ISBN 9781665428125

(doi: 10.1109/ICCV48922.2021.00144)

This is the Author Accepted Manuscript.

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

There may be differences between this version and the published version. You are advised to consult the publisher’s version if you wish to cite from it.

http://eprints.gla.ac.uk/276556/

Deposited on: 12 August 2022
In Defense of Scene Graphs for Image Captioning

Kien Nguyen*1, Subarna Tripathi*2, Bang Du1, Tanaya Guha3, and Truong Q. Nguyen1

1University of California San Diego, USA
2Intel Labs, USA
3University of Warwick, UK

Abstract

The mainstream image captioning models rely on Convolutional Neural Network (CNN) image features to generate captions via recurrent models. Recently, image scene graphs have been used to augment captioning models so as to leverage their structural semantics, such as object entities, relationships and attributes [32, 30, 6]. The literature however has mixed opinion about the usefulness of scene graphs in captioning. Few works have reported improvements in caption generation using scene graphs [21, 30], while several others have highlighted that scene graphs alone yield poor captioning results and can even harm captioning performance [10, 14]. In this paper, we identify the challenges in effective utilization of scene graphs for image captioning, and subsequently investigate how to best harness them for this task.

Scene graph representation consisting of nodes and edges can be derived from either (i) images where the nodes correspond to the objects present in the scene, termed as Visual Scene Graphs (VSG), or (ii) from a caption where nouns and verbs take on the roles of nodes and edges in a rule-based semantic parsing, termed as Textual Scene Graphs (TSG). The literature of scene graph generation and scene graphs for image captioning primarily refers to the VSG representation.

To be able to leverage scene graphs for captioning, we need paired VSG-caption annotations. This is currently unavailable. Hence, methods requiring explicit scene graphs end up training the VSG generator and the caption generator on disparate datasets [30, 6, 14]. The current practice is to train VSG generators on the Visual Genome (VG) dataset, train TSG to caption generation on COCO-captions dataset, and finally transform the VG-trained VSGs to captions utilizing the later. We note two issues with this approach:

- The VG-trained VSGs are highly biased towards certain types of relationships (e.g., has, on); the relationship distribution is significantly long-tailed, and even the top-performing VSG generators fail to learn meaningful relationships accurately [26]. This results in noisy VSGs, which in turn degrades the quality of captions [14].
- There is an assumption in the existing approaches that TSGs and VSGs are compatible. But, are VSGs and TSGs actually compatible? TSGs, when used as inputs, can gen-

1. Introduction

The mainstream image captioning models rely on convolutional image features and/or attention to salient regions and objects to generate captions via recurrent models [20, 1]. Recently, scene graph representations of images have been used to augment captioning models so as to leverage their structural semantics, such as object entities, relationships and attributes [32, 30, 6]. The literature however has mixed opinion about the usefulness of scene graphs in captioning. Few works have reported improvements in caption generation using scene graphs [21, 30], while several others have highlighted that scene graphs alone yield poor captioning results and can even harm captioning performance [10, 14]. In this paper, we identify the challenges in effective utilization of scene graphs for image captioning, and subsequently investigate how to best harness them for this task.

Scene graph representation consisting of nodes and edges can be derived from either (i) images where the nodes correspond to the objects present in the scene, termed as Visual Scene Graphs (VSG), or (ii) from a caption where nouns and verbs take on the roles of nodes and edges in a rule-based semantic parsing, termed as Textual Scene Graphs (TSG). The literature of scene graph generation and scene graphs for image captioning primarily refers to the VSG representation.

To be able to leverage scene graphs for captioning, we need paired VSG-caption annotations. This is currently unavailable. Hence, methods requiring explicit scene graphs end up training the VSG generator and the caption generator on disparate datasets [30, 6, 14]. The current practice is to train VSG generators on the Visual Genome (VG) dataset, train TSG to caption generation on COCO-captions dataset, and finally transform the VG-trained VSGs to captions utilizing the later. We note two issues with this approach:

- The VG-trained VSGs are highly biased towards certain types of relationships (e.g., has, on); the relationship distribution is significantly long-tailed, and even the top-performing VSG generators fail to learn meaningful relationships accurately [26]. This results in noisy VSGs, which in turn degrades the quality of captions [14].
- There is an assumption in the existing approaches that TSGs and VSGs are compatible. But, are VSGs and TSGs actually compatible? TSGs, when used as inputs, can gen-

*Authors have equal contributions
Figure 1. **SG2Caps** first creates Visual Scene Graphs (VSG) by combining (1) pseudolabel - output of a black-box VSG generator, and (2) HOI graph from an HOI inference model. Each object node of the VSG has a bounding box label. Object nodes, relations, attributes are color-coded in red, blue, green respectively. The output of VSG encoding is the input for the LSTM-based decoder for the caption generation.

Figure 2. Characterization of TSG and VSG. While TSG only contains salient contents such as *man*, *motorcycle*, *flag* for natural language description, VSG includes unnecessary details such as *wheel*, *tire*, *window*, *sign*, *pole*. Objects, attributes, edges are shown in *pink*, *green*, *blue* respectively. (Best viewed in color)

We argue that this is the main reason why previous efforts to exploit VSGs for captioning did not achieve desired results.

