

# Segmentation-grounded Scene Graph Generation

Siddhesh Khandelwal<sup>\*,1,2</sup> Mohammed Suhail<sup>\*,1,2</sup> Leonid Sigal<sup>1,2,3</sup>

<sup>1</sup>Department of Computer Science, University of British Columbia

<sup>2</sup>Vector Institute for AI

<sup>3</sup>CIFAR AI Chair

skhandel@cs.ubc.ca

suhail33@cs.ubc.ca

lsigal@cs.ubc.ca

## Abstract

Scene graph generation has emerged as an important problem in computer vision. While scene graphs provide a grounded representation of objects, their locations and relations in an image, they do so only at the granularity of proposal bounding boxes. In this work, we propose the first, to our knowledge, framework for pixel-level segmentation-grounded scene graph generation. Our framework is agnostic to the underlying scene graph generation method and address the lack of segmentation annotations in target scene graph datasets (e.g., Visual Genome [24]) through transfer and multi-task learning from, and with, an auxiliary dataset (e.g., MS COCO [29]). Specifically, each target object being detected is endowed with a segmentation mask, which is expressed as a lingual-similarity weighted linear combination over categories that have annotations present in an auxiliary dataset. These inferred masks, along with a Gaussian masking mechanism which grounds the relations at a pixel-level within the image, allow for improved relation prediction. The entire framework is end-to-end trainable and is learned in a multi-task manner. Code is available at [github.com/ubc-vision/segmentation-sg](https://github.com/ubc-vision/segmentation-sg).

## 1. Introduction

Scene graph generation, has emerged as a dominant problem in computer vision literature over the last couple of years [36, 44, 50, 51, 53]. The task involves producing a graph-based *grounded* representation of an image, which characterizes objects and their relationships. A scene graph representation, first introduced in [50], encodes a scene as a graph where nodes correspond to objects (encoding object instances with corresponding class labels and spatial locations) and directed edges corresponding to relationships. The ultimate goal of scene graph generation is to produce such representation from a raw image [24] or video [18]. Scene graph representations have proved to be important

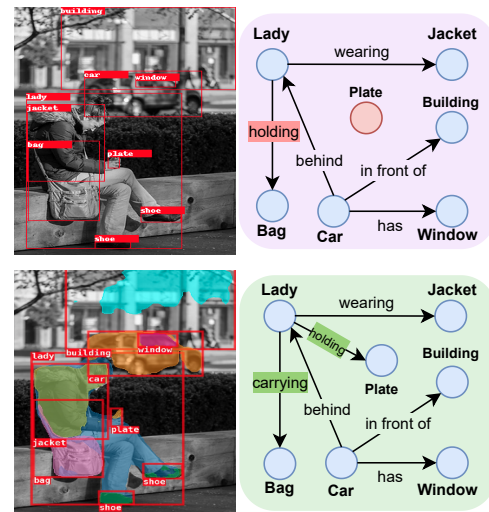


Figure 1. **Segmentation-grounded scene graph generation.** The image on top is the output of an existing scene graph generation method [45]. The bottom image is the output of augmenting our approach to [45]. The effective grounding of objects to pixel-level regions within the image leads to better relation predictions.

for a variety of higher level tasks (e.g., VQA[16, 45], image captioning [10, 52] and others). Most approaches to date have focused on appropriate modeling of context [51, 55], data imbalance in labels [30, 44] and, most recently, structural dependencies among the output variables [42].

One of the dominant limitations of all existing scene graph generation techniques, mentioned above, is the fact that both the nodes (objects) and edges (relations) are grounded to (rectangular) bounding boxes produced by the object proposal mechanism directly (e.g., pre-trained as part of R-CNN) or by taking a union of bounding boxes of objects involved in a relation. A more granular and accurate pixel-level grounding would naturally be more valuable. This has been shown to be the case in other visual and visual-lingual tasks (e.g., referring expression comprehension [15, 31], video segmentation with referring expression [22] and instance segmentation with Mask-RCNN [11]). In addition, grounding to segmentations could improve the overall performance of the scene graph generation by focus-

\*Denotes equal contribution

ing node and edge features on irregular regions corresponding to objects or *interface* between objects, constituting an interaction. The goal of our approach is to do just that.

However, pixel-level grounding comes with a number of unique challenges. The foremost of which is that traditional scene graph datasets, such as Visual Genome [24], do not come with instance-level segmentation annotations. This makes it impossible to employ a traditional fully supervised approach. Further, even if we were to collect segmentation annotations, doing so for a large set of object types typically involved in Scene Graph predictions would be prohibitively expensive<sup>1</sup>. To address this, we propose a transfer and multi-task learning formulation that uses an external dataset (e.g., MS COCO [29]) to provide segmentation annotations for some categories; while leveraging standard scene graph dataset (e.g., Visual Genome [24]) to provide graph and bounding box annotations for the target task.

On a technical level, for a target object that lacks segmentation annotations, its mask is expressed as a weighted linear combination over categories for which annotations are present in an external dataset. This transfer is realised by leveraging the linguistic similarities between the target object label and these supervised categories, thus enabling grounding objects to segmentation masks without introducing any annotation cost. For a pair of objects that share a relation, our approach additionally employs a Gaussian masking mechanism to assign this relation to a pixel-level region within the image. Through joint optimization over the tasks of scene graph and segmentation generation, our approach achieves simultaneous improvements over both tasks. Our proposed method is end-to-end trainable, and can be easily integrated into any existing scene graph generation method (e.g., [45, 55]).

**Contributions:** Our foremost contribution is that we propose the first, to our knowledge, framework for pixel-level segmentation-grounded scene graph generation/prediction, which can be integrated with any existing scene graph generation method. For objects, these groundings are realised via segmentation masks, which are computed through a lingual-similarity based zero-shot transfer mechanism over categories in an auxiliary dataset. To effectively ground relations at a pixel level, we additionally propose a novel Gaussian masking mechanism over segmentation masks. Finally, we demonstrate the flexibility and efficacy of our approach by augmenting it to existing scene graph architectures, and evaluating performance on the Visual Genome [24] benchmark dataset, where we consistently outperform baselines by up to 12% on relation prediction.

