2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) | 978-1-6654-4899-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/CVPRW53098.2021.00182

Target-Tailored Source-Transformation for Scene Graph Generation

Wentong Liao[1,*], Cuiling Lan[2], Michael Ying Yang[3], Wenjun Zeng[2], Bodo Rosenhahn[1]
[1]TNT, Leibniz University Hannover, Germany, [2]Microsoft Research Asia, Beijing, China,
[3]SUG, University of Twente, The Netherlands
[1]{liao,rosenhan}@tnt.uni-hannover.de,[2]{culan, wezeng}@utwente.nl, [3]{michael.yang}@utwente.nl

Abstract

Scene graph generation aims to provide a semantic and structural description of an image, denoting the objects (with nodes) and their relationships (with edges). The best performing works to date are based on exploiting the context surrounding objects or relations, e.g., by passing information among objects. In these approaches, to transform the representation of source objects is a critical process for extracting information for the use by target objects. In this paper, we argue that a source object should give what target object needs and give different objects different information rather than contributing common information to all targets. To achieve this goal, we propose a Target-Tailored Source-Transformation (TTST) method to propagate information among object proposals and relations. Particularly, for a source object proposal which will contribute information to other target objects, we transform the source object feature to the target object feature domain by simultaneously taking both the source and target into account. We further explore more powerful representation by integrating language prior with visual context in the transformation for scene graph generation. By doing so the target object is able to extract target-specific information from source object and source relation accordingly to refine its representation. Our framework is validated on the Visual Genome benchmark and demonstrated its state-of-the-art performance for the scene graph generation. The experimental results show that the performance of object detection and visual relationship detection are promoted mutually by our method. The code will be released upon acceptance.

1. Introduction

In recent years great successes have been witnessed on vision perceptual tasks such as object detection [20] and semantic segmentation [32]. However, such object-centric visual perception is still far from the goal of visual scene understanding which requires understanding the visual relationships between objects. Some recent work [8, 9] proposed to represent the visual scene as a scene graph which models objects and their attributes as nodes, and their relationships as edges, as illustrated in Fig. 1. Scene graph has been proved to be a promising alternative for many visual tasks such as image retrieval [8], image caption [27], visual question and answering [7].

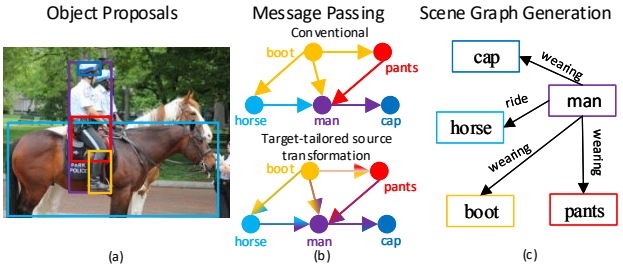

Figure 1. Given an image: (a) objects are proposed; (b) Messages are passed among objects to exploit context; (c) label the graph's nodes (objects) and edges (relations). In (b), we show the difference between the conventional message passing methods (top graph) and our target-tailored source-transformation ones (bottom graph). The arrows denote the message passing direction, the colors indicate the passed information. Different colors indicate the corresponding objects.

A natural idea to generate scene graph is to detect objects using an off-the-shelf object detector, and then predict their pairwise relationships *separately* [17, 31]. However, these approaches ignore the exploration of visual context, which could provide powerful inductive bias and strong regularities [30] that help detect objects and reason their relations. For example, "keyboard" and "mouse" often co-occur within a scene, and "man" tends to "ride" the "horse". Many works have exploited the visual context in different ways to help scene graph generation [24, 29, 11, 13, 3, 30]. Particularly, modeling message passing among objects is the most widely applied method for exploiting the visual context and its effectiveness has been proved for scene graph generation. In previous message passing methods, the representation of a source object is first transformed, via a learned *shared transformation* W before being remedied to

*This work was done when Wentong Liao was an intern at MSRA.

978-1-6654-4899-4/21/$31.00 ©2021 IEEE
DOI 10.1109/CVPRW53098.2021.00182

update the target object [25, 12, 13, 26]. To make the shared transformation suitable for any target object, W is unfortunately encouraged to learn information from source objects which is commonly useful for different target objects.

We argue that two important elements are overlooked in most of the existing message passing methods for scene graph generation. First, the semantic dependencies between target objects and source objects are ignored in the source-transformation step because the shared transformation matrix W is independent of the target object. For example as shown in Fig. 1(b)(top), a "boot" will contribute the same information to "person" as to "horse" (edges are denoted by the same color) with the shared transformation. Intuitively, a target-tailored information is more useful than a common information for a specific target object. For example, "boot" should contribute information of "wearing things" to "person" while contributing information of "riding gear" to "horse" (See Fig. 1(b)(bottom) edges are denoted by different colors). Second, how to effectively couple the visual and language context into the learning process has not attracted much attention. The visual appearance determines the visual context while the language prior guides how objects relate to each other in the linguistic domain. For instance, when we see "person on a motor" (visual context), we humans spontaneously infer the relation as "ride" rather than "on" or "sit" (language prior). These two modalities should be compatible and mutually promotive rather than implemented separately.

