Scene Graph Generation from Objects, Phrases and Region Captions

Yikang Li[1], Wanli Ouyang[1,2], Bolei Zhou[3], Kun Wang[1], Xiaogang Wang[1]

[1]The Chinese University of Hong Kong, Hong Kong SAR, China

[2]University of Sydney, Australia   [3]Massachusetts Institute of Technology, USA

Abstract

Object detection, scene graph generation and region captioning, which are three scene understanding tasks at different semantic levels, are tied together: scene graphs are generated on top of objects detected in an image with their pairwise relationship predicted, while region captioning gives a language description of the objects, their attributes, relations and other context information. In this work, to leverage the mutual connections across semantic levels, we propose a novel neural network model, termed as Multi-level Scene Description Network (denoted as MSDN), to solve the three vision tasks jointly in an end-to-end manner. Object, phrase, and caption regions are first aligned with a dynamic graph based on their spatial and semantic connections. Then a feature refining structure is used to pass messages across the three levels of semantic tasks through the graph. We benchmark the learned model on three tasks, and show the joint learning across three tasks with our proposed method can bring mutual improvements over previous models. Particularly, on the scene graph generation task, our proposed method outperforms the state-of-art method with more than 3% margin. Code has been made publicly available[1].

1. Introduction

Understanding visual scenes is one of the primal goals of computer vision. Visual scene understanding includes numerous vision tasks at several semantic levels, including detecting and recognizing objects, estimating the pairwise visual relations of the detected objects, and describing image regions with free-form sentences. In recent years, great progress has been made to build intelligent visual recognition systems for the three vision tasks, object detection [33, 34, 30], scene graph generation [31, 40, 28, 45], and image/region captioning [41, 10, 22].

The three vision tasks target on different semantic levels of scene understanding. Take the image in Fig.1 as an example. Object detection focuses on detecting in-
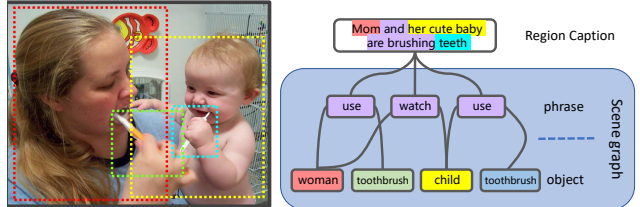
Figure 1. Image with annotations of different semantic levels: objects, phrases and region captions. Scene graph is generated using all objects and their relationships in the image.

dividual objects such as woman, toothbrush, and child. Scene graph generation recognizes not only the objects but also their relationships. Such relationships can be represented by directed edges, which connect two objects as a ⟨*subject-predicate-object*⟩ phrase, like ⟨*woman-use-toothbrush*⟩. Region captioning generates a free-form sentence involving uncertain number of the objects, their attributes, and their interactions, as shown in Fig.1. We can see that though there are connections among the three tasks, the weak alignment across different tasks makes it difficult to learn a model jointly. Our work explores the possibility in understanding the image from these three levels together through a single neural network model.

The key to connect these three tasks is to leverage the spatial and semantic correlations of their visual features. For example in Fig. 1, the phrase ⟨*woman-watch-child*⟩ provides the constraint that two persons are interacting with each other. This constraint validates the existence of the woman and child. In addition, the region caption *'mom and her cute babies are brushing their teeth'* provides constraints on the existence of the objects (*woman*, *child*, and *toothbrush*), their attributes (*cute*), and their relationships (the woman is *watching* the child and they are *using* toothbrush) within this area. Therefore, the features for these three tasks are highly correlated and can be the complementary information of each other. Based on this observation, we propose to jointly refine the features of different semantic levels by introducing a novel framework to align the three tasks and a message passing structure to leverage

the complementary effects for mutual improvements.

In this work, we propose an end-to-end Multi-level Scene Description Network (MSDN) to simultaneously detect objects, recognize their relationships and predict captions at salient image regions. This model effectively leverages the rich annotations at three semantic levels and their connections for image understanding.

The contributions of this paper are summarized as follows: 1) We propose a novel model to learn features of different semantic levels by simultaneously solving three vision tasks, object detection, scene graph generation and region captioning. 2) In the model, given an image, a graph is built to align the object, phrase, and caption regions within an image. Since images have different objects, phrases and captions, constructed graphs could be different for different images. We provide a dynamic graph construction layer in the CNN to construct such a graph. 3) A feature refining structure is used to pass message from different semantic levels through the graph. In this way, the three tasks are integrated into one single model, and the features of three semantic levels are jointly optimized. On the Visual Genome dataset [23], our proposed model outperforms the state-of-art methods on scene graph generation by 3.63%∼4.31%. The mutual improvement effects are also shown on the object detection and region captioning tasks. Code has been made publicly available to facilitate further research.

