised learning method for SGG to improve the representation of rare predicates. We first train a relational encoder by contrastive learning without using predicate labels, and then fine-tune a predicate classifier with labeled data. To apply contrastive learning to SGG, we newly propose data augmentation in which subject-object pairs are augmented by replacing their visual features with those from other images having the same object labels. By such augmentation, we can increase the variation of the visual features while keeping the relationship between the objects. Comprehensive experimental results on the Visual Genome dataset show that the SGG performance of SePiR is comparable to the state-of-the-art, and especially with the limited labeled dataset, our method significantly outperforms the existing supervised methods. Moreover, SePiR's improved representation enables the model architecture simpler, resulting in 3.6x and 6.3x reduction of the parameters and inference time from the existing method, independently.

1. Introduction

Scene graph generation (SGG) is a task that aims to capture high-level understanding of images or videos through a graph whose nodes and edges represent the objects and their relationships (predicates), respectively. Informative scene graphs have a potential to be effective for visual question answering (VQA) [33, 15, 29], image captioning [47, 38, 25], and image retrieval [41, 16]. With the advances in graph and object representations by deep learning, there has been tremendous progress in scene graph generation [35, 37, 44, 32, 6, 31, 22, 18, 4]. However, many challenges still exist mainly due to the imbalanced predicate class distribution.

Predicate classes of Visual Genome (VG) [17], for example, which is a widely-used dataset for scene graph generation, consist of massive abstract predicates (*e.g.*, on, has) and rare informative predicates (*e.g.*, standing on, carrying). In other words, the predicate classes have a heavily long-tailed distribution. Hereinafter, the massive abstract predicates are referred as *head* categories, the rare informative predicates as *tail* categories, and the remaining predicates as *body* categories. Since most of the existing SGG methods are based on supervised learning, they are strongly affected by such a biased dataset having many predicates in the head categories and few in the tail categories. As a result, these methods tend to create less informative scene graphs that contain a lot of abstract predicates but few detailed predicates. To tackle this issue, various methods have been proposed including re-sampling strategies [18], re-weighting functions [36], and debiasing methods [31, 8, 4]. These methods, however, do not contribute to increasing variations of data on the tail categories and rendering predicate representation robust, leading to overfitting to the tail categories at the expense of performance for the head categories.

In this paper, we propose SePiR (**Se**lf-supervised learning for **P**red**i**cate **R**epresentation), which is a novel self-supervised learning method to improve predicate representation for the tail categories. To resolve the problem that insufficient labeled data are available for the tail categories, we adopt self-supervised techniques, specifically contrastive learning methods such as MoCo [14] and SimCLR [5]. They can learn robust representation without using target labels through data augmentation. The overview of the proposed method is shown in Fig. 1. SePiR mainly consists of three parts: (1) an object detector to generate object features, *i.e.*, visual features of the objects and their locations, (2) a relational encoder to extract relationship between the objects by self-supervised learning, and (3) su-

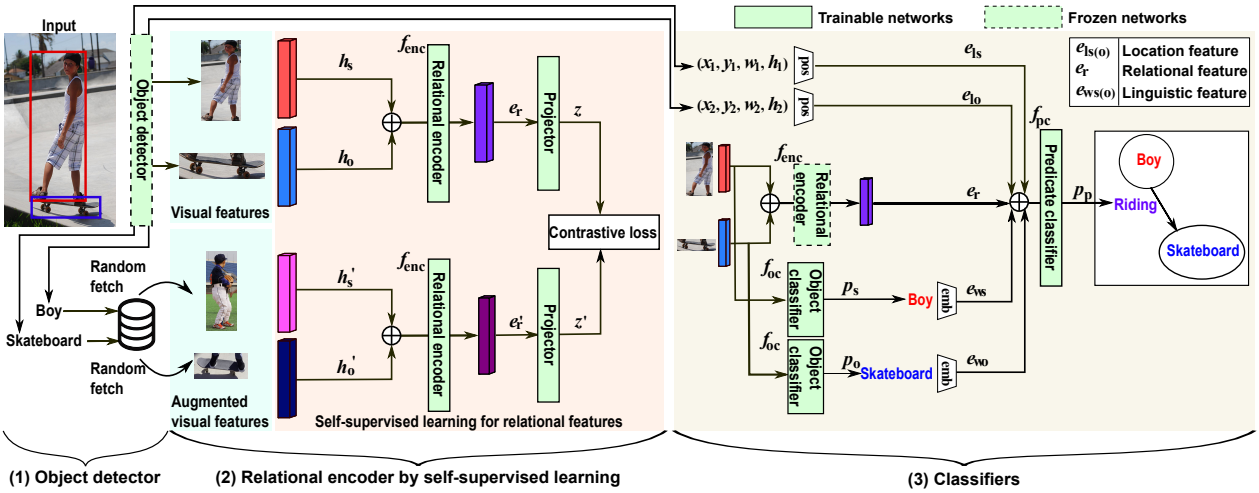

Figure 1. Overall pipeline of the proposed SePiR. (1) An object detector generates visual features of the subject (*boy*) and the object (*skateboard*). (2) Then a relational encoder is trained by self-supervised learning, where data augmentation for the subject-object pairs is realized by replacing the visual features with those having the same object labels. (3) Finally a predicate classifier is trained to predict predicate labels between the objects (*riding*) from three features: relational feature by the pre-trained relational encoder, location feature, and linguistic features.

pervised classifiers to predict object and predicate labels by using the pre-trained relational encoder. For the self-supervised part, we propose a data augmentation scheme that is effective to extract essential representations of the relationship. We realize such augmentation by replacing visual features of subject-object pairs with those from other images having the same object labels. This can increase the variation of the visual features of the subject-object pairs while keeping the relationship between the objects, which contributes to learning robust predicate representations of the tail categories in spite of the small amount of the labeled data. In addition, we introduce an attention-based object detector such as DETR [2] to capture object-specific visual feature that is effective for above augmentation.