Motivated by these observations, we propose a *target-tailored source-transformation* (TTST) method for message passing to exploit context for scene graph generation. "Target-tailored" means that when a source contributes information to different targets, we expect it to deliver target specific information, *i.e*., give as exactly as possible what the target needs. To achieve this goal, the source and target are simultaneously considered in the transformation of the source information to the target domain. Furthermore, we propose to integrate the language priors with visual context in the transformation process. By doing so, messages are propagated through the graph more effective and the learned representations are more powerful.

Our framework is depicted in Fig. 2. It builds on the Faster R-CNN detector [20] to generate object proposals (Fig. 2(a)). Then, a graph is initialized by connecting each pair of objects. We introduce a learned semantic relationship filter (**SRF**, see Sec. 3.2) to prune the spurious connections between objects (Fig. 2(b)) to facilitate the subsequent message passing processes. Then, we apply message passing through the graph while considering the semantic dependencies between the source and the target objects and relationships (see Sec. 3.3, Fig. 2(c)(e)). Finally, the labels of graph nodes are predicted with the context-rich features, and the edge labels are inferred by using the refined relationship features along with the semantic information of the connected object nodes (Fig. 2(d)).

Summary of Contributions. Our work has two major contributions:

- A novel target-tailored source-transformation method for message passing, which explores information from source object by considering the source object and target object simultaneously.
- Language context is utilized to help message passing and integrated with visual context to learn powerful representation for scene graph generation.

Our method reports the state-of-the-art results on the VG benchmark dataset for scene graph generation. Moreover, the experimental results show the mutual improvements of object detection and relationship detection via our method.

2. Related Works

Context for Visual Reasoning. Context has been explored to improve different scene understanding tasks for decades [4, 10, 28, 6, 16]. However, context for improving scene graph generation is still under explored. A number of works attempt to capture object context from an image in the message passing mechanism, such as through a graph model [12, 26], implementing RNN [30, 1, 22], or in an iterative refinement process [24].

Besides visual context, context from language priors [18] has been proved to be helpful for visual relationships detection and scene graph generation [17, 29, 13]. Lu *et al*. [17] utilized language priors to improve the detection of meaningful relationships between objects. Li *et al*. exploited language priors from region captions for scene graph generation by predicting image caption and detecting visual relationships in parallel. Yu *et al*. [29] distilled linguistic knowledge by training a parallel language branch as a teacher network to help the visual network (student) predict visual relationships. In contrast to above works that utilize language prior separately, we integrate the language priors with visual context in the transformation step to help message passing and learn better semantic representations.

Scene Graph Generation. Scene graph was first proposed in [8]. It generalizes the task of detecting object to also detecting their attributes and reasoning relationships between them. Scene graph generation which includes object detection and visual relationship detection are attracting increasing attention in computer vision [13, 3, 14, 33, 12, 30, 23, 22, 1, 2]. Context has been proved to be useful for scene graph generation and many works resort to message passing to capture the context of the two related objects [11, 30, 2, 22], or of the objects and their relationships [24, 13, 12, 26]. A key process for message passing is first to transform the representation of source objects into