2. Related Work

Object Detection: Object detection is the foundation of image understanding. Objects serve as bricks to build up the house of the scene graph. Since CNNs were firstly introduced to the object detection by Girshick *et al.* in R-CNN [15], many region-based object detection algorithms, such as Fast R-CNN [14], SPP-Net [18], Faster R-CNN [34], were proposed to improve the accuracy and speed. Although YOLO [33] and SSD [30] further sped up the detection process by sharing more layers between region proposal and region recognition, Faster R-CNN [34] is still a popular choice for object detection because of its excellent performance. Therefore, we will adopt the pipeline of Faster R-CNN as the basis of our proposed model.

Visual Relationship Detection: Visual Relationship detection is not a new concept. It has been investigated by numerous studies in the last decade. In the early days, most works targeted specific types of phrases [6, 9] or used visual phrases to improve other tasks [36, 16, 25, 35]. Recently, researchers pay more attention to general visual relationship detection [28, 40, 32, 44, 8, 50, 51] . Lu *et al.* utilized the language prior in detecting visual phrases and their components in [31]. Li *et al.* used the message passing structure among subject, object and predicate branches to model their dependencies [28]. Xu *et al.* built up a fully-connected graph to iteratively pass messages along

the scene graph [40]. Liang *et al.* applied the reinforcement learning method to the relationship and attribute detection [29]. However, connections between phrases and captions are not built up in existing works. In this paper, we will view the objects, phrases and region captions as different semantic levels and build up their connections based on their spatial and semantic relationships.

Image Captioning: Recently, increasingly more researchers put their attentions on interactions bwtween vision and language [27, 49, 1, 43, 7], of which, image captioning is a fantastic research topic that connects the two areas. It has been investigated for years [3, 13, 19, 24, 26, 39]. Recently, CNN plus RNN has been adopted as the standard pipeline for image captioning task [5, 11, 12, 22, 41]. Captioning was based on the whole image until the work of Johnson *et al.* [20] introduced the dense captioning task which focuses on the regions. Existing works on image/region captioning, however, do not explicitly leverage the scene graph. Our proposed model integrates the highly-structured scene graph into our model to learn better feature for region captioning. And in return, the captioning task can also provide additional information for scene graph generation.

Multi-task Learning: Multi-task learning [46, 42, 48, 21, 47] has been used to model the relationships among correlated tasks. In [46], a convex formulation was derived for multi-task learning. A group of related tasks was identified using statistical models in [42, 48]. Multi-task deep learning is used for learning facial key point detection aided by face attributes [48]. Group sparsity is used in [21] to determine a group of tasks that will share feature representations. Our work propose a novel way to leverage the complementary effects from three tasks of different semantic levels.

3. Multi-level Scene Description Network

An overview of our proposed MSDN is shown in Figure 2. It adopts the region-based detection pipeline in [34]. The model contains three parallel branches for three different vision tasks. MSDN is based on the convolutional layers of VGG-16 [37], which is shared by the region proposal network (RPN) and recognition network.

The entire process can be summarized as below: 1) Region proposal. To generate ROIs for objects, phases and, region captions. 2) Feature specialization. Given ROIs, to obtain specialized features that will be used for different semantic tasks. 3) Dynamic graph construction. Dynamically construct a graph to model the connections among feature nodes of different branches based on the semantic and spatial relationships of corresponding ROIs. 4) Feature refining. To jointly refine the features for different tasks by passing messages of different semantic levels along the graph. 5) Final prediction. Using the refined features to classify objects, predicates and generate captions. The scene graph
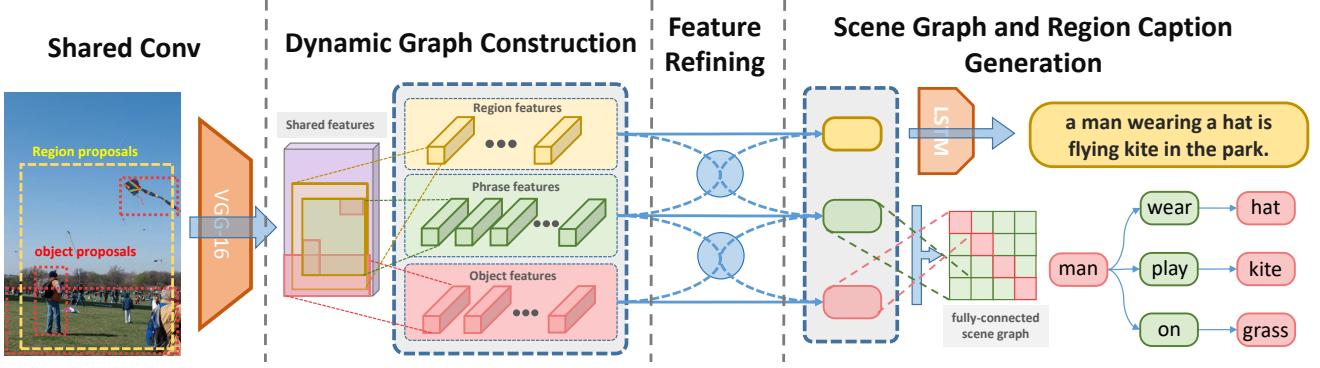
Figure 2. Overview of MSDN. The two RPNs [34] for object and caption regions are omitted for simplicity, which share the convolutional layers with other parts. Phrase regions are generated by grouping object regions into pairs. With the region proposals for objects, phrases, and captions, ROI-pooling is used for obtaining their features. These features go through two fully connected layers and then pass messages to each other. After message passing, features for objects are used for object detection, similarly for phrase detection and region captioning. Message passing is guided by the dynamic graph constructed from the object and caption region proposals. Features, bounding boxes and predicted labels for object (red), phrase (green) and region (yellow) are assigned with different colors.

is generated from detected objects and their recognized relationships (predicate).