## 2. Approach

We propose a novel multi-task learning framework that leverages instance-level segmentation annotations, obtained

<sup>1</sup>As per [1], labelling one image in VOC [7] takes 239.7 seconds.

via a *zero-shot* transfer mechanism, to effectively generate pixel-level groundings for the objects within a scene graph. Our approach, highlighted in Figure 2, builds on existing scene graph generation methods, but is *agnostic* to the underlying architecture and can be easily integrated with existing state-of-the-art approaches.

### 2.1. Notation

Let  $\mathcal{D}^g = \{(\mathbf{x}_i^g, \mathbf{G}_i^g)\}$  denote the dataset containing graph-level annotations  $\mathbf{G}_i^g$  for each image  $\mathbf{x}_i^g$ . We represent the scene graph annotation  $\mathbf{G}_i^g$  as a tuple of object and relations,  $\mathbf{G}_i^g = (\mathbf{O}_i^g, \mathbf{R}_i^g)$ , where  $\mathbf{O}_i^g \in \mathbb{R}^{n_i \times d_g}$  represents object labels and  $\mathbf{R}_i^g \in \mathbb{R}^{n_i \times n_i \times d'_g}$  represents relationship labels;  $n_i$  is the number of objects in an image  $\mathbf{x}_i^g$ ;  $d_g$  and  $d'_g$  are the total number of possible object and relation labels, respectively, in the dataset.

In addition, we assume availability of the dataset  $\mathcal{D}^m = \{(\mathbf{x}_i^m, \mathbf{M}_i^m)\}$ , where each image  $\mathbf{x}_i^m$  has corresponding instance-level segmentation annotations  $\mathbf{M}_i^m$ . Finally,  $d_m$  are the total number of possible object labels in  $\mathcal{D}^m$ .

As is the case with existing scene graph datasets like Visual Genome [24],  $\mathcal{D}^g$  does not contain any instance-level segmentation masks. Also,  $\mathcal{D}^m$  can be any dataset (like MS COCO [29]). Note, that in general, the images in the two datasets,  $\mathcal{D}^g$  and  $\mathcal{D}^m$ , are disjoint and the object classes in the two datasets may have minimal overlap (e.g., MS COCO provides segmentations for 80, while Visual Genome provides object bounding boxes for 150 object categories<sup>2</sup>).

For brevity, we drop subscript  $i$  for the rest of the paper.

### 2.2. Scene Graph Generation

Given an image  $\mathbf{x}^g \in \mathcal{D}^g$ , a typical scene graph model defines the distribution over the scene graph  $\mathbf{G}^g$  as follows,

$$\Pr(\mathbf{G}^g | \mathbf{x}^g) = \Pr(\mathbf{B}^g | \mathbf{x}^g) \cdot \Pr(\mathbf{O}^g | \mathbf{B}^g, \mathbf{x}^g) \cdot \Pr(\mathbf{R}^g | \mathbf{O}^g, \mathbf{B}^g, \mathbf{x}^g) \quad (1)$$

The bounding box network  $\Pr(\mathbf{B}^g | \mathbf{x}^g)$  extracts a set of boxes  $\mathbf{B}^g = \{\mathbf{b}_1^g, \dots, \mathbf{b}_n^g\}$  corresponding to regions of interest. This can be achieved using standard object detectors such as Faster R-CNN [39] or Detectron [46]. Specifically, these detectors are pretrained on  $\mathcal{D}^b$  with the objective to generate accurate bounding boxes  $\mathbf{B}^b$  and object probabilities  $\mathbf{L}^g = \{\mathbf{l}_1^g, \dots, \mathbf{l}_n^g\}$  for an input image  $\mathbf{x}^g$ . Note that this only requires access to the object (node) annotations in  $\mathbf{G}^g$ .

The object network  $\Pr(\mathbf{O}^g | \mathbf{B}^g, \mathbf{x}^g)$ , for each bounding box  $\mathbf{b}_j^g \in \mathbf{B}^g$ , utilizes feature representation  $\mathbf{z}_j^g$ , where  $\mathbf{z}_j^g$  is computed as  $\text{ROIAlign}(\mathbf{x}^g, \mathbf{b}_j^g)$ , which extracts features from the area within the image corresponding to the bounding box  $\mathbf{b}_j^g$ . These features, alongside object label probabilities  $\mathbf{l}_j^g$ , are fed into a context aggregation layer such as

<sup>2</sup>Visual Genome has a  $\sim 47\%$  image overlap with MS-COCO. However, they have differing object categories and annotations. We make no use of this implicit image overlap in our formulation.

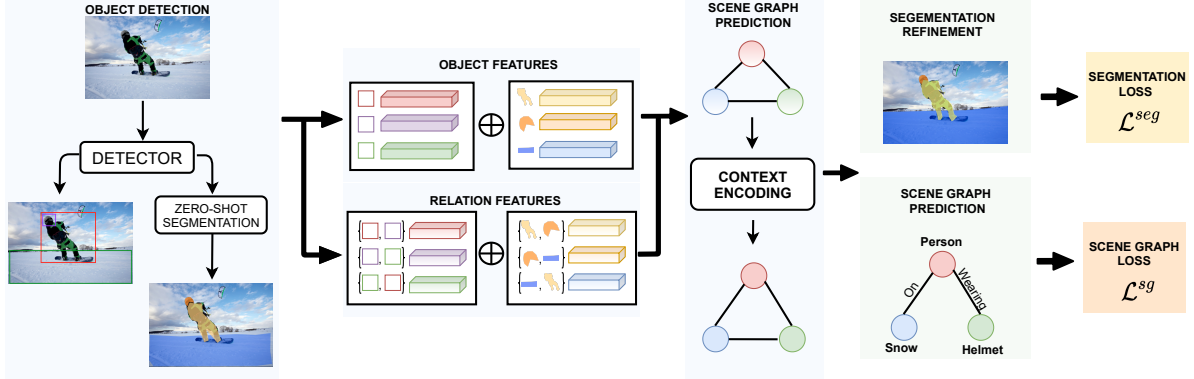


Figure 2. **Model Architecture.** For an image, the object detector provides a set of bounding boxes, and for each box, additionally generates instance-level segmentations via a *zero-shot* transfer mechanism. These inferred segmentation masks are incorporated into the nodes and edges of the underlying graph, before passing it into an existing scene graph prediction architecture like [55, 45]. The inferred segmentation masks are additionally refined by leveraging the global context captured by the context aggregation step of the scene graph prediction method. The proposed method is end-to-end trainable, and can be augmented to any existing scene graph method.