Our extensive experimental results show that SePiR achieves competitive SGG performance to the state-of-the-art supervised methods on VG dataset. Notably, our method significantly outperforms the existing methods on the limited labeled dataset, which indicates that our self-supervised method captures good representation of the predicates even if only a small portion of the labeled data are available. In addition, the number of model parameters and the inference time is drastically reduced by 3.6x and 6.3x independently from those of the state-of-the-art methods. This is due to the simpler model architectures of the encoder and the classifiers employed in the proposed method than those of the existing works because the improved representation can reduce the burden on the classifiers.

The contribution of this work is summarized as follows.

- We propose SePiR, which is the first self-supervised method to improve predicate representation for the scene graph generation.
- Our self-supervised relational encoder can extract robust relational features not by using predicate labels but by adopting data augmentation for subject-object pairs, in which visual features of the objects are replaced with those from other images having the same object labels.
- Experimental results show that SGG performance of SePiR is comparable to state-of-the-art supervised methods on full dataset, and superior to them on harder limited labeled dataset.
- SePiR also reduces the number of the model parameters and inference speed compared to the existing methods.

2. Related Work

Scene graph generation. Early scene graph generation methods emphasize on incorporating contexts via a message passing [35, 18], recurrent neural networks [44, 32], graph neural networks [6], or attention-based models [9, 23, 4] using Transformer [34]. Recent methods have focused on addressing the imbalanced predicate class distribution. Many debiasing methods [31, 8, 4] are proposed to remove the intrinsic bias in VG. Other methods have tackled the issue by applying re-sampling [18] or re-weighting of loss functions [36]. However, as long as they utilize supervised learning methods, they cannot decipher the intrinsic long-

tailed distribution, resulting in overfitting to the tail categories while sacrificing the performance of the head categories. Aside from supervised methods, there are mainly three methods to deal with the imbalance distribution: semi-supervised methods, weakly-supervised methods, and self-supervised methods. Although semi-supervised methods [39] need only a small fraction of annotated images, they still require human labor. Since weakly-supervised methods [40, 46, 30] attempt to acquire predicate labels by utilizing corresponding image captions, they end up to be constrained by the target captions. Therefore, we introduce the self-supervised learning method to SGG to be released from manual annotations and limitations of other target domains.

Self-supervised learning for computer vision. Self-supervised learning has a potential to acquire a general encoder with the help of a myriad of unlabelled images. Early self-supervised learning methods for computer vision utilize heuristic methods: predicting rotations [11], solving jigsaw puzzles [26], and colorizing grayscale images [45]. Recent methods [5, 14, 12, 7, 43, 1, 3, 10] take different approaches by incorporating augmentations and a contrastive loss, resulting in having the comparable performance to the supervised methods on downstream tasks such as image recognition. In these methods, two different views are created from a single image (*i.e.*, data augmentation), and the similarity loss between the encoded embeddings of the two views is calculated. Then, the parameters of the encoder are updated to minimize the similarity loss. Since naive optimizations lead to generating constant values (this phenomenon is called *collapse*), various methods to avoid the collapse have been proposed. Contrastive learning (*e.g.*, MoCo [14] and SimCLR [5]) avoids the collapse by utilizing a lot of negative samples. On the other hand, non-contrastive learning (*e.g.*, BYOL [12], SimSiam [7], and Barlow Twins [43]) avoids collapse by several heuristic methods instead of using negative samples. However, these methods are designed to capture the representation of objects, which is effective for the tasks like image recognition, object detection, and instance segmentation. We cannot apply such methods to SGG as it is unless adaptations to the predicate representation is considered.

3. Proposed Method

We address the imbalanced class distribution in scene graph generation by improving the predicate representation with the help of self-supervised learning. In previous self-supervised learning methods [5, 14, 12, 7, 43, 1], data augmentation preserving essential features of the input is the key technology to learn better representation. More concretely, they augment input images by applying transformation like resizing and color distortion so as not to change the information of the object in the image. In order to apply the self-supervised learning for predicate representation, we must consider what data should be augmented and what kind of augmentation should be applied to them. For the first question, we augment *subject-object pairs* since they have information to decide the predicate between the subject and the object. For the second question, we propose a new augmentation strategy that can increase the variation of the subject-object pairs while preserving the relationships between them.

In this section, we first outline an overall training flow of SePiR. Then we focus on the detail of our self-supervised learning method for the scene graph generation.

3.1. Training Flow of SePiR

Figure 1 outlines our scene graph generation method, which consists of the following three training steps: (1) training an object detector, (2) training a relational encoder in a self-supervised manner, and (3) training classifiers for predicate and object labels.