3.1. Region Proposal

Three sets of proposals are generated:

- object region proposals: directly generated using Region Proposal Network (RPN) proposed in [34];

- phrase region proposals: grouping the N object proposals to $N(N-1)$ object pairs (two identical proposals will not be grouped) which fully connects object proposals with directed edges;

- caption region proposals: directly generated by another RPN trained with ground truth region bounding boxes.

RPNs for object and caption region proposals share the base convolutional layers of VGG-16 [38]. The anchors of two RPNs are generated by clustering the logarithmic widths and heights of ground truth boxes the training set using k-means clustering [17]. To reduce the size of ROI sets, non-maximum suppression is used for object and caption ROIs separately.

3.2. Feature Specialization

Different branches correspond to different vision tasks. To make different branches learn their own features, we first feed the three sets of ROIs to ROI-pooling and then use different FC layer sets for different branches. In our implementation, we use two 1024-dim FC layers for each branch. After feature specialization, each branch has its own features for its specific task.
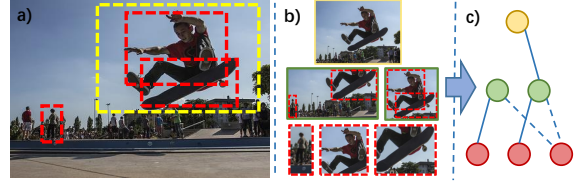

Figure 3. Dynamical graph construction. (a) the input image. (b) object(bottom), phrase(middle) and caption region(top) proposals. (c) The graph modeling connections between proposals. Some of the phrase boxes are omitted.

3.3. Dynamic Graph Construction

For different input images, the topology structures of the connections are different. Thus, the connection graph is dynamically built up based on the semantic and spatial relationships among the ROIs.

Connections between phrases and objects are naturally built during constructing phrase proposals. Each phrase proposal will be connected to two object proposals as a *subject-predicate-object* triplet with two directed edges. The connection between phrase and caption proposals is obtained based on their spatial relationship. When a caption proposal, denoted by $b^{(r)}$, covers enough fraction (the threshold 0.7 is used in our experiment) of a phrase proposal, denoted by $b^{(p)}$, there is an undirected edge between $b^{(r)}$ and $b^{(p)}$. We ignore the direct connection between captions and objects for simplicity as they can be connected indirectly through the phrase level.

From the steps above, a graph is constructed to model the connections among objects, phrases and caption proposals. Fig. 3 shows an example of this graph.

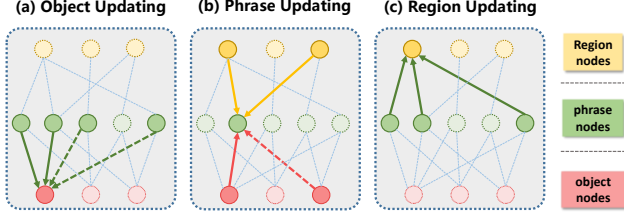The graph G, contains a node set V and an edge set

(a) Object Updating **(b) Phrase Updating** **(c) Region Updating**

Region
nodes

phrase
nodes

object
nodes

Figure 4. Feature refining for object nodes **(a)**, phrase nodes **(b)** and caption nodes **(c)**. The arrow means passing direction. The two kinds of lines connected to the object nodes are used to distinguish the *subject-predicate* and *predicate-object* connections.

E. For V, each node in V corresponds to the specialized features of an ROI. The edge set E contains a set of the undirected edges between caption and phrase, $E_{p,r}$, and two directed edge set, $E_{s,p}$ and $E_{o,p}$, where s and o denotes the *subject* and *object* in the phrase. In the following sections, we will use the denotations for simplicity.

3.4. Feature Refining

After determining the connections between different levels of nodes, message is passed among features through the edges of the graph. We divide the feature refining procedure into three parallel steps, object refining, phrase refining and caption refining (Figure 4). In addition, the refining procedure can be applied iteratively.

We will analyze the message passing from phrase nodes to object nodes as an example. And it can be extended to message passing between other types of nodes.

3.4.1   Refining Features of Objects

For each object node, there will be two kinds of connections, *subject-predicate* and *predicate-object*. We merge the phrase features into two sets according to the connection type and then refine the object feature with the merged phrase features.