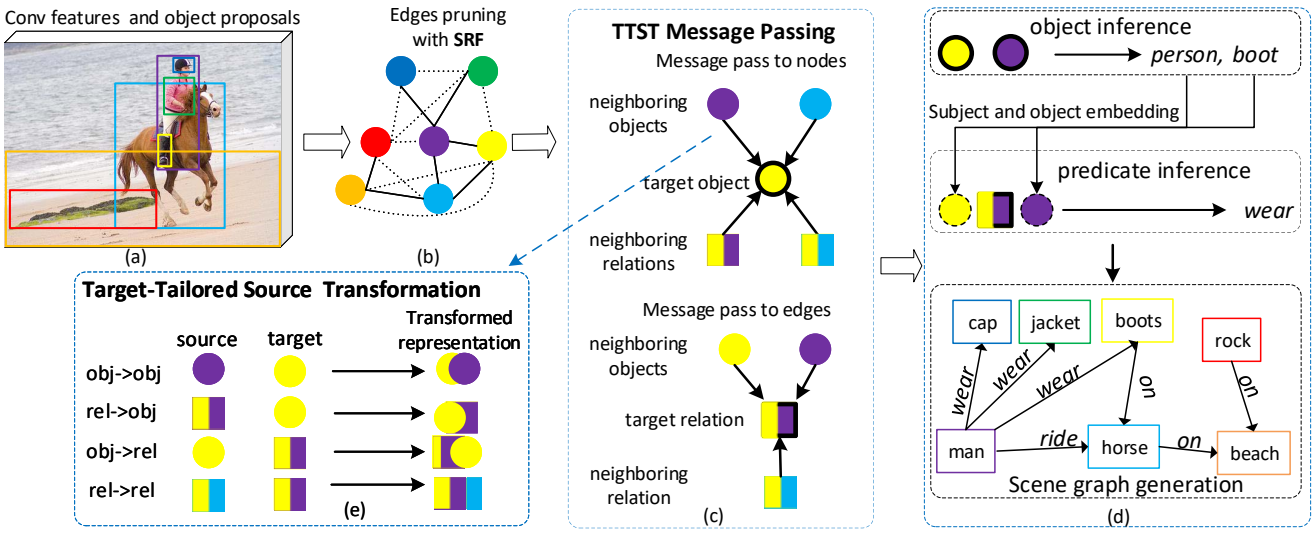

Figure 2. The pipeline of our framework. Given an image, (a) Faster R-CNN is implemented to propose object candidates and extract visual features, (b) then our semantic relation filter (SRF) prunes the connection between a pair of objects that are semantically weakly dependent (pointed lines). (c) Target-tailored source-transformation (depicted in (e)) is applied to learn context from connected nodes and edges across the graph. (4) After predicting the objects using the refined object features, their predicted label information are embedded to serve as subject or object in a relationship for relationships inferring. Finally, scene graph is generated. The colors indicate different objects. Circles denote objects and rectangles denote relationships.

a common domain by training a *shared transformation matrix*. Nevertheless, all existing transformation methods do not take the target into account. Consequently, to any target, the source contributes identical information. For instance, "horse" contributes the same content to "human" and "grass" after the transformation, even though an attention mechanism is used to weight the contribution. However, intuitively, the transformed content should be dependent on both the target and source. Our TTST for message passing is essentially different from previous works by considering the source objects and target object simultaneously. By doing so, for a different target object, the source object contributes different information, and thus the learned representation of target object is more powerful.

3. Proposed Approach

An overview of our proposed model is depicted in Fig. 2. Our goal is to infer a scene graph G for a given image I, which summaries the objects O as nodes and relations R between every two objects as edges. The inferring process can be formally defined as:

$$P(G|I) = P(B|I)P(O|B,I)P(R|B,O_s,O_o,I) \quad (1)$$

where B are locations of objects, O_s, R, O_o stand for subject, relation (predicate), object respectively, and $O_s, O_o \in O$. P denotes the inference probability. $P(B|I)$ can be modeled by an off-the-shelf object detector (Fig. 2(a)). We will discuss each inference module of $P(O|B,I)$ (Fig. 2(c)) and $P(R|B,O_s,O_o,I)$ (Fig. 2 (d)) in the following.

3.1. Object Proposals

Given an image, a set of object proposals O is detected by Faster R-CNN [20] (Fig. 2(a)). Each object $o_i \in O$ is associated with its located region $b_i = [x_i, y_i, w_i, h_i] \in B$, initially predicted label distribution over all C classes $p_i^o \in \mathbb{R}^{1\times C}$, and the pooled visual feature vector x_i^o.

3.2. Semantic Relationship Filter

With n object proposals, there are $\mathcal{O}(n^2)$ edges in the fully connected graph when considering every two objects have a relation (Fig. 2(b)). It has been pointed out in many previous works that most of the object pairs have no relationship due to the real-world regularities of objects interaction (dash edges in Fig. 2(b)). We have also observed that information propagated from the unrelated objects could deteriorate the system's performance because of the possible noise and interfering information. Furthermore, message passing through a fully connected graph is computationally costly and of low efficiency. To make the message passing processes more effective, we propose a *semantic relationship filter* (SRF) to remove the unlikely relationships, similar to what is done in [26].

For o_i, we compute its semantic representation by multiplying its estimated class distribution by the semantic semantic word embeddings matrix $\mathbf{W}_e$:

$$e_i^o = p_i^o \cdot \mathbf{W}_e. \quad (2)$$

Each entry in $\mathbf{W}_e$ is an embedding vector for the corresponding object class. It is learned from the region cap-

tion annotation of the VG dataset by adopting Glove [19]. A multi-layer perceptron (MLP) is trained to estimate a semantic relatedness score between o_i and o_j by feeding $[e_i^o, \tilde{b}_i, \tilde{b}_j, e_j^o]$. Here, $[\cdot]$ denotes a concatenation operation and $\tilde{b}_i$ is the normalization of b_i with respect to the union box of (o_i, o_j). Then, the object pairs with the top K relatedness scores which are also larger than an empirical threshold, are retained and denoted as R.

A relationship of (o_i, o_j) is denoted as r_{ij}. The visual feature is extracted from the union box of (o_i, o_j) and the spatial feature is extracted by an MLP from two binary masks, which indicates the places (with 1) of subject and object respectively. Then, an MLP takes the concatenation of the visual feature and the spatial feature as input to fuse to extract the basic representation x_{ij}.