To mitigate the above issues, we explore several novel ways to enhance VSGs in the context of captioning:

1. **Human-Object Interaction (HOI) information:** Humans tend to describe visual scenes involving humans by focusing on the human-object interactions at the exclusion of other details. If HOI information is extracted from an image, it can provide an effective way to highlight the ‘salient’ parts in its VSG, thereby bringing it closer to its corresponding TSG. Hence, we propose to harness pre-trained HOI inferences as partial VSGs, where all detected objects (not limited to humans) in a scene form the graph nodes and the HOI information augment a few relevant nodes with appropriate relationship and attributes.

2. **VSG grounding:** A unique aspect of VSG is that each node in an VSG is grounded, i.e., has a one-to-one association with the object bounding boxes in the image. This spatial information can be used to capture the relationship between objects. It is well known from the scene graph generation literature that the inter-object relationship classification performance greatly benefits from ground-truth bounding box locations [18, 17]. Despite this evidence, no VSG-based captioning model has yet used the spatial information of the nodes. We show that such information can significantly improve captioning performance.

This paper investigates how to best leverage VSGs for caption generation, if at all. To this end, we develop a new image captioning model, termed **SG2Caps**, that utilizes the VSGs alone for caption generation (see Fig. 1 for the main idea). In contrast to the existing work, we do not use any
image or object-level visual features; yet, we achieve competitive caption generation performance by exploiting the HOI information and VSG grounding. Directly utilizing the scene graph labels avoids expensive graph convolutions over high-dimensional CNN features, we show that it is still effective for caption generation via capturing visual relationships. This also results in 49% reduction in the number of trainable parameters comparing with the methods that require processing of both visual features and scene graphs. Researchers have shown that image captioning algorithms produce less accurate results due to inherent dataset biases and the unavailability of high quality human annotations. Our **SG2Caps** model leverages additional annotations from other sources beyond image caption datasets thereby reducing NLP bias.

Our contributions are summarized below.

- We show that competitive captioning performance can be achieved on the COCO-captions dataset using VSG alone, without any visual features.
- We experimentally show that VSGs and TSGs are not compatible with each other in the context of caption generation, and we propose to improve the learnable transformation directly from VSG to caption.
- We propose a new captioning model, **SG2Caps**, that utilizes spatial locations of VSG nodes and HOI information to generate better caption from VSGs. While node locations help to identify the meaningful relationships among objects (results in 5 point gain in CIDEr score), HOI captures the essence of natural language communication (results in 7 point gain in CIDEr score). Thus they help to close semantic gap between TSGs and VSGs for the purpose of image captioning.
- We also devise an extremely light-weight visual feature fusion strategy for the SG2Caps framework. The low-dimensional global visual features added as a summary node in the VSG, boosts image captioning performance atop blackbox scene-graph only model. Our **SG2Caps** model performs competitively against existing captioning models that operate on high-dimensional region level visual features.

2. Related work

**Image captioning:** Mainstream image captioning models [20, 4] directly feed convolutional image features to a recurrent network to generate natural language. The top-down approaches in such image captioning models rely on attention-based deep models [13, 15, 31, 24] where a partially-completed caption is utilized as the ‘context’. An attention, based on the context, is then applied to the output of one or more layers of a CNN. These approaches can predict attention weights without any prior knowledge of objects or salient image regions. Later, bottom-up approaches [1] enabled attention to be computed at the object-level using object detectors. Such object-level attention mechanism is the state-of-the-art in many vision-language tasks including image captioning and visual question answering.

**VSG in image captioning:** A number of works [27, 7, 18, 23, 28, 11, 33] devised approaches that strive to perform on VSG generation tasks on the benchmark VG dataset [5]. A few recent works [32, 30, 6, 21, 25, 10] have introduced the use of VSG (in addition to the visual features) with the hope that encoding of objects attributes and relationships would improve image captioning. Some of the works used implicit scene graph representation [32, 5], while others explored an explicit representation of relations and attributes [30, 21, 14, 9]. The explicit scene graph approaches integrate VSG features with CNN features from image or objects. Such explicit scene graph approaches use a scene graph generator as a blackbox. For example, Wang et al. [21] utilized FactorizableNet [11], some [30, 6, 35] used MotifNet [33] and others [10, 14] utilized Iterative Message Passing [23] as their blackbox scene graph generator.

Researchers found that VSG alone yields poor captioning results. The literature so far has mixed opinions about the usefulness of scene graphs. While Wang et al. [21] observed improvement in caption generation using VSG, Li and Jiang [10] did not find VSGs useful. Recently, an in-depth study concluded that it is the noise in VSG (often the relations) that harm the image captioning performance [14].

In contrast to the existing VSG-augmented captioning models, our model **SG2Caps** does not use object-level CNN features as inputs but utilizes the VSG labels primarily. We are aware of one work that used VSG labels as the only input in captioning and observed that VSG labels alone lead to unsatisfactory results [21]. Another recent method [35] was designed to utilize both the scene graphs and the region-level visual features. Our model significantly differs from them in several ways: (1) Our model incorporates novel techniques to make VSGs compatible for captioning; (2) we leverage HOI information to further improve captioning, and (3) our model can handle variable number of nodes in the VSGs. Our model achieves competitive image captioning performances with VSG labels alone. We also outperform the scene graph-only mode adapted from the work of Zhong et al. [35] that leverages a richer object class vocabulary (1600 vs 150) and external word embeddings.