Bi-directional LSTM [55], Tree-LSTM [45], or Graph Attention Network [51], to obtain refined features  $\mathbf{z}_j^{o,g}$ . These refined features are used to obtain the object labels  $\mathbf{O}^g$  for the nodes within the graph  $\mathbf{G}^g$ . Similarly, for the relation network  $\text{Pr}(\mathbf{R}^g | \mathbf{O}^g, \mathbf{B}^g, \mathbf{x}^g)$ , features corresponding to union of object bounding boxes are refined using message passing layers and subsequently classified to produce predictions for relations.

Existing models ground the objects in the scene graph to rectangular regions in the image. While bounding boxes provides an approximate estimate of the object locations, having a more granular pixel-level grounding achievable through segmentation masks is much more desirable. A major challenge is the lack of segmentation annotation in scene graph datasets like Visual Genome [24]. Furthermore, manually labelling segmentation masks for such large datasets is both time consuming and expensive. As a solution, we derive segmentation masks via a *zero-shot* transfer mechanism from a segmentation head trained on an external dataset  $\mathcal{D}^m$  (e.g. MSCOCO [29]). This inferred segmentation mask is then used as additional input to the object and relation networks to generate better scene graphs. Our approach factorizes the distribution over  $\mathbf{G}^g$  as,

$$\Pr(\mathbf{G}^g | \mathbf{x}^g) = \Pr(\mathbf{B}^g | \mathbf{x}^g) \cdot \Pr(\mathbf{M}^g | \mathbf{x}^g) \cdot \Pr(\mathbf{O}^g | \mathbf{B}^g, \mathbf{M}^g, \mathbf{x}^g) \cdot \Pr(\mathbf{R}^g | \mathbf{O}^g, \mathbf{B}^g, \mathbf{M}^g, \mathbf{x}^g) \quad (2)$$

where  $\mathbf{M}^g = \{\mathbf{m}_1^g, \dots, \mathbf{m}_n^g\}$  are the inferred segmentation masks corresponding to the bounding boxes  $\mathbf{B}^g$ . Such a factorization enables grounding scene graphs to segmentation masks and affords easy integration to existing architectures.

### 2.3. Segmentation Mask Transfer

For each image  $\mathbf{x}^g \in \mathcal{D}^g$ , we derive segmentation masks  $\mathbf{M}^g$  using annotations learned over classes in an external dataset  $\mathcal{D}^m$ . To facilitate this, like described in Section 2.2,

we pretrain a standard object detector (like Faster R-CNN [39]) on the scene graph dataset  $\mathcal{D}^g$ . However, instead of training the detector just on images in  $\mathcal{D}^g$ , we additionally jointly learn a segmentation head  $f_M$  on images in  $\mathcal{D}^m$ . Note that when training the object detector jointly on images in  $\mathcal{D}^g$  and  $\mathcal{D}^m$ , the same backbone and proposal generators are used, thus reducing the memory overhead.

For an image  $\mathbf{x}^g \in \mathcal{D}^g$ , let  $\mathbf{z}_j^g$  be the feature representation for a bounding box  $\mathbf{b}_j^g \in \mathbf{B}^g$ . Let,  $\tilde{\mathbf{m}}_j^g = f_M(\mathbf{z}_j^g)$ , where  $\tilde{\mathbf{m}}_j^g \in \mathbb{R}^{d_m \times m \times m}$ ,  $d_m$  represents the number of classes in  $\mathcal{D}^m$ , and  $m$  is the spatial resolution of the mask. Per class segmentation masks  $\mathbf{m}_j^g \in \mathbb{R}^{d_g \times m \times m}$  are then derived from  $\tilde{\mathbf{m}}_j^g$  using a *zero-shot* transfer mechanism<sup>3</sup>. Let  $\mathbf{S} \in \mathbb{R}^{d_g \times d_m}$  be a matrix that captures linguistic similarities between classes in  $\mathcal{D}^g$  and  $\mathcal{D}^m$ . For a pair of classes  $c_g \in [1, d_g]$ ,  $c_m \in [1, d_m]$ , the element  $\mathbf{S}_{c_g, c_m}$  is defined as,

$$\mathbf{S}_{c_g, c_m} = \mathbf{g}_{c_g}^\top \mathbf{g}_{c_m} \quad (3)$$

where  $\mathbf{g}_{c_g}$  and  $\mathbf{g}_{c_m}$  are 300-dimensional GloVe [37] vector embeddings for classes  $c_g$  and  $c_m$  respectively<sup>4</sup>.  $\mathbf{m}_j^g$  is then obtained as a linear combination over  $\tilde{\mathbf{m}}_j^g$  as follows,

$$\mathbf{m}_j^g = \mathbf{S}^\top \tilde{\mathbf{m}}_j^g \quad (4)$$

Note that such a transfer doesn't require *any* additional labelling cost as we rely on a publicly available dataset  $\mathcal{D}^m$ .

### 2.4. Grounding Nodes to Segmentation Masks

As mentioned in Equation 2, we incorporate the inferred segmentation masks in the object network

<sup>3</sup>Note that  $d_g \gg d_m$  in our case.

<sup>4</sup>For class names that contain multiple words, individual GloVe word embeddings are averaged.

$\Pr(\mathbf{O}^g|\mathbf{B}^g, \mathbf{M}^g, \mathbf{x}^g)$  to ground objects in  $\mathcal{D}^g$  to pixel-level regions within the image.