3.3. Target-Tailored Source-Transformation for Message Passing

3.3.1 Message Passing Revisited

The general approach of passing message to nodes i from its neighboring nodes $\mathcal{N}(i)$ at $l+1$ step is defined as:

$$z_i^{l+1} = \sigma(z_i^l + \Sigma_{j \in \mathcal{N}(i)} a_{ij} W z_j^l), \tag{3}$$

where a_{ij} is the weight for node j and is computed using attention mechanism typically. W is a shared learned transformation matrix which is used to project the representation of source objects to a common domain. $\sigma(\cdot)$ is a nonlinear operation. After several iterations, a representation with a high-order context is obtained and forwarded to the subsequent inference module. Wz_j contributes the same information to any target z_i. Ideally, the transformation should consider the semantic dependency between the target z_i and its source $\mathcal{N}(i)$. To this end, we propose the *target-tailored source-transformation* (TTST) for message passing to explore context through the graph, which is depicted in Fig. 2(c)(e) and discussed in the following.

3.3.2 TTST for Objects

To learn the context of objects and relationships at different semantic levels, messages are passed from both the neighboring objects $N^o(i)$ and relationships $N^r(i)$ to the target object. This message passing is formulate as:

$$\begin{aligned} \hat{x}_i = \sigma(x_i + & \frac{1}{|\mathcal{N}^o(i)|} \sum_{j \in \mathcal{N}^o(i)} f^{(o \to o)}([x_i, e_i], [x_j, e_j]) \\ & + \frac{1}{|\mathcal{N}^r(i)|} \sum_{j \in \mathcal{N}^r(i)} f^{(r \to o)}(x_i, x_{ij})). \end{aligned} \tag{4}$$

Note that, the superscript l is removed for simplicity. The superscript o and r represent object and relationship respectively. $f^{(\to)}(target, source)$ is our TTST operation and the arrow indicates the message passing direction. It is worth noting that e_i is computed by Eq. (2) which contains language prior. It is concatenated with the visual feature x_i as complete representation of object i. Therefore, $f^{(o \to o)}$ broadcasts the visual information as well as the language prior among object nodes. Consequently, both the visual context and the language prior between objects are learned and integrated into the refined representations of target objects. Because the transformation $f^{(\to)}(\cdot)$ "sees" the target and object simultaneously, it is target-tailored source-transformation. Moreover, the transformation is further better guided by the implicit language prior in e_i between different classes of objects. The ablation studies in Sec. 4.2 will show how the language prior affects the performance.

3.3.3 TTST for Relationships.

TTST is also applied to capture context for relationships from its neighboring objects (*i.e.*, the subject and object) and neighboring relationships $N^r(i, j)$ as follows:

$$\begin{aligned} \hat{x}_{ij} = \sigma(x_{ij} + & \frac{1}{2} \Sigma_{m \in [i,j]} f^{(o \to r)}(x_{ij}, x_m) \\ & + \frac{1}{|\mathcal{N}^r(i,j)|} \Sigma_{x_{nm} \in \mathcal{N}^r(i,j)} f^{(r \to r)}(x_{ij}, x_{nm})). \end{aligned} \tag{5}$$

$N^r(i, j)$ is defined as the set of relationships in which each relationship involves either o_i or o_j. After the first iteration in Eq. (4), x_i and x_j contain context of the language prior. Consequently, $f^{(o \to r)}(\cdot)$ also integrates context of the language prior to the relationship representation.

Each transformation $f(\cdot)$ is a separately learned MLP (two fully connected (FC) layers followed by a Relu operation). Each of them is responsible for passing messages in different directions and capturing different levels of context.

3.4. Inference

The inference module is depicted in Fig. 2(d). An object classifier is trained to predict the label distribution $\hat{p}_i^o$ of object proposal i using $\hat{x}_i$. Thus, $P(O|B, I)$ in Eq. (1) is achieved. To infer the graph edge label (*i.e.* relation class), we semantically embed $\hat{p}_i^o$ and $\hat{p}_j^o$ to further explore context information of (subject, relation, object).

$$e^{sub} = \hat{p}_{sub}^o \cdot W_{emb}^{sub}, \quad e^{obj} = \hat{p}_{obj}^o \cdot W_{emb}^{obj}, \tag{6}$$

where W^{sub} and W^{obj} denote the trainable embedding matrix of subject and object respectively. Then, the relationship is semantically represented as $\tilde{x}_{ij} = [e^{sub}, \hat{x}_{ij}, e^{obj}]$. Different from most of the previous works which simply combine the visual features or predicted label distribution of subject and object with the features of relationship, we further explore their context information. Finally, an MLP (consisting of two FC layers followed by a Relu and softmax operation sequentially) is trained to predict the relation

Table 1. Performance comparison with state-of-the-art on VG test set [24]. All numbers in %. We use the same object detection backbone provided by [30] for fair comparison. Because MSDN, FacNet and DRNet use their own data split, the comparison is for reference only. The results of VRD are taken from [24] which reimplemented VRD on VG dataset. The results of Graph R-CNN, KERN and Mem are taken from the original papers.