Phrase feature merge. Since the features from different phrases have different importance factors for refining objects, we use a gate function to determine weights. The features from multiple phrases are averaged by the gate as follows (we use *subject-predicate* as an example):

$$\tilde{x}_i^{(p\to s)} = \frac{1}{\|E_{i,p}\|} \sum_{(i,j)\in E_{s,p}} \sigma_{\langle o,p\rangle}\left(x_i^{(o)}, x_j^{(p)}\right) x_j^{(p)} \quad (1)$$

where $\tilde{x}_i^{(p\to s)}$ denotes the average of gated features from the phrase that connects the object by the *subject-predicate* connections with the i-th object node. $E_{s,p}$ is the set of *subject-predicate* connections and $\|E_{i,p}\|$ denotes the number of phrases connected with the i-th object as the $\langle subject - predicate\rangle$ pairs. $\sigma_{\langle o,p\rangle}$ denotes the gate function for the object-phrase connections which is controlled

by the source and target features:

$$\sigma_{\langle o,p\rangle}\left(x_i^{(o)}, x_j^{(p)}\right) = \sum_{g=1}^{G} \text{sigmoid}\left(w_{\langle o,p\rangle}^{(g)} \cdot \left[x_i^{(o)}, x_j^{(p)}\right]\right),$$

$$(2)$$

where G denotes the number of the gate templates for the input features, and we use 128 in our experiment. Each g of $w^{(g)}$ corresponds to a template. When the input feature matches the template, the value after sigmoid will be 1, and the gate will open. The weights $w_{\langle o,p\rangle}^{(g)}$ are learned. Similar to the procedure in (1), we can obtain the merged features $\tilde{x}_i^{(p\to o)}$ for the *predicate-object* connections.

Object feature refining. For the i-th object, there are two merged features, $\tilde{x}_i^{(p\to s)}$ and $\tilde{x}_i^{(p\to o)}$. Then refine the i-th object feature as follows:

$$x_{i,t+1}^{(o)} = x_{i,t}^{(o)} + F^{(p\to s)}\left(\tilde{x}_i^{(p\to s)}\right) + F^{(p\to o)}\left(\tilde{x}_i^{(p\to o)}\right)$$

$$(3)$$

where t denotes the refining step since the feature refining can be done iteratively. $F(\cdot) = W \cdot ReLU(\cdot)$, which is implemented by a ReLU followed by an FC layer because all the features in Eq. 3 are pre-ReLU ones. Since the merged features $\tilde{x}_i^{(p\to s)}$ and $\tilde{x}_i^{(p\to o)}$ are in the domain of phrase features, we use additional FC layers, $F^{(p\to s)}$ and $F^{(p\to o)}$, for modality transformation. In addition, the two FC layers do not share parameters.

3.4.2   Refining Features of Visual Phrase and Caption

Each phrase node is connected to two object nodes, which are $subject$ and $object$ in the $\langle subject - predicate - object\rangle$ triplet. And each caption node connects several phrase nodes. Similar to the procedure in refining features of objects, the refinement for phrase and caption also adopt the Merge-and-Refine paradigm:

$$x_{j,t+1}^{(p)} = x_{j,t}^{(p)} + F^{(s\to p)}\left(\tilde{x}_j^{(s\to p)}\right)$$
$$+ F^{(o\to p)}\left(\tilde{x}_j^{(o\to p)}\right) + F^{(r\to p)}\left(\tilde{x}_j^{(r\to p)}\right), \quad (4)$$
$$x_{k,t+1}^{(r)} = x_{k,t}^{(r)} + F^{(p\to r)}\left(\tilde{x}_k^{(p\to r)}\right),$$

where $x_{j,t+1}^{(p)}$ and $x_{j,t+1}^{(r)}$ are respectively the refined phrase features and caption features at time step $t + 1$. $\tilde{x}_j^{(s\to p)}$ and $\tilde{x}_j^{(o\to p)}$ denote the features merged from its subject and object respectively in the *subject-predicate-object* phrase for j-th phrase node, and $\tilde{x}_j^{(r\to p)}$ denotes the feature merged from its connected caption nodes. $\tilde{x}_k^{(p\to r)}$ are the merged feature for the k-th caption node.

With this feature refining structure, messages are passed through the graph to update the features of objects, phrases,

and captions by absorbing supplementary information from the connected nodes.

3.5. Scene Graph Generation

Since the feature refining step has pass message between object and phrase nodes, object and corresponding pair-wise relationship categories are predicted directly based on the features of objects and phrases.

We use a matrix to represent the scene graph, where the element (i, i) at diagonal position is the ith object and the element at the (i, j) position for $i \neq j$ is the phrase representing the relationship between the ith and jth object. For the ith object, it is predicted as an object class or $\langle background \rangle$ from its refined object features. Similarly, the (i, j)th phrase is predicted as a pre-defined predicate class or $\langle irrelavant \rangle$ for subject i and object j from phrase features. Then the scene graph is generated based on the matrix. If the object i and j are not classified as $\langle background \rangle$ and the predicate (i, j) is not $\langle irrelavant \rangle$, then the two objects are connected through the predicate (i, j). In this way, we will get a scene graph based on the matrix.