Specifically, for a particular image  $\mathbf{x}^g$ , the model  $\Pr(\mathbf{B}^g|\mathbf{x}^g)$  outputs a set of bounding boxes  $\mathbf{B}^g$ . For each bounding box  $\mathbf{b}_j^g \in \mathbf{B}^g$ , it additionally computes a feature representation  $\mathbf{z}_j^g$  and object label probabilities  $\mathbf{l}_j^g \in \mathbb{R}^{d_g+1}$ . Following the procedure described in Section 2.3, per-class segmentation masks  $\mathbf{m}_j^g$  are inferred for each bounding box  $\mathbf{b}_j^g$ . We define a segmentation aware representation  $\hat{\mathbf{z}}_j^g$  as,

$$\hat{\mathbf{z}}_j^g = f_{\mathbf{N}}([\mathbf{z}_j^g, \mathbf{m}_j^g]) \quad (5)$$

where  $f_{\mathbf{N}}$  is a learned network and  $[\cdot, \cdot]$  represents concatenation. Contrary to existing methods like [45, 55] that use the segmentation agnostic representation  $\mathbf{z}_j^g$ , we feed  $\hat{\mathbf{z}}_j^g$  and  $\mathbf{l}_j^g$  as inputs to the object network  $\Pr(\mathbf{O}^g|\mathbf{B}^g, \mathbf{M}^g, \mathbf{x}^g)$ .

## 2.5. Grounding Edges to Segmentation Masks

To facilitate better relation prediction, we leverage the inferred segmentation masks in the relation network  $\Pr(\mathbf{R}^g|\mathbf{O}^g, \mathbf{B}^g, \mathbf{M}^g, \mathbf{x}^g)$ . Specifically, for a pair of objects, we utilize a Gaussian masking mechanism to identify relation identifying pixel-level regions within an image.

Given a pair of bounding boxes  $(\mathbf{b}_j^g, \mathbf{b}_{j'}^g) \in \mathbf{B}^g$  that contain a possible edge and their corresponding object label probabilities  $(\mathbf{l}_j^g, \mathbf{l}_{j'}^g)$ , their respective segmentation masks  $(\mathbf{m}_j^g, \mathbf{m}_{j'}^g)$  are computed via the procedure described in Section 2.3. We define  $\mathbf{z}_{j,j'}^g$  as the segmentation agnostic feature representation representing the union of boxes  $(\mathbf{b}_j^g, \mathbf{b}_{j'}^g)$ , which is computed as  $\text{RoIAlign}(\mathbf{x}^g, \mathbf{b}_j^g \cup \mathbf{b}_{j'}^g)$ <sup>5</sup>.

Contrary to existing works that rely on this coarse rectangular union box, our approach additionally incorporates an intersection of the segmentation masks  $(\mathbf{m}_j^g, \mathbf{m}_{j'}^g)$  to provide more granular information. To this end, we define the union segmentation mask  $\mathbf{m}_{j,j'}^g$  as,

$$\mathbf{m}_{j,j'}^g = (\mathbf{K}_j \otimes \mathbf{m}_j^g) \odot (\mathbf{K}_{j'} \otimes \mathbf{m}_{j'}^g) \quad (6)$$

where  $\otimes$  is the convolution operation, and  $\odot$  computes an element-wise product.  $\mathbf{K}_j, \mathbf{K}_{j'}$  are  $\delta \times \delta$  sized Gaussian smoothing spatial convolutional filters parameterized by variances  $\sigma_x^2, \sigma_y^2$  and correlation  $\rho_{x,y}$ . These parameters are obtained by learning a transformation over the object label probabilities  $\mathbf{l}_j^g$ . Specifically,  $\sigma_x^2, \sigma_y^2, \rho_{x,y} = f_{\mathbf{N}}(\mathbf{l}_j^g)$ , where  $f_{\mathbf{N}}$  is a learned network.  $\mathbf{K}_{j'}$  is computed analogously using  $\mathbf{l}_{j'}^g$ . The attended union segmentation mask  $\mathbf{m}_{j,j'}^g$  affords the computation of a segmentation aware representation  $\hat{\mathbf{z}}_{j,j'}^g$  as follows,

$$\hat{\mathbf{z}}_{j,j'}^g = f_{\mathbf{E}}([\mathbf{z}_{j,j'}^g, \mathbf{m}_{j,j'}^g]) \quad (7)$$

where  $f_{\mathbf{E}}$  is a learned network.  $\hat{\mathbf{z}}_{j,j'}^g$  is then used as an input to the relation network  $\Pr(\mathbf{R}^g|\mathbf{O}^g, \mathbf{B}^g, \mathbf{M}^g, \mathbf{x}^g)$ .

<sup>5</sup> $\mathbf{b}_j^g \cup \mathbf{b}_{j'}^g$ , computes the convex hull of the union of the two boxes.

## 2.6. Refining Segmentation Masks

As described previously, our proposed approach incorporates segmentation masks to improve relation prediction. However, we posit that the tasks of segmentation and relation prediction are indelibly connected, wherein an improvement in one leads to an improvement in the other.

To this end, for each object  $\mathbf{b}_j^g \in \mathbf{B}^g$ , in addition to predicting the object labels  $\mathbf{O}^g$ , we learn a segmentation *refinement* head  $f_{\mathbf{M}'}$  to refine the inferred segmentation masks  $\mathbf{m}_j^g$ . However, as the scene graph dataset  $\mathcal{D}^g$  does not contain any instance-level segmentation annotations, training  $f_{\mathbf{M}'}$  in a traditionally supervised manner is challenging.