	Method	SGGen			SGCls			PredCls		
		R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100
[24] split	VRD [17]		0.3	0.5		11.8	14.1		27.9	35.0
	IMP [24]	14.6	20.7	24.6	31.7	34.6	35.4	52.7	59.3	61.3
	Graph R-CNN [26]	-	11.4	13.7	-	29.6	31.6	-	54.2	59.1
	Mem [22]	7.7	11.4	13.9	23.3	27.8	29.5	42.1	53.2	57.9
	KERN [2]	-	27.1	29.8	-	36.7	37.4	-	**65.8**	**67.6**
	MotifNet [30]	21.4	27.2	30.3	32.9	35.8	36.5	58.5	65.2	67.1
	MotifNet-Freq	20.1	26.2	30.1	29.3	32.3	34.0	53.6	60.6	62.2
	TTST (Ours)	**23.8**	**32.3**	**35.4**	**35.1**	**38.6**	**39.7**	**60.3**	64.2	66.4
other splits	MSDN [13]	-	11.7	14.0	-	20.9	24.0	-	42.3	48.2
	DRNet [3]	-	20.8	33.8	-	23.9	27.6	-	-	-
	FacNet [12]	-	13.1	16.5	-	22.8	28.6	-	-	-

class distribution using $\tilde{x}_{ij}$. Now, $P(R|B, O_s, O_o, I)$ in Eq. (1) is achieved. The labels of objects and relations that maximize Eq. (1) are selected.

4. Experiments

In this section we firstly clarify the experimental settings and implementation details. Then, we show quantitative and qualitative results on VG dataset in terms of scene graph generation. We compare our results with strong prior works. We conduct extensive ablation study on each module of our framework and discuss their effectiveness.

Datasets. The Visual Genome dataset [9] is the largest and most popular benchmark dataset for the task of scene graph generation. However, different works use different data splits. For a fair comparison, we adopted the most widely adopted ddataset split in [24]. In the data split, the most-frequent 150 object categories and 50 predicate types are selected. The dataset is split into training set with 75651 images and test set with 32422 images.

Implementation Details. Faster R-CNN [20] with VGG16 [21] as backbone is implemented as our underlying detector and basic visual feature extractor. The codebase is provided by [30]. The input images are scaled and then zero-padded to the size of 592×592. ROI-pooling [5] is applied to extract features of nodes and edge from the basic shared feature maps. In the SRF module, the embedding matrix W_e is initialized with the 300-D Word2vec provided by [17], and a two-layer MLP is trained to output a 1-D vector which then goes through a sigmoid function to squash the predicted score in $(0, 1)$. SRF retains at most 128 relationship proposals with threshold empirically set to 0.55 by considering the trade-off between high recall and accuracy of correct relationships. Each feature transformation $f(\cdot)$ in TTST is an MLP which consists of two FC layers (each followed by Relu operation) and outputs 512-D feature vectors for objects and 4096-D feature vectors for relationships, respectively. The embedding matrices $W_{emb}^{sub}, W_{emb}^{obj} \in \mathbb{R}^{50\times300}$ are randomly initialized, where each row indicate an object class.

Training. We perform stage-wise training. The object detector and the backbone are firstly fine tuned on VG and then frozen. Then, the following modules are trained with different supervision: SRF module is trained with logistic loss, and the TTST message passing module is trained with the sum of cross entropy for object classification and relation classification. SGD ($lr = 5 \times 10^{-3}$) is applied for optimization with momentum 0.9.

Evaluation. We look into three universal evaluation tasks for scene graph generation. (1) **Predicate classification** (PredCls): given the ground truth bounding boxes and labels of objects, predict edge (relation) labels. (2) **Scene graph classification** (SGCls): given ground truth bounding boxes of objects, predict node (objects) labels and edge labels. (3) **Scene graph detection** (SGGen): predict boxes, node labels and edge labels given an image. Only when the predicted labels of the subject, relation, and object of a relationship match the ground truth annotation, and the boxes of subject and object have more than 50% IoU with the ground truth ones simultaneously, this detection is counted as correct. The recall@K metrics ($K = [20, 50, 100]$) for relations are used to evaluate the system performance.

4.1. Quantitative Comparisons

The quantitative results from different models are compared in Tab. 1. Our method is compared with recent strong models: MotifNet [30] that learns regularities using RNN,

Table 2. Ablation studies on our model with accuracy in %. **TTST** denotes whether pass message to capture context through the graph using our proposed TTST message passing method. **Language** denotes using the language context in message passing. **PredE** denotes the semantic embedding of subject and object of a relationship as defined in Eq. (6). **SRF** stands for the semantic relationship filter which is trained to prune the spurious edges. The object detection performance (mAP) follows COCO metrics [15].

Model	TTST	Language	PredE	SRF	Detection	SGGen		SGCls		PredCls	
					mAP	R@50	R@100	R@50	R@100	R@50	R@100
1	-	-	-	-	16.6	12.7	15.9	26.6	27.4	52.4	54.1
2	✓	-	-	-	18.5	17.1	19.9	29.7	32.4	58.3	60.4
3	✓	✓	-	-	20.2	24.7	27.1	33.0	35.1	62.0	64.2
4	✓	✓	✓	-	20.4	29.3	33.1	37.3	38.3	**66.5**	**67.7**
5	✓	✓	✓	✓	**20.8**	**32.3**	**35.4**	**38.6**	**39.7**	64.2	66.4

Table 3. Ablation study on how the number of iterations of message passing to update the representation of nodes and edges affects the final performance. These are evaluated on our full model, which includes SRF, TTST, Language and PredE.