Object detector. Given a dataset $\mathcal{D}$ and an image $I \in \mathcal{D}$, an object detector plays a role in proposing candidates of the objects and generates their visual features $\boldsymbol{H}$, tentative object labels $\boldsymbol{V}_{\mathrm{t}}$, bounding boxes $\boldsymbol{B}$, and confidence scores $\boldsymbol{C}$ of $\boldsymbol{V}_{\mathrm{t}}$. In the first step, an object detector is trained in a supervised manner. We adopt an attention-based object detector (*e.g.*, DETR [2] and Deformable DETR [48]) instead of conventional object detectors based on region proposal network (RPN) (*e.g.*, Faster RCNN [28] and Feature Pyramid Network [19]). Object features extracted by an RPN-based detector occasionally contain unrelated information to object labels, such as backgrounds or other objects. To reduce the influence of this information, attention-based object detectors are beneficial because it is able to detect boundaries of objects via the attention mechanisms and only extract object-related features. After training the object detector, the model parameters will be frozen in the remaining steps.

Relational encoder by self-supervised learning. A relational encoder generates a relational feature $\boldsymbol{e}_{\mathrm{r}}$ from the visual features of the subject-object pair as

$$\boldsymbol{e}_{\mathrm{r}} = f_{\mathrm{enc}}([\boldsymbol{h}_{\mathrm{s}}, \boldsymbol{h}_{\mathrm{o}}]), \tag{1}$$

where f_{enc} represents the relational encoder, $\boldsymbol{h}_{\mathrm{s}}, \boldsymbol{h}_{\mathrm{o}} \in \boldsymbol{H}$ are the visual features of the subject-object pair, and $[\cdot, \cdot]$ indicates the concatenation of the feature. How to create the subject-object pairs and how to train the relational encoder in a self-supervised manner are explained in Sec. 3.2. After training the relational encoder, the model parameters will be frozen in the last classification step.

Classifiers. In the last step, an object classifier and a predicate classifier are trained with the pre-trained object detector and the relational encoder. The object classifier is used to predict the final object labels $\boldsymbol{p}_{\mathrm{s}}$ and $\boldsymbol{p}_{\mathrm{o}}$ for the subject and the object by

$$\boldsymbol{p}_{\mathrm{s}} = \mathrm{Softmax}(f_{\mathrm{oc}}(\boldsymbol{h}_{\mathrm{s}})), \tag{2}$$

$$\boldsymbol{p}_{\mathrm{o}} = \mathrm{Softmax}(f_{\mathrm{oc}}(\boldsymbol{h}_{\mathrm{o}})), \tag{3}$$

where f_{oc} represents the object classifier. After the object labels of the subject-object pairs are determined, multiple inputs are fed to the predicate classifier to predict the predicate labels,

$$\boldsymbol{p}_{\mathrm{p}} = \mathrm{Softmax}(f_{\mathrm{pc}}([\boldsymbol{e}_{\mathrm{r}}, \boldsymbol{e}_{\mathrm{ls}}, \boldsymbol{e}_{\mathrm{lo}}, \boldsymbol{e}_{\mathrm{ws}}, \boldsymbol{e}_{\mathrm{wo}}])), \tag{4}$$

where f_{pc} is a function for the predicate classifier. $\boldsymbol{e}_{\mathrm{ls}}$ and $\boldsymbol{e}_{\mathrm{lo}}$ are location features computed by the function *pos* that takes corresponding bounding boxes as inputs. $\boldsymbol{e}_{\mathrm{ws}}$ and $\boldsymbol{e}_{\mathrm{wo}}$ are linguistic features generated by the GloVe [27] embedding function *emb*, which takes the corresponding object labels $\boldsymbol{p}_{\mathrm{s}}$ and $\boldsymbol{p}_{\mathrm{o}}$ as inputs. We use such location and linguistic features to enhance information about the interaction between subjects and objects, because the relational feature $\boldsymbol{e}_{\mathrm{r}}$ obtained by the proposed self-supervised learning may not have enough interactive information. The effect of these features are shown in the ablation study in Sec. 4.5.

To optimize the object classifier and the predicate classifier, we calculate a loss as

$$L = L_{\mathrm{o}} + L_{\mathrm{p}}, \tag{5}$$

where, L_{o} is a focal loss [20] for object classification as

$$L_{\mathrm{o}} = -\alpha(1 - \boldsymbol{p}_{\mathrm{o}})^{\gamma} \log(\boldsymbol{p}_{\mathrm{o}}), \tag{6}$$

and L_{p} is a predicate classification loss calculated by a standard cross entropy.

3.2. Self-Supervised Learning for Relational Features

In recent contrastive and non-contrastive self-supervised learning methods, large batch size is desired to enhance the robustness of the models [5, 14]. Hence, unlike conventional scene graph generation methods utilizing target bounding boxes to generate subject-object pairs, we would like to use as many pairs as possible. However, not all the possible subject-object pairs are reliable because the object detector sometimes makes false predictions. In this section, we first describe how to truncate subject-object pairs, and then explain the details of the proposed self-supervised learning method based on the truncated subject-object pairs.