3. Proposed model

Our model, **SG2Caps**, consists of a VSG generator, VSG encoder and a language decoder (Fig. 1). Given an image $I$, the VSG generator constructs a VSG $G$ that is particularly suited for caption generation. The VSG consists of objects, their attributes, spatial locations and inter-object relationships. The VSG encoder then generates context-aware em-
beddings, which are input to the language decoder. The language decoder consists of a long-short-term-memory network (LSTM) attention followed by an LSTM-based language model that generates the captions.

3.1. VSG generator

Our VSG generator has the following two components. **VG150 Pseudolabel**: Off-the-shelf VSG generators provide object classes as the node labels and pairwise relationships. We learn our own attribute classifier and train a VSG generator on VG150 using MotifNet [33]. This pretrained VSG generator is applied on the COCO images to create scene graph pseudolabels so as to create visual scene graphs with nodes, their attributes, their locations and pairwise relationships. The objects, attributes and relationships correspond to the vocabulary of VG150 consisting of 150 object classes, 50 relations and 203 attributes. Please note that COCO does not provide scene graph annotations. Also note that pseudolabel path is not bound to the specific one used in this paper, it can be any black-box visual scene graph generator.

**Partial COCO scene graph**: In parallel, we use an object detector pretrained on COCO images to create a list of COCO objects that serve as nodes of another graph, termed as HOI graph. Then we use an HOI interferer to generate a partial scene graph since it has a limited relationship information with mostly disconnected nodes but all in COCO vocabulary. Images where person category objects are not detected, the partial VSGs do not contain any HOI augmentation; it consists of only the list of nodes created from detected object instances of other COCO categories.

3.2. VSG encoder

We use the union of pseudolabels and HOI graphs as the VSG of an image. A VSG is a tuple \( G = (N, E) \), where \( N \) and \( E \) are the sets of nodes and edges. In our formulation, there are four kinds of nodes in \( N \): object nodes \( o \), attribute nodes \( a \), bounding box nodes \( b \) and relationship nodes \( r \). We denote \( o_i \) as the \( i \)-th object, \( r_{ij} \) as the relationship between \( o_i \) and \( o_j \), \( b_i \) as the bounding box coordinates of \( o_i \) and \( a_{i,l} \) as the \( l \)-th attribute of object \( o_i \). Each node in \( N \) is represented by a \( d \)-dimensional vector pertaining to the node type i.e., \( e_o, e_a, e_b \) and \( e_r \). The edges in \( E \) are defined as follows:

- If an object \( o_i \) has an attribute \( a_{i,l} \) there is a directed edge from \( o_i \) to \( a_{i,l} \).
- There is a directed edge from \( o_i \) to its bounding box \( b_i \).
- If there exists a relationship triplet \(< o_i , r_{ij} , o_j >\), two directed edges from \( o_i \) to \( r_{ij} \) and from \( r_{ij} \) to \( o_j \) are constructed.

Next, we transform the original node embeddings \( e_o, e_a, e_b, e_r \) to a new set of context-aware embeddings \( X = \{ x_{r_{ij}}, x_{o_i}, x_{a_{i,l}}, x_{b_i} \} \), where \( x_{r_{ij}} \) is the relationship embedding for the relationship node \( r_{ij} \), \( x_{o_i} \) and \( x_{a_{i,l}} \) are the object embedding, attribute embedding and bounding box embedding for the object node \( o_i \). We use five spatial graph convolution functions \( g_r, g_o, g_b, g_a, \) and \( g_e \) to generate the above embeddings. All of them have same structure with independent parameters: a fully-connected layer, followed by a ReLU.

**Relationship embedding** \( x_{r_{ij}} \): Given a relationship triplet \(< o_i , r_{ij} , o_j >\) in \( G \), \( x_{r_{ij}} \) is defined in the context of the subject \( (o_i) \), object \( (o_j) \) and predicate \( (r_{ij}) \) together as follows:

\[
x_{r_{ij}} = g_r(e_{o_i}, e_{r_{ij}}, e_{o_j})
\]

**Attribute embedding** \( x_{a_i} \): For an object node \( o_i \) with its attributes \( a_{i,1:N_{a_i}} \) in \( G \), the embedding \( x_{a_i} \) is given by

\[
x_{a_i} = \frac{1}{N_{a_i}} \sum_{l=1}^{N_{a_i}} g_a(e_{o_i}, e_{a_{i,l}})
\]

where \( N_{a_i} \) is the number of attributes for \( o_i \). Here the context of an object with all its attributes are incorporated.