IteNr.	Object Detection	SGGen			SGCls			PredCls		
	mAP	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100
1	19.1	19.3	26.5	30.1	29.8	32.3	34.7	57.6	59.8	62.1
2	**20.8**	**23.8**	**32.3**	**35.4**	**35.1**	**38.6**	39.7	60.3	64.2	66.4
3	20.6	23.1	32.2	35.4	34.4	37.1	**40.1**	**60.5**	**65.7**	**67.8**

capturing context by message passing (IMP [24]), MSDN [13], FacNet [12], Graph R-CNN [26], Mem [22]), CRF-like work DRNet [3], VRD [17] which uses language prior, KERN [2] that exploits statistical prior knowledge and the strong frequency baseline MotifNet-Freq [30]. It is worth noting that their basic object detectors are reported to have 20.4% in mAP@0.5, while our basic object detector has 16.6% in mAP@0.5, which means that we do not have advantage in the front-end object detector. Thus, the comparison is not in favor of ours in terms of object detection.

As compared in Tab. 1, our method outperforms other methods on all metrics (a little inferior to MotifNet and KERN for PredCls. We will analyze the reasons in Sec. 4.2). It demonstrates that our method improves scene graph generation significantly. Our method outperforms MotifNet and the strong frequency baseline MotifNet-Freq, which indicates that our model not only learns the co-occurrence statistics of combination (subject, relation, object) from the training data but also explores the context in the given scene. Specifically, our method is superior to FacNet, Graph R-CNN and Mem which attempt to capture context using message passing approaches. It demonstrates that our model is more efficient in message passing than the recent state-of-the-art message passing models for scene graph generation. Compared to VRD which explicitly exploits language prior, our method shows significant improvement. It suggests that, compared to using language prior separately to predict the relationship labels, our model effectively integrates it with the visual context and learns more powerful representation.

4.2. Ablation Studies

Three modules are applied to boost the performance of scene graph generation: SRF, TTST and an embedding operation of subject and object for prediction relationship (PredE). To study how each of them affects the final performance, we perform several ablation experiments and present the results in Tab. 2 and Tab. 4. Due to the limited GPU memory, 128 object pairs are retained for the subsequent message passing. Thus, we define a confidence score for an object pair as the product of the predicted label confidences of subject and object. The object pairs with top 128 confidence scores are selected when SRF is not utilized.

TTST. Model 1 is the baseline which predicts the relationship labels by combining the features of subject, union box, and object. Comparing Model 1 and Model 2, we find that TTST boosts the overall performance significantly. It indicates that TTST captures context cross the graph effectively and learn powerful representations. Such visual context is clearly helpful for understanding the interaction between objects and object detection (1.9% mAP gain).

Language context. Then, we add the language prior into the TTST (Model 3) as described in Eq. (4). It reports further improvement in the final performance. It demonstrates that: 1) language prior helps explore the context among objects and relationships, and 2) TTST effectively integrates language prior with visual context through the message passing rather than only using it as association information as in previous works, *e.g.*, [17, 13].

Embedding. By embedding the predicted class information of subject and object for predicting their relation, the

Table 4. Ablation study of how different message passing directions in the TTST modules affect the performance. "rel-obj" denotes passing message from relationship to object, and the other notation are similar. The full model is implemented.

Model					Detection	SGGen		SGCls		PredCls	
	obj-obj	rel-obj	obj-rel	rel-rel	mAP	R@50	R@100	R@50	R@100	R@50	R@100
0	-	-	-	-	16.6	14.1	18.5	27.7	30.5	54.2	58.4
1	-	-	✓	-	16.7	14.8	19.7	28.7	32.0	59.7	62.8
2	-	-	-	✓	16.6	14.7	19.6	27.9	31.4	58.2	61.3
3	-	-	✓	✓	16.7	15.1	20.2	29.0	32.3	60.9	63.4
4	✓	-	-	-	19.8	25.5	28.1	32.5	36.8	55.8	59.7
5	-	✓	-	-	16.8	14.2	18.5	28.0	31.1	55.5	59.5
6	✓	✓	-	-	20.2	26.4	28.6	33.8	37.7	56.1	59.8
7	✓	✓	✓	✓	**20.8**	**32.3**	**35.4**	**38.6**	**39.7**	**64.2**	**66.4**