To alleviate this issue, we again leverage the auxiliary dataset  $\mathcal{D}^m$ , which contains segmentation annotations. For an image  $\mathbf{x}^m \in \mathcal{D}^m$  we compute the bounding boxes  $\mathbf{B}^m$ . Note that this does not require any additional training as the object detector is jointly trained using both  $\mathcal{D}^g$  and  $\mathcal{D}^m$  as described in Section 2.3. For a bounding box  $\mathbf{b}_j^m \in \mathbf{B}^m$ , the corresponding per class masks are computed as,  $\mathbf{m}_j^m = f_{\mathbf{M}}(\mathbf{z}_j^m)$ , where  $\mathbf{z}_j^m$  is the feature representation for  $\mathbf{b}_j^m$ , and  $f_{\mathbf{M}}$  is the segmentation head defined in Section 2.3. The refined mask  $\hat{\mathbf{m}}_j^m$  is then computed as,

$$\hat{\mathbf{m}}_j^m = \mathbf{m}_j^m + f_{\mathbf{M}'}(\mathbf{z}_j^{o,m}) \quad (8)$$

where  $\mathbf{z}_j^{o,m}$  is the representation computed by the context aggregation layer within the object network  $\Pr(\mathbf{O}^m|\mathbf{B}^m, \mathbf{M}^m, \mathbf{x}^m)$ . Note that this network is identical to the one defined in Equation 2. The segmentation *refinement* head  $f_{\mathbf{M}'}$  is a zero-initialized network that learns a residual update over the mask  $\mathbf{m}_j^m$ . As ground-truth segmentation annotations are available for all objects  $\mathbf{B}^m$ ,  $f_{\mathbf{M}'}$  is trained using a pixel-level cross entropy loss.

$f_{\mathbf{M}'}$  is trained alongside the scene graph generation model, and the refined masks are used during inference to improve relation prediction performance. Specifically, for a particular image  $\mathbf{x}^g \in \mathcal{D}^g$ , we follow the model described in Equation 2 to generate predictions. However, instead of directly using the inferred masks obtained using the zero-shot formulation in Section 2.3, we additionally refine it using  $f_{\mathbf{M}'}$ . For a particular mask  $\mathbf{m}_j^g$  corresponding to a bounding box  $\mathbf{b}_j^g$ , we compute  $\hat{\mathbf{m}}_j^g$  as,

$$\hat{\mathbf{m}}_j^g = \mathbf{m}_j^g + f_{\mathbf{M}'}(\mathbf{z}_j^{o,g}) \quad (9)$$

where  $\mathbf{z}_j^{o,g}$  is the representation computed by the context aggregation layer. The refined mask is used in the object and relation networks as described in Sections 2.4 and 2.5.

## 2.7. Training

Our proposed approach is trained in two stages. The first stage involves pre-training the object detector to enable bounding box proposal generation for a given image. Given



datasets  $\mathcal{D}^g$  and  $\mathcal{D}^m$ , the object detector is jointly trained to minimize the following objective,

$$\mathcal{L}^{obj} = \mathcal{L}^{rcnn} + \mathcal{L}^{seg} \quad (10)$$

where  $\mathcal{L}^{rcnn}$  is the Faster R-CNN [39] objective, and  $\mathcal{L}^{seg}$  is the pixel-level binary cross entropy loss [11] applied over segmentation masks. Note that images in  $\mathcal{D}^g$  do not contribute to  $\mathcal{L}^{seg}$  due to lack of segmentation annotations.

The second stage of training involves training the scene graph generation network to accurately identify relations between pairs of objects. Given datasets  $\mathcal{D}^g$  and  $\mathcal{D}^m$ , the scene graph generation network is jointly trained to minimize the following objective,

$$\mathcal{L} = \mathcal{L}^{sg} + \mathcal{L}^{seg} \quad (11)$$

where  $\mathcal{L}^{sg}$  depends on the architecture of the underlying scene graph method our approach is augmented to. For example, in the case of MOTIF [55],  $\mathcal{L}^{sg}$  consists of two cross-entropy losses, one to refine the object categorization obtained from the pretrained detector, and the other to aide with accurate relation prediction.  $\mathcal{L}^{seg}$  is identical to the segmentation loss described in Equation 10, and is used to learn the refinement network  $f_{M'}$  (Section 2.6). As images in  $\mathcal{D}^m$  do not contain scene graph annotations, they only contribute to  $\mathcal{L}^{seg}$ . Similarly, images in  $\mathcal{D}^m$  only affect  $\mathcal{L}^{sg}$ .

### 3. Experiments

We perform experiments using two datasets: the Visual Genome Dataset [24] and the COCO dataset [29].

**Visual Genome.** For training and evaluating the scene graph generation performance, we use the Visual Genome dataset [24]. We use the widely adopted preprocessed version of Visual Genome from [50]. This subset contains 108k images across 150 object categories and 50 relation labels. Images with more than 40 object bounding boxes are filtered out from the test set due to memory constraints.

**MS-COCO** For training and evaluating the segmentation masks, we use the MSCOCO 2017 dataset [29], which contains 123k images, split into 118k training and 5k validation images, across 80 categories. As the ground-truth annotations for the test set are not available, as is common practice, results are reported on the validation set.

Note that our approach is *agnostic* to the choice of the auxiliary dataset  $\mathcal{D}^m$ . The choice of using MS-COCO is motivated by its popularity in the community. As MS-COCO has images in common with Visual Genome, there is a possibility of information leakage across the two datasets. However, due to the differences in annotation types, such leakage is *not* observed in practice. A further analysis is presented in Section B of the **supplementary**. For simplicity, this image overlap is not removed when computing the results described in Section 4 (performance without overlap is reported in **supplementary** Tables A4, A5).

### 3.1. Scene Graph Generation Model

Our proposed framework is generic and can be easily integrated with various scene graph generation models. In this work we experiment with two scene graph architectures, namely MOTIF [55] and VCTree [45].

In MOTIF [55], the object and relation networks (Equation 1) are each instantiated by bidirectional LSTMs [12]. For an image  $x_j^g \in \mathcal{D}^g$ , the extracted bounding boxes  $\mathbf{B}^g$  are arranged based on their  $x$ -coordinate position, and passed through the bidirectional LSTM networks. Instead of assuming a linear ordering between the objects, VCTree [45] generates a dynamic binary tree, with the aim of explicitly encoding the parallel and hierarchical relationships between objects. The object and relation networks are instantiated as bidirectional TreeLSTMs [43]. When augmenting our approach with MOTIF [55] and VCTree [45], we identically replicate the object and relation networks proposed in the respective works. Additional details are provided in Section A of the **supplementary**.