**Bounding box embedding** \( x_{b_i} \): Given \( o_i \) with its bounding box \( b_i \), \( x_{b_i} \) is defined as:

\[
x_{b_i} = g_b(e_{o_i}, e_{b_i})
\]

**Object embedding** \( x_{o_i} \): An object node \( o_i \) plays different roles based on the edge directions, i.e., whether \( o_i \) acts as a subject or the object in a triplet. Following past work [30], our object embedding takes the entire triplet into consideration. We define \( x_{o_i} \) as follows:

\[
x_{o_i} = \frac{1}{N_{r_{i}}} \left[ \sum_{o_j \in sbj(o_i)} g_s(e_{o_i}, e_{o_j}, e_{r_{ij}}) + \sum_{o_j \in obj(o_i)} g_o(e_{o_j}, e_{o_i}, e_{r_{ij}}) \right]
\]

where \( N_{r_{i}} = |sbj(i)| + |obj(i)| \) is the number of relationship triplets involving \( o_i \). Each node \( o_j \in sbj(o_i) \) acts as an object while \( o_i \) acts as a subject.

Note that our model differs from SGAE in the ways we learn parameters for \( x_{b_i} \) and fuse different output embeddings. We combine \( x_{o_i}, x_{b_i}, x_{a_i} \) with a sum operation at node-level to form \( X \) before feeding them to the attention LSTM, whereas SGAE simply concatenated them.

**Fusing visual features** We employ a simple strategy to augment the scene graph features optionally with visual features. We add a summary projection node in the output graph embedding. The summary node is generated using global pooling of the image-level \( P_b \) features, followed
by a projection layer that takes the 256-dimensional summary node feature to 128-dimensional vector, followed by a ReLU non-linearity. The projected summary node dimension is compatible with the GCN embedding size. This projected summary node feature is concatenated with the \( \mathcal{X} \) vector generated as mentioned above. The above feature fusion strategy based on summary node also differs from others [30, 35] which fuse features at individual node level. This feature fusion has a negligible parameter overhead (256×128 additional parameters) on top of SG2Caps.

### 3.3. Language Model

Given the VSG \( \mathcal{G} \) of an image \( I \), we want to generate a natural language sentence \( w_{1:T} = w_1, w_2, ..., w_T \) that describes the image. In SG2Caps, we follow a 2-layer encoder-decoder LSTM architecture for this part. The encoder, an attention LSTM, takes the VSG encoding \( \mathcal{X} \) as input. This is in contrast to the popular captioning models which CNN features are used. The decoder is an LSTM-based language decoder. This architecture is the same as the work of Anderson et al. [1].

The LSTM operations are denoted as: \( \mathbf{h}_t = LSTM(\mathbf{x}_t, \mathbf{h}_{t-1}) \), where, \( \mathbf{x}_t \) is the LSTM input vector and \( \mathbf{h}_t \) is the LSTM output vector. Let the LSTM states for the attention layer and the decoder layer at time step \( t \) be \( \mathbf{h}_t^1 \) and \( \mathbf{h}_t^2 \). At each time step \( t \), the attention LSTM captures contextual information \( \mathbf{x}_t^1 \) by concatenating the previous hidden state of the decoder LSTM, the mean-pooled VSG features (\( \bar{\mathbf{f}} = \frac{1}{k} \sum_{i=1}^{k} \mathbf{f}_i \)) and the previously generated word representation as following:

\[
\mathbf{x}_t^1 = \text{concat}(\mathbf{h}_{t-1}^2, \bar{\mathbf{f}}, \mathbf{W}_e \mathbf{u}_t)
\]

where \( \mathbf{W}_e \) is the word embedding matrix for vocabulary \( \Sigma \) and \( \mathbf{u}_t \) is its one-hot encoding at \( t \).

We generate normalized attention weights \( \alpha_{t} \) for the VSG features at time step \( t \) as follows:

\[
\begin{align*}
\mathbf{a}_{t,t} &= \mathbf{w}_a^T \text{tanh}(\mathbf{W}_{fa} \mathbf{f}_t + \mathbf{W}_{ha} \mathbf{h}_t^1) \\
\alpha_{t,t} &= \text{softmax}(\mathbf{a}_{t,t})
\end{align*}
\]

where \( \mathbf{w}_a^T \in \mathbb{R}^H, \mathbf{W}_{fa} \in \mathbb{R}^{H \times D_f}, \mathbf{W}_{ha} \in \mathbb{R}^{H \times H} \) are learnable weights. The attended VSG features, \( \mathbf{f}_t \), input to the decoder LSTM is thus a convex combination of the input features \( \mathbf{f}_t \).

\[
\hat{\mathbf{f}}_t = \sum_{i=1}^{N_f} \alpha_{t,i} \mathbf{f}_i
\]

The \( \mathcal{X} \) formulation allows for learning attention coefficients for each object-level node of the scene graph.

The input to the decoder LSTM consists of the previous hidden state from the attention LSTM layer, and the attention weighted VSG features: \( \mathbf{x}_t^2 = [\mathbf{h}_t^2, \hat{\mathbf{f}}_t] \). For a sequence of words \( w_1, w_2, ..., w_T \), denoted as \( w_{1:T} \), the conditional distribution over possible output words at \( t \) is given by:

\[
p(w_t | w_{1:t-1}) = \text{softmax}(\mathbf{W}_p \mathbf{h}_t^2 + \mathbf{b}_p),
\]

where, \( \mathbf{W}_p \in \mathbb{R}^{2|\mathcal{X}| \times H} \) and \( \mathbf{b}_p \in \mathbb{R}^{2|\mathcal{X}|} \) are learned weights and biases. The distribution over complete output sequences is calculated as the product of conditional distributions.