3.6. Region Caption Generation

Different from the object and phrase nodes, the region features contains a wide range of information, such as objects, their interactions and attributes, scene-related information, *etc*. Therefore, we feed them into an LSTM-based language model to generate natural sentences to describe the region. We adopt the vanilla language model widely used for image captioning [20, 22].

The language model takes the image vector as input and outputs a free-form sentence to describe the content in the region. The model consists of four parts: 1) an image encoder, which is used to transform the image feature to the same domain of word features; 2) a word encoder, to transform the one-hot vector to a word embedding; 3) a two-layer LSTM model, which is to encode the image information and the temporal dependencies within the sequence; 4) a word decoder, which is used to decode the output feature of LSTM to a distribution over words.

At the first time step, image vectors are transformed to the same domain of word vectors by image encoder. Then coded image feature will be fed into a two-layer LSTM. At the second step, the $\langle start \rangle$ token will be fed into the model to indicate the start of the sentence. Then the predicted word at time t will be fed into the model as input until the $\langle end \rangle$ or the maximum length is reached.

4. Experiment

Scene graph generation can be viewed as an intermediate task connecting the object detection and region captioning, which aims at capturing the structural information of an image with a set of pair-wise relationships. Compared to object detection, the scene graph generation measures the feature learning from more aspects. And different from the region captioning, the performance of the scene graph generation model is easier to measure quantitatively and it excludes the influence brought by the different language model implementations. Therefore, the experiment part mainly focuses on this task.

Some explanatory experiments are also done on the object detection and region captioning tasks to show mutual improvements brought by the joint inference across semantic levels.

4.1. Dataset

All the experiments are done on the Visual Genome [23] dataset. The objects and relationships are from the Relationship subset, and the region caption annotations are based on the Region Description subset. The two subsets share the image but target on different tasks.

First, we do some preprocessing on the relationship annotations. We normalize the words in different tenses and then select the top-150 frequent object categories and top-50 predicate categories. Moreover, the object boxes whose shorter edges are smaller than 16 pixels are removed. After preprocessing, there are 95998 images left.

For the remaining 95998 images, we further pre-process the region caption annotations. All the words are changed to lower case. Top-10000 frequent words (including punctuations) are used to build up the dictionary and all the other words are changed to $\langle unknown \rangle$ token. In addition, all the small regions with shorter edges smaller than 32 are removed. NLTK [4] is used to tokenize the sentence.

After the two preprocessing steps above, a cleansed dataset containing the annotations of localized objects, phrases and region descriptions are built for our experiments. From the 95998 images in the dataset, 25000 images are sampled as the testing set and the remaining 70998 images are used as the training set.

4.2. Implementation Details

Model training details Our model is initialized on the ImageNet pretrained VGG-16 network [38]. To reduce the number of parameters, we only use 1024 neurons of the fully-connected layers from the original 4096 ones and then scale up the weights accordingly as initialization. The newly introduced parameters are randomly initialized. We first train RPNs and then jointly train the entire model from the base learning rate 0.01 using SGD with gradients clipping. The parameters of VGG convolutional layers are fixed at first, and then trained with 0.1 times the learning rates of other layers after the first decay of the base learning rate. In addition, there is no weight decay for the language model and the parameters are updated using Adam.

ID	Message Passing	Cap. branch	Cap. Supervision	FR-iters	PredCls		PhrCls		SGGen	
					Rec@50	Rec@100	Rec@50	Rec@100	Rec@50	Rec@100
1	-	-	-	0	49.28	52.69	7.31	10.48	2.39	3.82
2	✓	-	-	1	63.12	66.41	19.30	21.82	7.73	10.51
3	✓	✓	-	1	63.82	67.23	20.91	23.09	8.20	11.35
4	✓	✓	✓	1	66.70	**71.02**	23.42	25.68	10.23	13.89
5	✓	✓	✓	2	**67.03**	71.01	**24.22**	**26.50**	**10.72**	**14.22**
6	✓	✓	✓	3	66.23	70.43	23.16	25.28	10.01	13.62

Table 1. Ablation studies of the proposed model. **PredCls** denotes predicate recognition task. **PhrCls** denotes phrase recognition task. **SGGen** denotes the scene graph generation task. **Message passing** denotes whether to add feature refining structure to pass message. **Cap. branch** denotes whether to use the caption branch as an extra connection source. **Cap. Supervision** indicates whether to use region caption annotation as the supervision to guide the learning of the caption branch. **FR-iters** denotes the number of feature refining iterations.

Loss Functions For the object branch, we use the cross-entropy loss for the object classification and the smooth L1 loss for the box regression. For the phrase branch, the cross-entropy loss is used for predicting the labels of predicates. For the caption branch, the cross-entropy loss is used for generating the every word of free-form sentences and the smooth L1 loss is used for regressing the corresponding proposals. Three losses are summed up equally. Since every step at feature refining parts is differentiable, BP can be applied for the feature refining part.