### 3.2. Evaluation

**Relationship Recall (RR).** To measure the performance of models we use the **mean Recall @K (mR@K)** metric introduced in [4, 45]. The mean Recall metric calculates the recall for predicate label independently across all images and then averages the result. We report the mean Recall instead of the conventional Regular Recall(R@K) due to the long-tail nature of relation labelling in Visual Genome that leads to reporting bias [44]. Mean Recall reduces the influence of dominant relationships such as `on` and `has`, and gives equal weight to all the labels in the dataset.

**Zero-Shot Recall (zsR@K).** Introduced in [33], zsR@K computes the Recall@K for subject-predicate-object triplets that are not present in the training data.

These evaluation metrics are computed for three different sub-tasks: 1) Predicate Classification (**PredCls**): predict the relation labels given the ground truth object bounding boxes and labels; 2) Scene Graph Classification (**SGCls**): predict the object and relation labels given the ground truth object bounding boxes; 3) Scene Graph Detection (**SGDet**): given an image, predict the entire scene graph.

**Segmentation Precision.** As the Visual Genome dataset [24] does not contain any instance-level segmentation annotations, as a proxy we use the MSCOCO dataset [29] to measure the performance of the segmentation refinement procedure described in Section 2.6. To make the evaluation similar to scene graph generation, we analogously define three sub-tasks to measure the improvement in the quality of segmentation masks. These sub-tasks, namely (**PredCls**), (**SGCls**), and (**SGDet**), are identical to the ones defined earlier. For each of these sub-tasks, the standard evaluation metrics on COCO are reported [11].

Model	Detector	Method	Predicate Classification			Scene Graph Classification			Scene Graph Generation		
			mR@20	mR@50	mR@100	mR@20	mR@50	mR@100	mR@20	mR@50	mR@100
IMP [50]	VGG-16 [41]	-	-	9.8	10.5	-	5.8	6.0	-	3.8	4.8
MOTIF [55]	VGG-16 [41]	-	10.8	14.0	15.3	6.3	7.7	8.2	4.2	5.7	6.6
VCTree [45]	VGG-16 [41]	-	14.0	17.9	19.4	8.2	10.1	10.8	5.2	6.9	8.0
MOTIF <sup>†</sup>	VGG-16 [41]	Baseline	13.7	17.5	18.9	7.5	9.2	9.8	5.2	6.8	7.9
		Seg-Grounded	<b>14.6</b>	<b>18.7</b>	<b>20.3</b>	<b>7.9</b>	<b>9.8</b>	<b>10.5</b>	<b>5.6</b>	<b>7.3</b>	<b>8.1</b>
		Baseline	14.1	18.0	19.4	8.0	9.9	10.6	5.8	7.7	9.0
VCTree <sup>†</sup>	ResNeXt-101-FPN [34, 49]	Baseline	14.4	18.4	19.8	8.1	9.9	10.7	4.4	5.7	6.4
		Seg-Grounded	<b>14.8</b>	<b>18.9</b>	<b>20.5</b>	<b>8.7</b>	<b>10.8</b>	<b>11.6</b>	<b>5.3</b>	<b>7.0</b>	<b>7.8</b>
		Baseline	13.7	17.4	19.0	8.1	9.9	10.6	5.3	6.9	7.9
VCTree <sup>†</sup>	ResNeXt-101-FPN [34, 49]	Baseline	14.4	18.4	19.8	8.1	9.9	10.7	4.4	5.7	6.4
		Seg-Grounded	<b>14.8</b>	<b>18.9</b>	<b>20.5</b>	<b>8.7</b>	<b>10.8</b>	<b>11.6</b>	<b>5.3</b>	<b>7.0</b>	<b>7.8</b>
		Baseline	13.7	17.4	19.0	8.1	9.9	10.6	5.3	6.9	7.9

Table 1. **Scene Graph Prediction on Visual Genome.** Mean Recall (mR) is reported for three tasks, across two detector backbones. Our approach is augmented to and contrasted against MOTIF [55] and VCTree [45]. † denotes our re-implementation of the methods.

Detector	Method	Predicate Classification						Scene Graph Classification						Scene Graph Generation					
		AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
VGG-16 [41]	No Refine	31.5	63.8	28.1	21.8	36.4	43.9	32.5	58.9	31.8	17.0	35.3	42.3	23.2	44.7	21.6	8.1	26.0	35.1
	MOTIF <sup>†</sup> + Refine	<b>42.4</b>	<b>78.1</b>	<b>40.9</b>	<b>33.0</b>	<b>46.6</b>	<b>55.8</b>	<b>37.5</b>	<b>63.5</b>	<b>38.8</b>	<b>21.0</b>	<b>40.7</b>	<b>48.4</b>	24.7	45.8	23.9	8.6	27.9	38.1
	VCTree <sup>†</sup> + Refine	41.9	77.6	40.3	32.8	46.1	55.2	37.4	63.4	38.6	20.9	40.5	48.4	<b>24.9</b>	<b>46.1</b>	<b>24.1</b>	<b>8.6</b>	<b>28.1</b>	<b>38.4</b>
ResNeXt-101-FPN [34, 49]	No Refine	54.8	87.6	58.3	46.3	57.8	68.1	51.6	76.7	56.9	37.9	53.7	62.2	39.2	61.2	42.4	<b>20.0</b>	42.3	55.7
	MOTIF <sup>†</sup> + Refine	<b>59.3</b>	<b>90.6</b>	<b>64.7</b>	<b>52.0</b>	<b>62.2</b>	<b>70.6</b>	<b>54.6</b>	<b>78.2</b>	<b>61.1</b>	<b>41.1</b>	<b>56.8</b>	<b>64.1</b>	39.2	61.2	42.4	19.9	42.3	<b>55.8</b>
	VCTree <sup>†</sup> + Refine	59.0	90.4	64.2	51.7	62.0	70.4	54.3	77.9	60.4	41.0	56.4	63.8	39.2	61.2	42.4	19.9	42.3	55.7

Table 2. **Segmentation Refinement on MSCOCO.** Standard COCO precision metrics are reported across three tasks and two detector backbones. Task formulation is identical to Table 1. ‘No Refine’ is the baseline where the segmentation masks are obtained from the pre-trained detector. As ground truth masks are unavailable in Visual Genome, evaluation on MSCOCO serves as a proxy.