Given a target ground truth captions \( w_{1:T}^* \), and a captioning model with parameters \( \theta \), we train our encoder-decoder model using one of the two loss functions:

(i) Minimize a cross-entropy loss:

\[
L_{XE}(\theta) = - \sum_{t=1}^{T} \log(p_y(w_t^* | w_{1:t-1}^*))
\]

(ii) Maximize a reinforcement learning (RL)-based reward [15]

\[
R_{RL}(\theta) = \mathbb{E}_{w_{1:T} \sim p}[rw(w_{1:T})]
\]

where, \( rw(\cdot) \) is the score function (e.g. the CIDEr metric).

### 4. Experiments

#### 4.1. Datasets

**The COCO-Captions [3]:** We conducted the experiments and evaluation our proposed SG2Caps model on the Karpathy split of COCO-Caption dataset for the offline test. This split has 113, 287/5, 000/5, 000 train/val/test images, each of which has 5 captions.

**Visual Genome (VG) [8]:** We used the VG150 dataset to pretrain a scene graph generator and attribute classifier. VG comes with scene graph annotations, such as object categories, object attributes, and pairwise relationships. These are utilized to train an object proposal detector, attribute classifier, and relationship classifier as our VSG parser. We first pre-trained a faster R-CNN based object detector [16] using object annotations from Visual Genome with an additional 2-layer multi-layer-perceptron layer for 203 one-vs-all attribute classifiers. Then the Neural Motif [33] model serves as the ROI head to predict the pair-wise relationships.

**Verbs in COCO:** The V-COCO dataset contains a subset of COCO images, and was created for evaluating the HOI task. It has annotations for 16K people-instances in 10K images with their actions and associate objects labeled. We utilize VSGNet [19], a pre-trained HOI model, to generate the inference on COCO images.

#### 4.2. Experimental settings

**Processing pseudolabels:** The VSGs generated by an inference on COCO-caption images utilize the pre-trained MotifNet [33] as a black box (refer to Section 4.1). The predicted scene graphs are noisy, containing many object proposals many of which are duplicates. They also contain predicted attributes for each proposal and pairwise relationship for all objects. We adapt these VSGs to make them better suited for caption generation. A blackbox VSG generators were trained to achieve high retrieval performance measured by recall. This metric only cares about higher
fraction of correct matches to be returned, but does not penalize duplicate objects. However, a VSG to be used for caption generation needs to be free from duplicate objects.

We thus discard the less accurate (confidence score below 0.25) object predictions, and apply non-maximum-suppression (NMS) on object proposals with intersection-over-union (IOU) threshold of 30%. We also discard weak relationships, if the confidence score is below 30% and keep only the best attribute per node provided the confidence is above 90%.

**Processing HOI graph:** HOI graphs are constructed by extracting relationships and attributes from the HOI inference network [19], which utilizes the instance detection results from detectron2. Images, where human objects are detected with a score of 0.5 or more, are selected as inputs to the HOI network. Outputs of the HOI network are \(< agent(human) - instrument - object > triplets associated with a HOI relation, hr. For example, in the case of ‘a person hitting a ball with a bat’, the triplet takes the form of \(< person(agent) - bat - ball > for the action ‘hitting’. To transform such a triplet into a graph, all agents, objects and instruments are considered as the nodes of the graph, and the associated HOI-relation (hr) is added as a relationship between the subject and the object. The list of such HOI relations consists of 10 different semantic verb actions. On the other hand, for inferences without objects e.g. stand, we add them as an attribute e.g. standing to the subject, since it is not possible to form any directed edge between a subject and an object. The list of such object-less actions that are utilized as attributes consists of 16 different semantic verbs. HOI graphs generated in the above way have such relations or attributes for 47726 training images and 1912 test images. Other images transform to graphs with only the detected objects as nodes with only one relation \(< object1, AND, object2 > per image. Since the goal of utilizing scene graphs for captioning is to enrich the model with objects relationships, we argue that spatial locations of nodes should be leveraged. Our VSG thus contains a list of nodes each consisting of object class label, bounding box, attributes and a set of edges. A few object categories (15 such categories) such as **potted plant** from the generated HOI graph don’t appear in the captioning vocabulary. In such cases, the HOI node category is mapped to the closest word plant from the captioning vocabulary. For our experiments, we use the union of pseudolabels and HOI graphs as the VSG of an image. For the inference in language generation, we use greedy search.

**Visual Features** When using additional visual features, we utilize low-dimensional global image features in contrast to the recent captioning methods which use high-dimensional region-level image features (256-D vs 2048-D). We append a projected summary node to the GCN output for each image, where the projected summary node is generated via a learnable projection head applied to the 256-D global pooled F6 features per image.