How to truncate subject-object pairs. Truncation of the subject-object pairs is performed by the following two steps. First, the pre-trained object detector outputs a fixed number of objects, and the number of objects is reduced by a merging technique. Then, after all the possible subject-object pairs are created (*e.g.*, if the number of objects after merging is 50, the number of possible pairs becomes $50 \times (50 - 1) = 2450$), they are truncated using a pair confidence threshold.

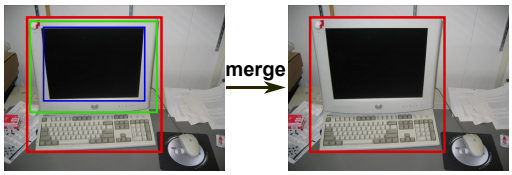

Figure 2. An example of merging technique. Since attention-based object detectors tend to produce multiple proposals indicating the same object (in this case, a *computer*), these bounding boxes are merged into a single proposal.

We decrease the number of detected objects by a merging technique to reduce redundancy. Since recent attention-based object detectors [2, 24] do not perform non-maximum suppression (NMS) unlike the RPN-based object detector, we have observed that the attention-based object detectors tend to detect all the possible regions of each object in the image. The example is shown on the left in Fig. 2. In the figure, three bounding boxes with different shapes indicate the same computer. In that case, if the computer is set as a subject and the computer mouse is set as an object, three $computer \rightarrow mouse$ pairs that contain similar information to each other are generated. These redundant pairs are meaningless because they do not contribute to increasing variations in a batch. Hence, we merge these multiple bounding boxes like the right side of Fig. 2 so that the merged bounding box has maximized information about the computer. The detailed procedure of the merging technique is explained in supplementary material A.1.

In addition to the merging technique, we truncate the number of pairs using the pair confidence so that the model is trained efficiently and reliably. We introduce a pair confidence c_{pair}, which indicates a probability of whether the pair is formed or not, for each subject-object pair as

$$c_{\mathrm{pair}} = c_{\mathrm{s}} \times c_{\mathrm{o}}, \tag{7}$$

where $c_{\mathrm{s}}, c_{\mathrm{o}} \in \boldsymbol{C}$ represent the confidence scores of the objects belonging to each subject-object pair, which are generated by the pre-trained object detector. Then, we extract the pairs that satisfy a requirement $c_{\mathrm{pair}} > c_{\mathrm{th}}$, where c_{th} is a threshold of the pair confidence.

Details of the self-supervised learning. An overview of the self-supervised learning for the relational encoder is shown on the left side of Fig. 1. Among outputs of the pre-trained object detector, we harness visual features $\boldsymbol{H}$

and tentative object labels $\boldsymbol{V}_{\mathrm{t}}$. In previous contrastive and non-contrastive self-supervised learning methods (*e.g.*, SimCLR [5] and BYOL [12]), they create two views by applying different augmentations to a single image. Geometric and color transformations are commonly used as data augmentations for the self-supervised learning of visual features. However, such augmentations cannot be applied to our predicate case because predicates are implicit features and are not explicitly shown in images. Therefore, we introduce the following augmentation method. First, we randomly fetch two images $I'_{\mathrm{s}}, I'_{\mathrm{o}} \in \mathcal{D}$ that respectively contain objects having the same object labels as the tentative object labels $v_{\mathrm{ts}}, v_{\mathrm{to}} \in \boldsymbol{V}_{\mathrm{t}}$, which are given to the subject and the object by the pre-trained object detector. Second, visual features $\boldsymbol{h}'_{\mathrm{s}}$ and $\boldsymbol{h}'_{\mathrm{o}}$ are extracted from the same-labeled objects in I'_{s} and I'_{o}, respectively. Then the original visual features $\boldsymbol{h}_{\mathrm{s}}$ and $\boldsymbol{h}_{\mathrm{o}}$ are replaced by $\boldsymbol{h}'_{\mathrm{s}}$ and $\boldsymbol{h}'_{\mathrm{o}}$ for data augmentation.

After the augmentation process, both the original object feature set $(\boldsymbol{h}_{\mathrm{s}}, \boldsymbol{h}_{\mathrm{o}})$ and augmented set $(\boldsymbol{h}'_{\mathrm{s}}, \boldsymbol{h}'_{\mathrm{o}})$ are fed into the relational encoder, and relational features $\boldsymbol{e}_{\mathrm{r}}$ and $\boldsymbol{e}'_{\mathrm{r}}$ are produced. Then the projector generates projections of $\boldsymbol{e}_{\mathrm{r}}$ and $\boldsymbol{e}'_{\mathrm{r}}$ as $\boldsymbol{z}$ and $\boldsymbol{z}'$ for the calculation of contrastive loss. Finally, the relational encoder and projector are optimized by the same contrastive loss function proposed in SimCLR [5] using all projections. The loss function aims to maximize the similarity of relational features between the original and augmented pairs, while reducing the similarity between the original and negative pairs.

4. Experiments

4.1. Dataset and Evaluation Metrics

Dataset. We evaluate our model on VG [17]. The most frequent 150 object categories and 50 predicate categories are chosen for evaluation, following the same split procedure in the previous works [44, 35]. After filtering out images that do not have bounding boxes or predicates according to an official implementation of [18], we utilize 57,723, 5,000, and 26,446 images for training, validation, and evaluation. The number of total triplets is 405,860, 33,203, and 183,642 in a train set, a validation set, and a test set.