### 3.3. Implementation Details

**Detector.** For our detector architecture, we use the two-stage Faster-RCNN [39] framework. To demonstrate the flexibility of our approach, we experiment with two different backbones within the Faster-RCNN framework: 1) VGG-16 [41] pre-trained on the ImageNet [40] dataset, and 2) ResNeXt-101-FPN [34, 49] backbone pre-trained on the MSCOCO [29] dataset. We first fine-tune the detector jointly on the Visual Genome and MSCOCO datasets, refining the classifiers and regressors, and simultaneously learning a segmentation network on images in MSCOCO. When training the scene graph generators, the detector parameters are frozen. Note that for the baselines, the detector is fine-tuned only on the Visual Genome, and hence no segmentation is learned.

**Scene Graph Models.** For training the scene graph models we use an SGD optimizer with an initial learning rate of  $10^{-2}$ . Following prior works, we integrate the frequency bias [55] into the training and inference procedure. During inference, in SGDet task, we filter pairs of objects that do not have any bounding box overlap for relation prediction.

## 4. Results

**Relationship Recall.** We report the mean Recall values comparing the baseline and proposed method in Table 1. To

ensure a fair comparison, we additionally report the numbers obtained via re-implementing the baselines. Note that in case of MOTIF [55], our re-implementation provides significantly higher performance compared to the reported numbers in [55]. For both the MOTIF [55] and VCTree [45], irrespective of the backbone architecture, we observe a consistent improvements in the recall rate across all three tasks when incorporating our proposed approach.

For MOTIF [55], we observe an improvement of 7.0% on average at mR@20, 50 and 100 across all settings and backbones. Specifically, on the VGG backbone [41], we obtain a relative improvement of 6.5%, 5.3%, and 7.7% on mr@20 across the three tasks. Similarly, for the ResNeXt-101-FPN [34, 49] backbone, we observe relative improvements of 2.8%, 11.2%, and 10.3% on mr@20. Similarly for VCTree [45], an average improvement of 12.6% is observed across tasks and backbones. We attribute the performance improvement to the ability of our model to effectively ground objects and relations to pixel-level regions, thus providing more discriminative features. We provide additional results and individual relation recall comparisons in Section C of the **supplementary**.

**Zero-Shot Recall.** We report the Zero-Shot Recall value zsR@20 and zsR@100 in Table 3. We observe a consistent improvement in zero-shot recall when using the pro-

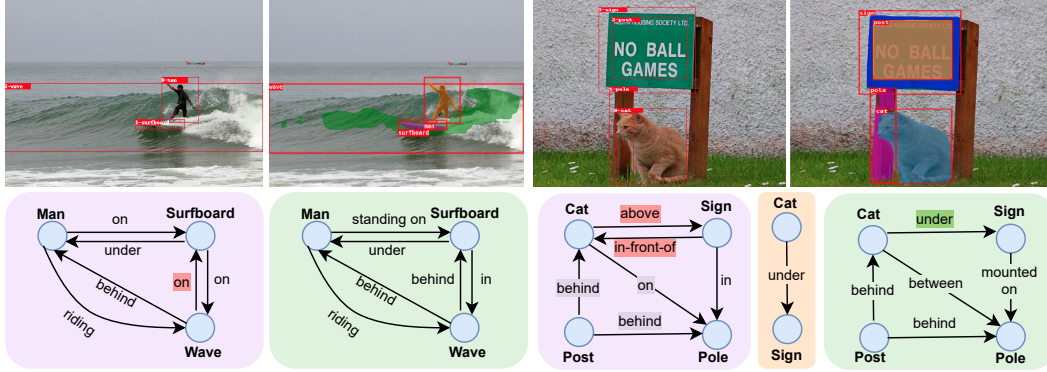


Figure 3. **Qualitative Results.** Visualizations of scene graphs generated by using VCTree [45] (in purple) and our approach augmented to VCTree (in green). The left two images contrast performance on relation retrieval. The right two images contrast performance on zero-shot relation retrieval, with the zero-shot triplet shown in yellow. Our approach additionally generates pixel-level object groundings.

Model	Detector	Method	PredCls	SGCls	SGDet
			zsr@20/100	zsr@20/100	zsr@20/100
MOTIF <sup>†</sup>	VGG-16 [41]	BL	1.7/6.7	0.2/1.1	0.0/0.4
		SG	<b>3.2/9.3</b>	<b>0.4/1.6</b>	<b>0.1/0.5</b>
	ResNeXt-101-FPN [34, 49]	BL	1.9/7.2	0.3/1.2	0.0/0.5
		SG	<b>4.1/10.5</b>	<b>0.8/2.5</b>	<b>0.1/1.0</b>
VCTree <sup>†</sup>	VGG-16 [41]	BL	1.8/7.3	0.6/1.8	0.1/0.5
		SG	<b>3.5/10.2</b>	<b>0.7/2.4</b>	<b>0.3/0.9</b>
	ResNeXt-101-FPN [34, 49]	BL	1.8/7.1	0.4/1.2	0.1/0.7
		SG	<b>4.3/10.6</b>	<b>0.8/2.5</b>	<b>0.3/1.5</b>

Table 3. **Zero-Shot Recall on Visual Genome.** Results are reported for three tasks across two detector backbones. Our approach is augmented to and contrasted against MOTIF [55] and VCTree [45]. <sup>†</sup> denotes our re-implementation of the methods.

posed scene graph generation framework. Our method outperforms the baselines by an average of 94.5% and 97.9% on MOTIF and VCTree respectively.