### 4.3. Results

Our implementation is built upon the source code [29] of SGAE [30]. Our models are trained on a single NVIDIA 1080Ti GPU running pytorch 0.4.0 in python 2.7.15. We evaluate our caption generation model on standard metrics such as BLEU@1(B@1), BLEU@4 (B@4), ROUGE(R), METEOR(M), CIDEr(C) and SPICE(S).

**TSG-VSG incompatibility:** First we show the incompatibility between TSG and VSG in Table 1. All the entries in this table are from our reproduced models. TSG, when used as an input, generates excellent captions using the GCN and LSTM language model, as can be seen from the first row. The above TSG-caption model was trained with cross-entropy loss. However, if we simply use that model and perform inference using the pseudolabel as inputs, it performs significantly worse. Although the HOI augmentation improves the CIDEr score by 7 points, the overall caption generation performance still remains poor. The takeaway message is that although VSG and TSG are similar type of representations, but they are not directly compatible to each other for caption generation.

**Caption generation performance comparison:** In our work, we focus on scene graphs for caption generation, and thus limit our experiments and comparisons to the LSTM language model. Table 2 shows the main result of our proposed **SG2Caps** method. The bottom half of the table denotes methods that use only scene graph labels as input to the captioning model. Our graph construction differs from [21] in a few ways. Wang et al. used FactorizableNet [11] as the relation detection model on top of RPN from the Faster R-CNN. Wang et al. didn’t use bounding box locations which we find important for caption generation. We use a simpler relation model [33] for scene graph detection. **SG2Caps** also outperforms the most recent captioning model, Sub-GC [35] adapted for G-only mode. Since the paper does not report the G-only model performance, we have adapted their code [34] so as to use only the scene

<table>
<thead>
<tr>
<th>Model</th>
<th>B@1</th>
<th>B@4</th>
<th>M</th>
<th>R</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSG †</td>
<td>96.2</td>
<td>56.7</td>
<td>35.5</td>
<td>68.7</td>
<td>158.6</td>
<td>31.2</td>
</tr>
<tr>
<td>PL</td>
<td>46.7</td>
<td>7.7</td>
<td>14.6</td>
<td>35.7</td>
<td>31.5</td>
<td>9.0</td>
</tr>
<tr>
<td>PL + HOI</td>
<td>50.7</td>
<td>9.1</td>
<td>15.8</td>
<td>37.4</td>
<td>38.6</td>
<td>9.4</td>
</tr>
</tbody>
</table>

Table 1. **Incompatibility between TSG and VSG.** Performance of the caption model trained on textual scene graph (TSG) while evaluating on different scene graphs as input. TSG row denotes sentence scene graph as inputs. † denotes our reproduced results with cross-entropy loss using [29]. PL denotes pseudolabel and (PL + HOI) denotes the union of pseudolabel and HOI graph as the input respectively. Inference on TSG-trained model fail to generate decent captions with VSG input.
Table 2. Comparison with State-of-the-art captioning models.

We compare with both image captioning SoTA methods (that use both region-level visual features and scene-graphs labels) and methods that use scene graphs as only input. Here, all methods use LSTM language model. RSG-G1 and RSG-G2 refer to without and with gate-wise gating in the G-only setup from RSG. All results are from the corresponding papers. † denotes our implementation based on [34]. ⊕ denotes captioning models that use global visual features. SG2Caps - RL denotes CIDEr-based optimization, all others use cross-entropy loss.

<table>
<thead>
<tr>
<th>Model</th>
<th>B@4</th>
<th>M</th>
<th>R</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>visual scene graph + visual features</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGAE [30]</td>
<td>36.9</td>
<td>27.7</td>
<td>57.2</td>
<td>116.7</td>
<td>20.9</td>
</tr>
<tr>
<td>R-SCAN [9]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>114.9</td>
<td>20.9</td>
</tr>
<tr>
<td>SubGC [35]</td>
<td>36.9</td>
<td>27.9</td>
<td>56.8</td>
<td>114.8</td>
<td>20.8</td>
</tr>
<tr>
<td>KMSL [10]</td>
<td>33.8</td>
<td>-</td>
<td>54.9</td>
<td>110.3</td>
<td>19.8</td>
</tr>
<tr>
<td>Attribute [22]</td>
<td>31.0</td>
<td>26.0</td>
<td>-</td>
<td>94.0</td>
<td>-</td>
</tr>
<tr>
<td>RSG [21]</td>
<td>34.5</td>
<td>26.8</td>
<td>55.9</td>
<td>108.6</td>
<td>20.3</td>
</tr>
<tr>
<td>SCST:Att2in [15]⊕</td>
<td>31.1</td>
<td>26.0</td>
<td>54.3</td>
<td>101.3</td>
<td>-</td>
</tr>
<tr>
<td><strong>SG2Caps (ours)⊕</strong></td>
<td><strong>32.6</strong></td>
<td><strong>26.4</strong></td>
<td><strong>55.0</strong></td>
<td><strong>106.6</strong></td>
<td><strong>19.8</strong></td>
</tr>
<tr>
<td><strong>visual scene graph only</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSG-G1 [21]</td>
<td>22.8</td>
<td>20.6</td>
<td>46.7</td>
<td>66.3</td>
<td>13.5</td>
</tr>
<tr>
<td>RSG-G2 [21]</td>
<td>22.9</td>
<td>21.1</td>
<td>47.5</td>
<td>70.7</td>
<td>14.0</td>
</tr>
<tr>
<td>SubGC [35]†</td>
<td>31.5</td>
<td>25.5</td>
<td>53.4</td>
<td>98.9</td>
<td>18.4</td>
</tr>
<tr>
<td><strong>SG2Caps (ours)†</strong></td>
<td><strong>32.0</strong></td>
<td><strong>26.2</strong></td>
<td><strong>54.9</strong></td>
<td><strong>104.4</strong></td>
<td><strong>19.5</strong></td>
</tr>
<tr>
<td><strong>SG2Caps - RL (ours)†</strong></td>
<td><strong>33.0</strong></td>
<td><strong>26.2</strong></td>
<td><strong>55.6</strong></td>
<td><strong>112.3</strong></td>
<td><strong>19.4</strong></td>
</tr>
</tbody>
</table>