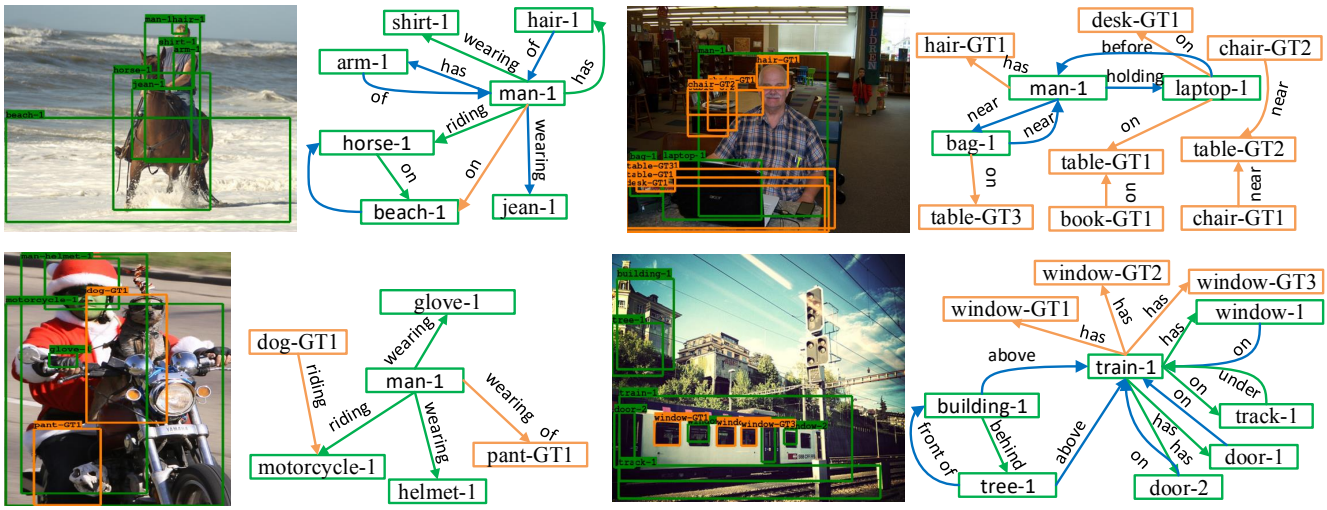

Figure 3. Qualitative results from our model in the scene graph generation setting. Green boxes denote the correctly detected objects while orange boxes denote the ground truth objects that are not detected. Green edges correspond to the correctly recognized relationships at the R@20 setting while orange edges denote the ground truth relationships that are not recognized. The blue edges denote the recognized relationships that however do not exist in the ground truth annotations.

performance is further improved, particularly in the PredCls setting (4.5% gain in R@50 and 3.5% gain in R@100). It indicates that to explicitly designate the role of the two objects involved in a relationship into subject and object separately helps learn the co-occurrence of relationship triplet (subject, relation, object). Furthermore, the performance of object detection is also slightly improved (0.2% mAP gain).

SRF. Finally, we apply SRF to prune the spurious edges to get a sparsely connected graph (Model 5). We notice that almost all performances are improved, except that PredCls is a bit inferior to Model 4 because SRF removes some "good" candidates of relationship (w.r.t. the evaluation criteria, *e.g.*, what is "good" for R@50 and R@100 may not be "good" for R@20). This is also the reason why our model is a little inferior to KERN and MotifNet for PredCls in Tab. 1. The improvements for SGGen and SGCls demonstrates that SRF is effective in selecting the object pairs which are likely to have relationships, especially when the object proposals are not good enough provided by the front-end object detector. We also notice that SRF helps get better object detection performance (0.4% mAP gain). We analyze the gain taken by SRF as follows. Even though deep learning technologies enable the network to learn powerful features from reasonable input, the learned features contain noise or interfering information, because of the imperfect model, training strategy, *etc*. If the input is preprocessed in order to remove noise or interfering information, the model is likely to learn better features. In our model, the spurious relations between objects broadcast the interference via message passing through the graph and deteriorate the model learning process. SRF effectively reduce such kind of interference by removing the spurious relations.

Iteration of Message Passing. TTST works in an iterative way to update the representation of nodes and edges, it is necessary to study how different numbers of iterations affect the final performance. Our full model is trained in

different iterations of message passing and reports the results in Tab. 3. We notice that the overall performance increases with more iterations of message passing and most of the performance reaches the best after 2 iterations. After 3 iterations, some performances drop, especially object detection. But the performance in the PredCls task setting is still slightly improved. We analyze the reason as follows.

The context is captured by passing message to the neighbors via TTST. In one iteration the message is broadcast to its neighboring nodes. More iterations will broadcast the message to further nodes (edges) and capture wider context. Therefore, overall performance is improved. However, the noise and interfering information are also broadcast through the graph. With more iterations, each node/edge accumulates such harmful information in parallel with collecting context from others. Normally, a graph for an image is not large and information will go through the graph within 2 steps starting from any node (see Fig. 3). Thus, the context is already extracted sufficiently in two iterations and harmful information keeps accumulating with more iterations. Consequently, the model performance becomes worse with the deteriorated performance in object detection. Other existing works which pass messages iteratively also reported similar problem [13, 16, 12]. However, slight gains are obtained for PredCls which isolates the performance of object detection. It is because the relation representations are more complex and difficult than that of an individual object. More iterations will help refine the relation representations. The weaker object detection decreases the performance of SGGen and SGCls. Based on this study, we use 2 iterations in our final scheme.