Evaluation metrics. Our proposed model is evaluated on three sub-tasks: (1) predicate classification (PredCls), (2) scene graph classification (SGCls), and (3) scene graph detection (SGDet). PredCls is the simplest task, where a model aims to predict only predicates when the true bounding boxes and object labels are given. In SGCls, only the target bounding boxes are given and the model is required to predict object labels and predicates between them. As for SGDet, the model is used to predict triplets without being informed of target bounding boxes and object labels.

The metrics we used are recall@K (R@K), which is a fraction of times the correct relationship is predicted in the top K confident relationship predictions, and mean recall@K (mR@K), which is a mean of R@K for each predicate category. The mean recall is proposed by VCTree [32] and KERN [6] to reduce the effect of long-tailed imbalanced class distribution in the dataset. We can see performance for the rare predicate more clearly by the mean recall than the ordinary recall metric.

4.2. Implementation Details

Detailed settings for the three training steps of SePiR are described as follows.

Object detector. We adopt Conditional DETR [24] as the attention-based object detector. We train the Conditional DETR with ResNet101-DC5 backbone in a supervised manner on VG for 50 epochs, starting with the pre-trained weights for MS COCO [21]. We train the object detector using 8 GPUs with batch size 8 (1 batch size on 1 GPU). Except for the batch size, we use the same hyperparameters as an official implementation of [24]. After 50-epochs training, we achieve a detection performance of mAP $= 30.5$ on the test set of VG.

Relational encoder by self-supervised learning. In a self-supervised manner, we train the relational encoder for 10 epochs with 32 batch size utilizing LARS [42] optimizer on single A100 GPU. The learning rate starts from 0.02 and gradually decreases by 0.9999 per iteration. The details of the network architectures are described in supplementary material A.2. We set the threshold of pair confidence to $c_{\mathrm{th}} = 0.09$ by observing the behavior of the object detector, which is described in supplementary material A.3.

Classifiers. We train the predicate classifier and the object classifier for 40,000 iterations with 16 batch size on a single A100 GPU. We set $\alpha = 0.25$ and $\gamma = 2.0$ in focal loss for object classification in Eq. (6), according to the original paper [20]. To exhibit a capacity of the SGG performance ranging from high R@100 to high mR@100, we combine SePiR with three debiasing methods for the imbalanced class distribution: Bilevel sampling [18], RTPB (CB) [4], and a re-weighting loss. The details of the debiasing methods are described in supplementary material A.4.

4.3. Comparison with the Previous Methods

Scene graph generation performance. Experimental results of the proposed method and the state-of-the-art existing methods on the three sub-tasks are shown in Table 1. The results of R/mR@20 and 50 are in supplementary material B.1. Especially, SePiR incorporating the re-weighting

Models	PredCls		SGCls		SGDet	
	R@100	mR@100	R@100	mR@100	R@100	mR@100
KERN [6]	67.6	19.2	37.4	10.0	29.8	7.3
GPS-Net [22]	68.8	22.8	40.1	12.6	31.7	9.8
PCPL [36]	52.6	37.8	28.4	19.6	18.6	11.7
BGNN [18]	61.3	32.9	38.5	16.5	35.8	12.6
Seq2Seq-RL [23]	68.5	30.5	39.0	16.2	34.4	12.1
DTrans + RTPB (CB) [4]	47.5	38.1	25.5	22.8	23.4	19.0
Motifs [44, 32]	67.1	15.3	36.5	8.2	30.3	6.6
Motifs + TDE [31]	55.8	28.3	29.5	15.2	8.4	9.9
Motifs + BA-SGG [13]	52.5	31.7	31.0	17.5	26.9	15.6
Motifs + RTPB (CB) [4]	42.5	37.7	26.9	20.6	22.5	15.5
VCTree [32]	68.1	19.4	38.8	10.8	31.3	8.0
VCTree + TDE [31]	54.5	26.6	31.2	13.4	23.3	10.3
VCTree + BA-SGG [13]	51.8	32.6	35.0	21.2	25.5	15.7
VCTree + RTPB (CB) [4]	43.3	35.6	30.0	**25.8**	21.3	15.1
SePiR + Bilevel sampling	64.6	33.2	36.8	18.5	32.1	13.1
SePiR + RTPB (CB)	31.8	40.3	18.1	20.7	15.6	16.4
SePiR + Reweight	28.9	**43.2**	15.9	23.6	16.6	**19.7**

Table 1. The performance of PredCls, SGCls, and SGDet on VG. The scores of the existing methods are referred from the cited papers. The bold font indicates the best mR for each task.