Mini-batch preparation for training A mini-batch contains one image. After generating proposals with RPN layers, we use 0.7 and 0.75 as the NMS threshold for object proposals and caption proposals respectively and keep at most 2000 boxes after NMS. Then we sample 256 object proposals and 128 caption proposals from each image. As the number of phrase proposals is too large and the positive samples are sparse, we sample 512 with 25% positive instances. In addition, we assign $\langle irrelavant \rangle$ to the negative phrase samples, $\langle background \rangle$ to the negative objects, and the $\langle end \rangle$ to the negative caption proposals.

Details for inference. In testing, we set the NMS threshold to 0.35 and 0.45 for object and caption region proposals. After the graph for the image is constructed, features from all the sampled proposals are used for refining their features.

4.3. Evaluation on Scene Graph Generation

4.3.1 Experiment settings

Performance Metric. Following [31], the *Top-K recall* (denoted as *Rec@K*) is used as the main performance metric, which is the fraction of the ground truth instances hit in the top-K predictions. The reason of using *recall* instead of *mean average precision(mAP)* is that the annotations of the relationships are incomplete. *mAP* will falsely penalize the positive but unlabeled predictions. In our experiment, *Rec@50* and *Rec@100* will be reported.

Task Settings. Since scene graph generation involves the classification of the $\langle subject\text{-}predicate\text{-}object \rangle$ triplet and localization of objects. We evaluate our proposed

model on three sub-tasks of scene graph generationz proposed in [40]:

- **Predicate Recognition** (PredCls): To recognize the relationship between the objects given the ground truth location of object bounding boxes. This task is aimed at examining the model performance on the classification of the predicates alone.

- **Phrase Recognition** (PhrCls): To predict the predicate categories as well as the object categories given the ground-truth location of objects. This task evaluates the model performance on the recognition of both predicates and objects.

- **Scene Graph Generation** (SGGen): To detect objects and recognize their pair-wise relationships. The object is correctly detected if it is correctly classified and its overlap with the ground truth bounding box is larger than 0.5. A relationship is correctly detected if both the *subject* and *object* are correctly detected and the *predicate* is correctly predicted. The location of objects is not provided.

4.3.2 Comparison with existing works

We compare our proposed MSDN with the following methods under the three task settings: (1) The model using Language Prior (**LP**) [31], which detects objects first and then estimate the categories of predicate using visual features and word embeddings. (2) Iterative Scene Graph Generation (**ISGG**) [40], which uses the iterative message passing along the scene graph with a GRU-based feature refining scheme. We have reimplemented their model. The model is trained and tested on the cleansed dataset mentioned in Section 4.1. All the methods are based on the VGG-16 model.

From the results in Table 2, we can see that our proposed model performs better than the existing works. Compared to the ISGG model [40], our model introduces the caption branch to provide more context information for phrase

Task		LP [31]	ISGG [40]	Ours
PredCls	R@50	26.67	58.17	**67.03**
	R@100	33.32	62.74	**71.01**
PhrCls	R@50	10.11	18.77	**24.34**
	R@100	12.64	20.23	**26.50**
SGGen	R@50	0.08	7.09	**10.72**
	R@100	0.14	9.91	**14.22**

Table 2. Evaluation on the Visual Genome dataset [23]. We compare our proposed model with existing works on the three tasks illustrated in Sec. 4.3.1. The result of LP is reported in [40]. ISGG is reimplemented by ourselves and evaluated on our cleansed dataset.

Object Det.	FRCNN [34]	Baseline-3-bran.	Ours
mean AP(%)	6.72	6.70	**7.43**
Acc. Top-1(%)	53.57	53.14	**61.12**
Acc. Top-5(%)	83.50	83.25	**89.86**
Region Caption	Baseline	Baseline-3-bran.	Ours
AP [20](%)	4.41	4.28	**5.39**

Table 3. Object detection and region captioning results evaluated on Visual Genome dataset [23]. *Baseline-3-bran.* has 3 separate branches without message passing.

recognition. In addition, our model passes message as residual, which makes the model easier to train.

4.3.3 Component Analysis

There are many components that influence the performance of MSDN. Table 1 shows our investigation on the performance of different settings of MSDN on the Visual Genome dataset [23].

Message passing. Model 1 in Table 1 is the baseline that does not use message passing to refine features and does not have the branch for caption. Model 2 is based on Model 1 and passes message between related object and phrase nodes. By comparing Model 1 and 2 in Table 1, we can see that passing message with the feature refining structure proposed in Sec. 3.4 can help to leverage the connection between the objects and phrases, which significantly improves the model performance by $5.34\% \sim 6.69\%$ on SGGen task.

Caption region branch. Based on Model 2, Model 3 only has an extra caption branch without the caption supervision. We remove the LSTM language model in Fig. 2 and only use the caption branch as extra context information source. Model 3 has $0.47\% \sim 0.84\%$ gain when compared with Model 2. This improvement is more likely to come from the more parameters introduced by the caption branch.