Message Passing Direction As formulated in Eq. (4)(5), message are propagated in four directions in TTST modules: obj-obj, rel-obj, obj-rel and rel-rel. We evaluated how each of the message passing direction affects the performance of our model. The results presented in Tab. 4 shows that any direction of message passing improves the performance of the framework (compared with Model 0) and the full message passing model has the best performance (Model 7). By comparing Model 1-3 with Model 4-6 correspondingly, we notice that passing message to objects (Eq. (4)) improves the performance of object detection (mAP is improved from 16.6% to 20.2%). Consequently, the performance of SGGen (from 14.1% to 26.4%) and SGCls (from 27.2% to 33.8%) are improved significantly. When information is propagated to relationships (Eq. (5)), the performance of PredCls is improved from 54.2% to 60.9% (12.4% relative gain). Consequently, the overall performance is improved. Compared with those improvements from *-obj and *-rel separately, the full model shows further overall improvement. It demonstrates that the TTST modules learn the context by propagating information among objects and relationships effectively and benefit the mutual promotion of object detection and relationship detection.

Improvements on Object Detection. As shown in Tab. 2 and Tab. 4, TTST modules not only improve the performance of visual relationship detection but also the performance of object detection, which is one of the most important tasks for visual scene understanding and critically affects the overall performance of scene graph generation. We achieve the goal of mutual promotion of visual relationship detection and objects detection.

4.3. Qualitative Results

Fig. 3 shows scene graphs generated by our model from the test set. We can see that our model is able to infer relationships between object pairs correctly (green edges) and generate high-quality scene graphs. Some true relationships that are not annotated in the ground truth also can be inferred correctly (blue edges), *e.g.*"man-wearing-jeans" in the first image. It implies that our model works even better than what the quantitative results demonstrate because the unannotated but correctly predicted relationships would deteriorate the performance under current evaluation metrics.

From the examples, we notice that when the detector fails, all the inference of edges to the object will be false, and this situation often occurs when detecting small objects. For example in the right image of the first row, many small or occluded objects are not correctly detected (orange boxes) and all edges connecting them are not recognized correctly. Another common failure case is caused by the ambiguity of relation types, *e.g.* "wear" vs. "wearing".

5. Conclusion

This paper proposes a novel and effective target-tailored source-transformation (TTST) for message passing to generate scene graph. Our model includes a SRF that effectively prunes the spurious connections between objects, and TTST modules that learn context by simultaneously "seeing" the target and source objects. Language prior is used to help message passing and integrated with visual context to learn powerful representations. The experimental results show that our method significantly outperforms the state-of-the-art methods for scene graph generation and meanwhile the performance of object detection is improved. The extensive ablation studies demonstrate the contribution of each proposed module to the framework.

Acknowledgment

This work was supported by the Center for Digital Innovations (ZDIN), Federal Ministry of Education and Research (BMBF), Germany under the project LeibnizKILabor(grant no.01DD20003) and the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122).

References

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

[2] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, pages 6163–6171, 2019. 2, 5, 6

[3] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *CVPR*, pages 3076–3086, 2017. 1, 2, 5, 6

[4] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *CVPR*, pages 1271–1278, 2009. 2

[5] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 5

[6] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, pages 3588–3597, 2018. 2

[7] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 1988–1997, 2017. 1

[8] 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. 1, 2

[9] 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*, 123(1):32–73, 2017. 1, 5

[10] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip HS Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, pages 239–253. Springer, 2010. 2

[11] Yikang Li, Wanli Ouyang, and Xiaogang Wang. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, pages 1347–1356, 2017. 1, 2

[12] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *ECCV*, pages 346–363. Springer, 2018. 2, 5, 6, 8

[13] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, pages 1261–1270, 2017. 1, 2, 5, 6, 8

[14] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *ICCV*, pages 848–857, 2017. 2

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6

[16] Yong Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *CVPR*, pages 6985–6994, 2018. 2, 8

[17] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, pages 852–869, 2016. 1, 2, 5, 6

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv*, 2013. 2

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014. 4

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 3, 5

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 5

[22] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In *CVPR*, pages 8188–8197, 2019. 2, 5, 6

[23] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. In *NIPS*, pages 560–570, 2018. 2

[24] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 5410–5419, 2017. 1, 2, 5, 6

[25] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015. 2

[26] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, pages 690–706, 2018. 2, 3, 5, 6

[27] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, pages 10685–10694, 2019. 1

[28] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, pages 17–24, 2010. 2

[29] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *ICCV*, pages 1974–1982, 2017. 1, 2

[30] 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, 5, 6

[31] 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

[32] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 1

[33] Bohan Zhuang, Lingqiao Liu, Chunhua Shen, and Ian Reid. Towards context-aware interaction recognition for visual relationship detection. In *ICCV*, pages 589–598, 2017. 2