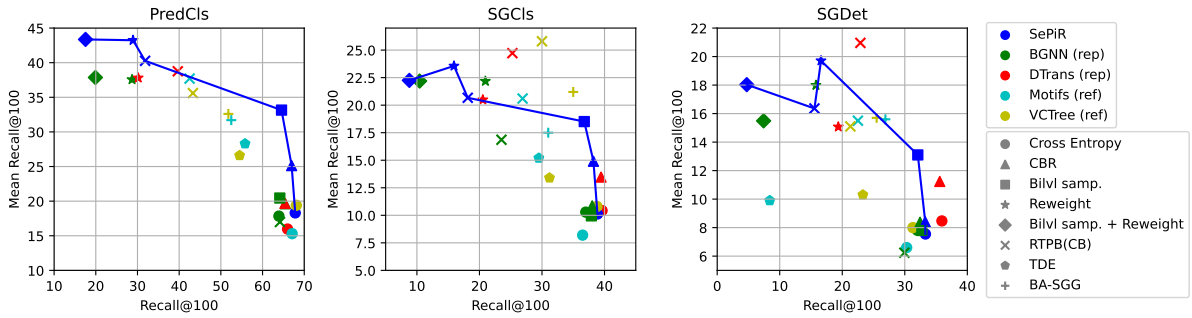

Figure 3. Trade-off curves between R@100 and mR@100 comparing SePiR and the existing methods (Motifs [44], VCTree [32], BGNN [18], and DTrans [4]) for PredCls, SGCls, and SGDet tasks on VG. *rep* means our reproduction, and *ref* means reference from the original papers. A model has an advantage if the trade-off curves locate in the upper right area, where both R@100 and mR@100 are high. SePiR is competitive to the state-of-the-art supervised methods.

loss (*Reweight*) achieves the highest mR@100 on PredCls and SGDet tasks.

For a fairer comparison, we devise an evaluation method using a trade-off between R@100 and mR@100 as shown in Fig. 3. Before delving into the result, we explain a way of looking at the graphs. Previous methods recently have focused more on mR@K than on R@K to corroborate that their methods are effective for the imbalanced class distribution of predicate classes. However, it would be undemanding to obtain high mR@K by sacrificing the performance of R@K. Denoting only R@K or mR@K is not enough to compare various scene graph generation models. Hence, we have to follow a direction that aims to achieve high mR@K while performing higher score on R@K and that plans to devise the evaluation method considering both mR@K and R@K. This motivation leads us to conceive the graph plotting the trade-off between mR@K and R@K as Fig. 3. Colors in the graph correspond to different models, and the markers in the graph indicate different methods for long-tailed class distribution, such as loss functions and sampling strategies, which are detailed in supplementary material A.4. With the trade-off curves, we are able to discern an advantage of the model if the markers are in the upper right area of the graph, where both R@K and mR@K are high.

In Fig. 3, we can see that our proposed method achieves competitive performance to the state-of-the-art supervised models, BGNN [18] and DTrans [4]. Note that since the results of BGNN and DTrans are from our reproduction experiment, the values are different from those of Table 1. The results of Motifs [44] and VCTree [32] are taken from literature and the same as Table 1 because we cannot reproduce their results from the published source codes.

As for PredCls, although SePiR slightly outperforms other methods for all the range, SePiR does not show explicit advantages for SGCls. This means that our method is especially good at predicate classification, which is the result we are aiming for. For SGDet, which is the most im-

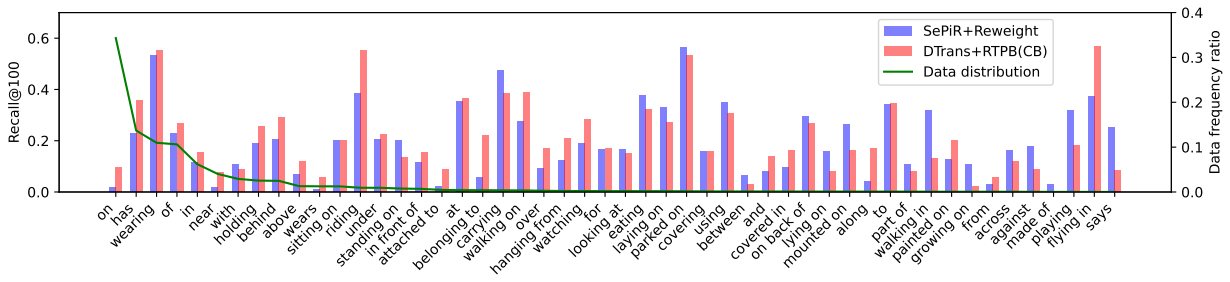

Figure 4. Comparison of SGDet R@100 for each predicate class between SePiR and DTrans (left axis). The distribution of the predicate classes in the test split is also shown in the green line (right axis). The predicates are sorted in descending order. SePiR outperforms DTrans in the rare predicate categories.

Models	# Parameters				Inference time (SGDet)
	Object detector	Relational Encoder	Classifiers	Total	
BGNN [18]	161.7 M	181.2 M		343.0 M	0.38 s
DTrans [4]	161.7 M	181.2 M		342.9 M	0.38 s
SePiR	**62.4 M**	**21.3 M**	**10.0 M**	**93.6 M**	**0.06 s**

Table 2. The computational performance of SePiR, BGNN [18], and DTrans [4]. A sum of parameters of *Relational Encoder* and *Classifiers* for BGNN and DTrans correspond to the number of parameters required in architectures except for the object detector. For SePiR, *Relational Encoder* includes the relational encoder and the projector, and *Classifiers* includes the object/predicate classifiers.

portant task, SePiR is comparable or superior to the other methods, except for DTrans. mR@100 of DTrans from our reproduction is 20.9 and higher than those in Table 1. The difference between SePiR and DTrans comes from the different architectures for improving object labels. Whereas the relational encoders in DTrans make use of contexts and predicate representation for refining object labels, SePiR does not utilize them.