graphs (excluding the visual features) for the captioning. Please note, we used their scene graphs [34] while reproducing their G-only captioning model which has 1600 object classes while our object class vocabulary contains only 230 words. The graph construction and the GCN architecture in [35] both differ from SG2Caps. SG2Caps is able to process variable number of nodes per VSG, while [35] can operate on a fixed number of nodes per VSG per the architecture design. Our SG2Caps significantly outperforms the existing G-only models by large margin. With HOI augmentation and bounding box feature, SG2Caps produces competitive results close to the SoTA models that rely on object detection CNN features.

We also compare our model with the state-of-the-art captioning models that use both the visual features and explicit scene graphs. All such methods utilized 2048-D object detection CNN features for each salient region. Wu et al. [22] utilized classified object attributes on top of the visual features. The “Know more, say less” model (KMSL) extracts features for objects and relations based on the scene graph, which are passed through two attention heads and finally combined using a flat attention head. R-SCAN [9] utilized soft alignment using attention and visual features to associate the region features and the relation features to the word features. Their scene graph generation was trained on VGG dataset [12] instead of Visual Genome dataset widely used in other papers. RSG [21] explored on how to integrate relation-aware scene graph features encoded by Graph Convolution with region-level image features to boost image captioning performance. SGC [14] employed conditional graph-attention network to fuse scene graph features and region-level visual features and performed in-depth study on when scene graphs are useful for image captioning and when they are not. Sub-GC [35], although focused on sub-graph based comprehensive and diverse caption generation, they also report competitive image captioning results based on the full-graph. We include it as one of the most recent captioning model for comparison. SGAE [30] reported the best caption generation performance on our test dataset. SGAE utilized a shared memory for leveraging language inductive bias from the textual scene graphs and employed spatial graph convolution on scene graph representation and region-level visual features. These methods apply graph convolutions on high dimensional visual features, increasing the memory footprint of the captioning model. Unlike high dimensional (2048-D) region-level visual features, we utilize low-dimensional (256-D) image-level visual features in our visual-feature augmented SG2Caps framework. For example, if an image has 20 objects on an average, SG2Caps needs to operate on 160× fewer dimensional inputs per image, resulting into significantly lower memory footprint while operating on a batch of 100 images. Augmenting the visual features in the SG2Caps framework results in additional 2-point gain of CIDEr score and overall performs competitively with best captioning models requiring graph convolutions on high-dimensional visual features. Please note that our aim is to avoid the GCN operations on the high-dimensional feature space and keep the memory footprint low with a negligible parameter overhead, and thus we avoid the use of object detection CNN features. We compared with the best captioning model that uses global feature [15]. However, SG2Caps can also be extended to use the object detection CNN features per VSG node similar to existing state-of-the-art methods. Results in Table 2 show that our SG2Caps substantially minimizes the gap between the performances of state-of-the-art caption generation models and a scene-graph-only model. We leverage visual features in a memory-efficient way. The visual-feature augmented SG2Caps is also able to retain the low-memory footprint. We note that the number of trainable parameters for captioning models that use both the high dimensional CNN features and scene graph labels such as SGAE are significantly higher than our scene graph only model. For example, SG2Caps has 49% fewer parameters comparing with SGAE [30] full-model (21M vs 41M) for the same language decoder model. The optional visual feature fusion model has an overhead of 0.03M parameters.

Ablation experiments: Table 3 shows the results of our ablation experiments. We observe that when the post-processed pseudolabels are directly used for training a cap-
Figure 3. Qualitative examples: generated captions from different baseline models from Karpathsy test split for COCO image ids #177861, #45710, #553879, #873899 respectively.

Table 3. Ablation results: Performance of our SG2Caps model, when the object groundings and HOI are leveraged. All models were trained with Cross-entropy loss.