Region Caption Supervision. Model 4 further uses additional supervision of region caption sentences for the region caption branch. It outperforms Model 3 by $2.03\% \sim 2.64\%$. The improvement mainly comes from the complementary features learned with additional information. Supervision helps the caption branch learn it own specialized features, which can provided extra information for other branches. Compared to the object and predicate categories, region captions provide another way to understand the image.

The number of feature refining iterations. Model 4~6 are different in the number of iterations in message passing. By comparing Model 4~6, the results show that two iterations may be the optimal settings for the scene graph generation. Compared to Model 4 with one iteration, Model 5

with two iterations constructs the connection between captions and objects indirectly, which brings $0.33\% \sim 0.49\%$ gain. However, more iterations make the model harder to train. Therefore, when we refine the features for three iterations, the training issue suppress the gain brought by the better feature refining. Therefore, the performance of Model 6 will deteriorate by $0.21\% \sim 0.27\%$.

4.4. Evaluation on Object Detection

We further evaluate our proposed MSDN on object detection task.

Setup. We directly use the objects within the dataset prepared in 4.3.1. All the objects have at least one relationship with other objects. We adopt the mean Average Precision (mAP) metric as one evaluation metric. In addition, as most of the objects are small, poor localization of the objects highly influences the mAP metrics, we also report the accuracy of the object classification with the ground truth bounding boxes given.

We compare our proposed MSDN with Faster R-CNN [34] (**FRCNN**) trained on the same dataset. In addition, to check whether the additional supervision can benefit the feature learning of convolutional layers, we also show the results for the model with the feature refining structure removed (**Baseline-3-bran.**) and use the object branch for object detection (like the model 1 in 1).

Results. Since the Visual Genome Dataset has many object classes that are small and hard to detect, the mAP is small for all approaches. Nevertheless, our model outperforms Faster R-CNN and baseline model with three separated branches on the Visual Genome Dataset, because our model introduces more context information from phrases and captions (when trained with more than two iterations) to the objects as complementary source, which serves as visual cues to help recognize objects.

4.5. Evaluation on Region Captioning

We further evaluate our model on the region caption task.

Setup. We adopt the evaluation metric proposed by Johnson *et al.* in [20] for region captioning. It measures the mean Average Precision across a range of thresholds for both localization and language accuracy. The Meteor scores [2] are used as the language metric, because it is
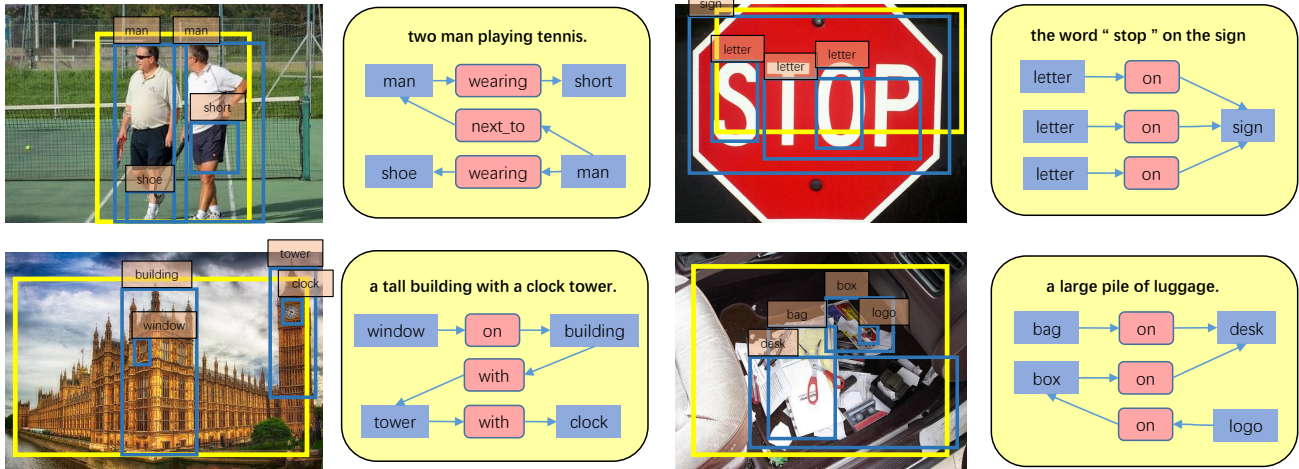
Figure 5. Qualitative results for region captioning. The most salient regions with captions are shown (yellow boxes). We also show several relationships that are connected to the captions. The connection is built by our proposed dynamic graph generation in Sec. 3.3.

highly correlated with human judgments. During the evaluation, the ground truth bounding regions are merged as one region with several reference annotations if they are heavily overlapped with each other (based on IOU with threshold of 0.7).