Fig. 4 shows comparison of the individual R@100 score of SGDet for each predicate between SePiR+Reweight and our reproduced DTrans+RTPB(CB). We can see that SePiR achieves better prediction on tail categories than DTrans. The detailed comparisons for "head," "body," and "tail" categories are described in supplementary material B.1. Although the overall mR score of DTrans is higher than SePiR, the result shows that our method is more effective to predict long-tailed rare relationships.

Computational performance. To see the efficiency of the proposed model, we evaluated the number of the model parameters and the computation time in comparison to the state-of-the-art methods. The result is shown in Table 2. When it comes to the number of parameters, our model has the lowest amount of parameters ($\sim 90\,\mathrm{M}$) compared to those of the other models ($\sim 340\,\mathrm{M}$). The significant reduction comes from the simple relational encoder and the attention-based object detector. As for the relational encoder, the message passing used in BGNN and the attention-based architecture using Transformer in DTrans are computationally heavier than SePiR's encoder realized by simple multilayer perceptron. We also measure the inference time per image on a single A100 GPU taking the average of 26,446 test images. In the case of BGNN and DTrans, we use each official implementation. Our model achieves the fastest speed compared to state-of-the-art supervised methods on SGDet. In particular, our model runs about 6.3 times faster than other methods. The significant reduction also attributes to the MLP-centric architectures of the relational encoder in SePiR.

4.4. Experiments with Limited Labeled Data

In this section, the experimental results on the limited labeled dataset like [5] are presented to corroborate that our model acquires informative predicate representation by self-supervised learning. To be specific, we pre-train the relational encoder with the full train set of VG without predicate labels and train the predicate classifier with limited labelled datasets (randomly sampled 5%, 10%, and 30% of images in a train set of VG). For comparison, we also train state-of-the-art supervised methods, BGNN and DTrans, in a supervised manner with the same limited labeled datasets. We evaluate all the models on SGDet and utilize the trade-off graphs following the previous section for a fair comparison.

The result with the limited labeled dataset is shown in Fig. 5, and that experimented with the full dataset is exhibited in Fig 3. Clearly, we can see that SePiR locates in more right upper area than the existing supervised methods in all the limited labeled dataset whereas there seems less gap of the performance between SePiR and the supervised methods with respect to the fully labeled dataset. Especially in the case of 5% dataset, the highest mR@100 of SePiR is

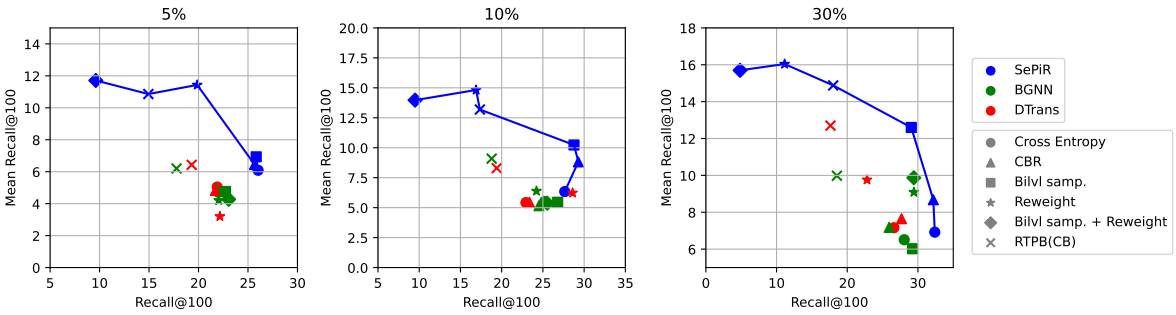

Figure 5. The comparison of SGDet performance to BGNN [18] and DTrans [4] with limited (5%, 10% and 30%) labeled dataset. We used the same debiasing methods as Fig. 3. Our method achieves much higher performance than state-of-the-art supervised methods in all the limited labeled dataset.

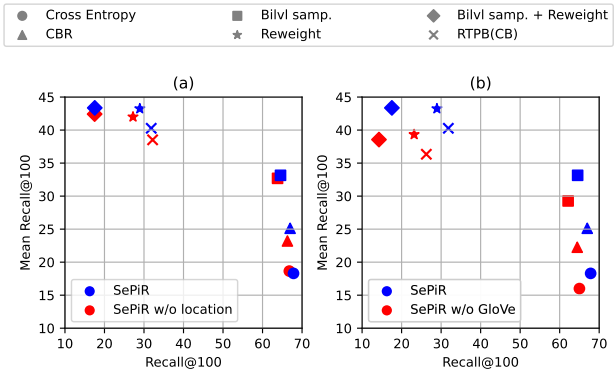

Figure 6. The comparison of performance on PredCls between SePiR and SePiR without (a) location feature and (b) linguistic feature. SePiR achieves higher performance than SePiR without each component.

about 6 points larger than the highest of mR@100 of the supervised methods, and the performances of the supervised methods stick to lower mR@100 regardless of the methods for the imbalanced class distribution. Comparisons for each predicate between SePiR and DTrans are shown in supplementary material B.2. SePiR has a large advantage on tail categories even with limited labeled dataset. The results indicate that SePiR is able to improve predicate representation without using large amount of labeled data by self-supervised learning, although the state-of-the-art supervised methods poorly capture the representation if only the limited labeled data are provided. This result also implies that SePiR has a potential to acquire more robust predicate representation if we harness tremendous of images that do not have predicate labels.