<table>
<thead>
<tr>
<th>Model</th>
<th>bbox</th>
<th>HOI</th>
<th>g_feat</th>
<th>B@1</th>
<th>B@4</th>
<th>M</th>
<th>R</th>
<th>C</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>72.0</td>
<td>29.3</td>
<td>24.5</td>
<td>52.6</td>
<td>92.6</td>
<td>17.8</td>
</tr>
<tr>
<td>BBox</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>73.1</td>
<td>30.3</td>
<td>25.1</td>
<td>53.4</td>
<td>97.4</td>
<td>18.4</td>
</tr>
<tr>
<td>Global</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>73.3</td>
<td>30.7</td>
<td>25.5</td>
<td>53.9</td>
<td>98.1</td>
<td>18.5</td>
</tr>
<tr>
<td>SG2Caps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>75.0</td>
<td>32.6</td>
<td>26.4</td>
<td>55.0</td>
<td>106.6</td>
<td>19.8</td>
</tr>
<tr>
<td>SG2Caps-G</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>75.0</td>
<td>32.6</td>
<td>26.4</td>
<td>55.0</td>
<td>106.6</td>
<td>19.8</td>
</tr>
</tbody>
</table>

Qualitative results: Our baseline generates convincing caption sentences. Fig. 3 shows that the caption quality improves over the baseline to BBox and SG2Caps models. We also show captions from SGAE [30], one of the best captioning models that uses both visual features and scene graphs. SG2Caps is able to generate high-quality and competitive captions without using visual features (see Fig. 3). Fig. 4 shows noisy pseudolabels and HOI graphs and their corresponding SG2Caps output for example images (image ids 548361 and 258628) from COCO Karpathsy test split. The HOI graph detects semantic verb relations which are complementary to the pseudolabel that detects many other objects, relations and their attributes. As seen in the first example, the words in the output caption come from both the pseudolabels (such as baseball player, ball, field) and the HOI-graph (throw, ball).

5. Conclusion

Explicit encoding of objects, attributes and relations are useful information for image captioning. However, blindly using visual scene graphs for captioning fails to produce reasonable caption sentences. The proposed SG2Caps pipeline enables networks pre-trained for (1) SGDet on other scene graph datasets, and (2) semantic roles on HOI datasets to greatly reduce the gap in accuracy on COCO caption datasets – indicating strong captioning models can be achieved with low dimensional objects and relations label space only. These results further strengthen our defense of scene graph for image captioning. We hope our observations can open up new opportunities for vision and language research in general.
Supplementary Material

SG2Caps with global visual features:

For the global features, we first extract P6 features from the ResNeXt101-FPN backbone using global max pooling, resulting in 256-D global visual features per image. This is treated as the global visual features - one set of features for each image. Next, we employ a global feature fusion strategy as follows. We add a summary projection node in the output graph embedding. The summary node is generated using global pooling of the image-level F6 features, followed by a projection layer that takes the 256-dimensional summary node feature to 128-dimensional vector, and then a ReLU non-linearity. The projected summary node dimension is compatible with the GCN embedding size. This projected summary node feature is concatenated with the scene graph embedding vector generated in the SG2Caps. This feature fusion has a negligible parameter overhead (256×128 additional parameters) on top of SG2Caps. Using scene graph embeddings results in 7 point gain in CIDEr score (106.6 vs 99.5). It also means an extremely lightweight global feature fusion module boosts the SG2Caps model by further 2 points in CIDEr score (106.6 vs 104.4) when global visual features are being used.

Training Details

The main model, SG2Caps was trained on a single Tesla M40 24GB. The batch size was set at 300, with an accumulate number of 1. The learning rate was initially set at 5e−4, with a learning rate decay of .8 that applied every 3 epochs. This model was trained for 8.5k iterations using the cross entropy loss before moving to the reinforcement learning, which was trained for an additional 3k iterations. The SG2Caps-G model was trained with two Tesla M40 24GBs, with a batch size of 200 with an accumulate number of 2. The rest of the training details are the same. This model trained on the cross entropy loss for 8.5k iterations.

Additional Examples

We have anecdotally compared the SG2Caps results with the SGAE for several images. Results in Figure 6 show three such images where our model performed closely to the SGAE model. This is good because it shows that our SG2Caps model can produce similar captions in comparison to SGAE, even though our model has almost half of the trainable parameters of SGAE. Results in Figure 7 show three such images where SG2Caps generated better captions than the SGAE model.

Results in Figure 5 show different images where our SG2Caps model performed worse, in comparison to SGAE. As is seen in Figure 5 in image #195750, where the model sees multiple humans, SG2Caps returns a caption stating that a group of people doing an action, rather than one or two people doing the action with the rest of the humans just being part of the crowd.

References

[12] Yuanzhi Liang, Yalong Bai, Wei Zhang, Xueming Qian, Li Zhu, and Tao Mei. Rethinking visual relationships for high-level image understanding, 02 2019. 7
Figure 5. Qualitative examples where SG2Caps performed worse than SGAE: generated captions from different baseline models from Karpathy test split for COCO image ids #235778, #195750, #214224 respectively.

Figure 6. Qualitative examples where SG2Caps is as good as SGAE: generated captions from different baseline models from Karpathy test split for COCO image ids #191069, #243213, #553447 respectively.

Figure 7. Qualitative examples where SG2Caps is arguably better than SGAE: generated captions from different baseline models from Karpathy test split for COCO image ids #307523, #459665, #209868 respectively.