To make the model comparable, we re-implement the main part of Densecap [20] using Faster R-CNN [34] pipeline based on VGG-Net (**Baseline**) and use the same language model as our proposed model. Our implementation performs comparably with the original Densecap under same settings (4.41% vs 4.62% evaluated on our cleansed dataset). In addition, similar to 4.4, we also include another baseline model with three separated branch without message passing (**Baseline-3-bran.**). All the models are evaluated on our cleansed dataset.

Quantitative Results. From Table. 3, we can see that, our proposed model outperforms the other two baseline models. Because we have excluded the influence brought by the number of parameters and utilized the same language model for them, the gain is obtained by the extra information introduced through the message passing. And the messages passed to the region come from the scene graph composed by the objects and their relationships. Such structural information can help the region branch infer the content of the region. In addition, by comparing the two baseline models, simply introducing extra supervision will not improve the accuracy.

Qualitative Results. Region captioning results with the highest score are shown in Figure 5. We also show the objects and their relationships that are connected to the captions through the dynamic graph. We can see that the region captioning result is highly correlated to the scene graph. We also observe failure case (bottom right in Figure 5, where

the misclassification of objects and relationships would mislead the caption branch to recognize the region as *a large pile of luggage*.

5. Conclusion

This paper targets on scene understanding by jointly modeling three vision tasks, *i.e.* object detection, visual relationship detection and region captioning, with a single deep neural network in an end-to-end manner. The three tasks at different semantic levels are tightly connected. A Multi-level Scene Description Network (MSDN) model is proposed to leverage such connection for better understanding image. In MSDN, given an input image, a graph is dynamically constructed to establish the links among regions with different semantic meaning. The graph provides a novel way to align features from different tasks. Experimental results show that this joint inference process brings improvement in all the three tasks.

Acknowledgment

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 2

[2] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005. 7

[3] K. Barnard, P. Duygulu, D. Forsyth, N. d. Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 2003. 2

[4] S. Bird. Nltk: the natural language toolkit. In *ACL*, 2006. 5

[5] X. Chen and C. L. Zitnick. Learning a recurrent visual representation for image caption generation. *arXiv preprint arXiv:1411.5654*, 2014. 2

[6] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 33–40, 2013. 2

[7] B. Dai, D. Lin, R. Urtasun, and S. Fidler. Towards diverse and natural image descriptions via a conditional gan. *arXiv preprint arXiv:1703.06029*, 2017. 2

[8] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. *CVPR*, 2017. 2

[9] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV*, 2012. 2

[10] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 1

[11] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 2

[12] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 2

[13] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010. 2

[14] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[16] A. Gupta and L. S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, 2008. 2

[17] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1979. 3

[18] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *CVPR*, 2014. 2

[19] Y. Jia, M. Salzmann, and T. Darrell. Learning cross-modality similarity for multinomial data. In *ICCV*, 2011. 2

[20] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *arXiv preprint arXiv:1511.07571*, 2015. 2, 5, 7, 8

[21] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011. 2

[22] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1, 2, 5

[23] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 2, 5, 7

[24] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *TPAMI*, 2013. 2

[25] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *CVPR*, 2010. 2

[26] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Generalizing image captions for image-text parallel corpus. In *ACL*, 2013. 2

[27] S. Li, T. Xiao, H. Li, B. Zhou, D. Yue, and X. Wang. Person search with natural language description. In *CVPR*, 2017. 2

[28] Y. Li, W. Ouyang, X. Wang, and X. Tang. Vip-cnn: Visual phrase guided convolutional neural network. *CVPR*, 2017. 1, 2

[29] X. Liang, L. Lee, and E. P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. *CVPR*, 2017. 2

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 1, 2

[31] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. 1, 2, 6, 7

[32] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik. Phrase localization and visual relationship detection with comprehensive linguistic cues. *arXiv preprint arXiv:1611.06641*, 2016. 2

[33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015. 1, 2

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 3, 7, 8

[35] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2

[36] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 2

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 5

[39] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using un-aligned text corpora. In *CVPR*, 2010. 2

[40] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. *arXiv preprint arXiv:1701.02426*, 2017. 1, 2, 6, 7

[41] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhut-dinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015. 1, 2

[42] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 2007. 2

[43] D. Yu, J. Fu, T. Mei, and Y. Rui. Multi-level attention networks for visual question answering. In *CVPR*, 2017. 2

[44] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017. 2

[45] H. Zhang, Z. Kyaw, J. Yu, and S.-F. Chang. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *ICCV*, 2017. 1

[46] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012. 2

[47] Y. Zhang, D.-Y. Yeung, and Q. Xu. Probabilistic multi-task feature selection. In *NIPS*, 2010. 2

[48] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014. 2

[49] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, 2015. 2

[50] B. Zhuang, L. Liu, C. Shen, and I. Reid. Towards context-aware interaction recognition. *ICCV*, 2017. 2

[51] B. Zhuang, Q. Wu, C. Shen, I. Reid, and A. v. d. Hengel. Care about you: towards large-scale human-centric visual relationship detection. *arXiv preprint arXiv:1705.09892*, 2017. 2