4.5. Ablation Study

SePiR utilizes location and linguistic features as inputs of the predicate classifier aside from the relational features. In this study, we corroborate the validity of using such features. More ablation studies about self-supervised methods, word embeddings, and object detectors are described in supplementary material B.3.

Location feature. For the feature to represent the location of the objects in images, SePiR utilizes the bounding box information. To see the effectiveness of using the location feature, we compare the performance of SePiR with and without the location feature. Fig. 6 (a) shows the relationship between R@100 and mR@100 for PredCls. We can see that SePiR achieves slightly higher performance than the method without location feature. The result shows that explicit knowledge of locations improves predicate predictions.

Linguistic feature. SePiR employs GloVe word embedding [27] as linguistic feature that helps the model predict predicates with the help of external linguistic knowledge. Fig. 6 (b) shows the performance of PredCls with/without using GloVe word embedding. Obviously, the performance of SePiR with GloVe is superior to that without GloVe, which indicates that linguistic feature is imperative to improve predicate predictions.

5. Conclusion

In this paper, we introduce SePiR, self-supervised learning method to improve predicate representation in scene graph generation. Experimental results show that our method is competitive to state-of-the-art supervised methods using full dataset and is superior to them in the case of limited labeled dataset, implying that SePiR can capture essential predicate representations for scene graph generation. For future work, we will enhance the model so that it can acquire more robust predicate representations by utilizing myriad of open set images. We consider that these representations are beneficial not only for scene graph generation but also for other tasks using predicate representation, such as human-object interaction and visual relation detection.

References

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with Transformers. In *ECCV*, pages 213–229, 2020.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, volume 33, pages 9912–9924, 2020.

[4] 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*, volume 36, pages 212–220, 2022.

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

[7] Xinlei Chen and Kaiming He. Exploring simple Siamese representation learning. In *CVPR*, pages 15750–15758, 2021.

[8] 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*, pages 1581–1590, 2021.

[9] Naina Dhingra, Florian Ritter, and Andreas Kunz. BGT-Net: Bidirectional GRU Transformer network for scene graph generation. In *CVPR*, pages 2150–2159, 2021.

[10] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *ICML*, pages 3015–3024, 2021.

[11] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent – a new approach to self-supervised learning. In *NeurIPS*, volume 33, pages 21271–21284, 2020.

[13] 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.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[15] Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. *arXiv:2007.01072*, 2020.

[16] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pages 3668–3678, 2015.

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv:1602.07332*, 2016.

[18] 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.

[19] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014.

[22] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. GPS-Net: Graph property sensing network for scene graph generation. In *CVPR*, pages 3746–3753, 2020.

[23] 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.

[24] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. In *ICCV*, pages 3651–3660, 2021.

[25] Kien Nguyen, Subarna Tripathi, Bang Du, Tanaya Guha, and Truong Q. Nguyen. In defense of scene graphs for image captioning. In *ICCV*, pages 1407–1416, 2021.

[26] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84, 2016.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Conf. Empirical Methods in Natural Lang. Process.*, pages 1532–1543, 2014.

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, volume 28, 2015.

[29] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *CVPR*, pages 8376–8384, 2019.

[30] Jing Shi, Yiwu Zhong, Ning Xu, Yin Li, and Chenliang Xu. A simple baseline for weakly-supervised scene graph generation. In *ICCV*, pages 16393–16402, 2021.

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

[32] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019.

[33] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *CVPR*, pages 1–9, 2017.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017.

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

[36] 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.

[37] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. In *ECCV*, pages 670–685, 2018.

[38] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, pages 10685–10694, 2019.

[39] Yuan Yao, Ao Zhang, Xu Han, Mengdi Li, Cornelius Weber, Zhiyuan Liu, Stefan Wermter, and Maosong Sun. Visual distant supervision for scene graph generation. In *ICCV*, pages 15816–15826, 2021.

[40] Keren Ye and Adriana Kovashka. Linguistic structures as weak supervision for visual scene graph generation. In *CVPR*, pages 8289–8299, 2021.

[41] Sangwoong Yoon, Woo Young Kang, Sungwook Jeon, SeongEun Lee, Changjin Han, Jonghun Park, and Eun-Sol Kim. Image-to-image retrieval by learning similarity between scene graphs. In *AAAI*, volume 35, pages 10718–10726, 2021.

[42] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv:1708.03888*, 2017.

[43] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, pages 12310–12320, 2021.

[44] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018.

[45] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666, 2016.

[46] Yiwu Zhong, Jing Shi, Jianwei Yang, Chenliang Xu, and Yin Li. Learning to generate scene graph from natural language supervision. In *ICCV*, pages 1823–1834, 2021.

[47] Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. Comprehensive image captioning via scene graph decomposition. In *ECCV*, pages 211–229, 2020.

[48] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.