**Segmentation Accuracy.** As segmentation annotations are not present in Visual Genome [24], we evaluate our proposed segmentation refinement on the MSCOCO dataset [29]. This provides a suitable proxy, wherein segmentation improvements on the MSCOCO dataset can be translated, to some degree, on Visual Genome. We report the standard COCO evaluation metrics, namely AP (averaged over IoU thresholds),  $AP_{50}$ ,  $AP_{75}$ , and  $AP_S$ ,  $AP_M$ ,  $AP_L$  (AP at different scales), on three different scene graph evaluation tasks in Table 2. ‘No Refine’ acts as a strong baseline, wherein the segmentation masks are generated from the pre-trained detector. It is evident that our proposed segmentation refinement improves on mask quality across tasks and detector backbones. As the ground truth bounding boxes and labels are available for the Predicate Classification task, the observed improvement here is the largest (34.6% higher AP on VGG). Analogously, the observed improvement on the Scene Graph Generation (SGDet) task is the lowest (7.3% higher AP on VGG) as any errors made by the pre-trained detector are forwarded to the scene graph network. To further highlight the effectiveness of our joint training approach, we report per-class improvement on AP in Sec-

Ablation	Scene Graph Classification			
	mR@20	mR@50	mR@100	zsr@20/100
Base	8.1	9.9	10.6	0.3/1.2
Joint	8.5	10.5	11.1	0.4/1.5
Joint + OG	9.0	11.1	11.8	0.6/2.1
Joint + OG + $EG_{avg}$	9.1	11.4	12.2	0.8/2.4
Joint + OG + $EG_{union}$	9.1	11.3	12.2	0.7/2.4
Joint + OG + $EG_{Gaussian}$	9.3	11.5	12.2	0.7/2.3
Final Model	<b>9.4</b>	<b>11.6</b>	<b>12.3</b>	<b>0.8/2.5</b>

Table 4. **Ablation.** Mean Recall (mR) and Zero-shot Recall (zsr) are reported. VCTree [45] is the base architecture for all methods. Please refer Section 4 for model definitions.

tion C of the **supplementary**. Note that no noticeable improvement is observed for the SGDet task when using the ResNeXt-101-FPN [34, 49] backbone. We believe this is a direct consequence of the backbone using feature pyramid networks (FPNs) [28] to extract features. As FPNs effectively capture global context using lateral connections, the detector provides much richer object representations. This makes the context aggregation in the scene graph network redundant, making refining segmentation masks harder.

**Ablation.** We conduct an ablation study over the various components in our model using VCTree. All models are trained with the ResNeXt-101-FPN [34, 49] backbone. The results on the **SGCls** task are shown in Table 4. ‘Base’ is defined as the vanilla VCTree model learned over the detector trained only on the Visual Genome dataset. To understand the effect our joint detector pre-training has on the overall performance, we define ‘Joint’ as the VCTree model learned over this jointly pre-trained detector. It can be seen that just the joint pre-training of the detector provides considerable improvements (5% on mR@20).

We incrementally add components of our proposed approach to the ‘Joint’ detector to better highlight their importance. ‘Joint + OG’ is defined as the model that uses the jointly trained detector and the object grounding mechanism described in Section 2.4. Similarly, ‘Joint + OG +  $EG_x$ ’ describes the model that additionally uses our pro-

posed relation grounding mechanism defined in Section 2.5. The subscript  $x$  in  $EG_x$  refers to the type of masking mechanism used to combine the segmentation masks for a pair of objects. We experiment with averaging (*avg*), taking the logical or (*union*), and the proposed Gaussian masking (*Gaussian*). Finally, our complete model with the additional segmentation mask refinement (Section 2.6) is denoted as ‘Final Model’. From Table 4 it can be seen that using both object and relation grounding helps with performance, and using a Gaussian masking mechanism is superior to other alternatives. Additionally, fine-tuning the segmentation masks not only helps improve its quality, but also provides better scene graph generation performance.

**Qualitative Results.** We qualitatively contrast the performance of the VCTree model [45] augmented with our proposed approach against its vanilla counterpart in Figure 3. The two images on the left show results from the relation retrieval task. Our approach (in green) predicts more granular and spatially informative relationships `standing on` and `behind`, as opposed to the baseline (in purple) which is heavily biased towards the more common and less informative relation `on`. The two images on the right highlight the ability of our approach to generalize in zero-shot scenarios. As the triplets of `cat` with `sign` are absent from the training dataset, the baseline approach (in purple) defaults to predicting incorrect relations of `above` and `in-front-of`. On the contrary, our approach accurately predicts the correct relation `under`.

## 5. Related Work

**Scene Graph Generation.** Scene graph generation is a popular topic in the vision community [23, 26, 30, 36, 44, 50, 51, 53, 54, 55]. First introduced in [50], they proposed an iterative message passing module to refine the features of node and edge prior to classification. Subsequent works proposed use of different architectures such as Bi-directional LSTM [55], Tree-LSTM [45], Graph Neural Networks [51] and novel message passing algorithm [27, 38] for representation learning. While improved context aggregation can lead to better scene graph performance, recent works have focused more on alleviating issue arising from long tail distribution of relation labels. Tang *et al.* [44] propose the use of a causal inference framework to debias the prediction of model obtained from biased training. Knyazev *et al.* [23], proposed a graph density aware loss to address the imbalance in the Visual Genome dataset.

**Zero-Shot Segmentation.** Zero-shot learning is an active area of research in computer vision [9, 48]. The sub-field of zero-shot segmentation, however, is relatively recent [2, 13, 14, 20, 21, 58]. A majority of work in this area has been on zero-shot semantic segmentation [2, 13, 20, 47, 58], where the aim is correct categorize each pixel in an image. Zhao *et al.* [58] propose the open-vocabulary scene parsing

task, wherein hypernym/hyponym relations from WordNet [35] are leveraged to build label relationships, and subsequently segment classes. Bucher *et al.* [2] combine visual-semantic embeddings along with a generative model and classifier to obtain masks for unseen classes. Kato *et al.* [20] leverage a semantic to visual variational mapping over class labels, along with a data driven distance metric, to generate zero-shot segmentation masks. Hu *et al.* [13] introduce uncertainty aware losses to mitigate the effect of noisy training examples for robust semantic segmentation. For the task of zero-shot instance-level segmentation, Khandelwal *et al.* [21] leverage linguistic and visual similarities to learn a transformation over segmentation heads from classes with abundant annotations to classes with zero/few annotations.

**Multi-task Learning.** Multi-task Learning (MTL) involves optimizing for several tasks simultaneously and transferring information across task for improved performance [3]. Most methods in MTL [5, 32, 57] learn a shared representation of layers along with multiple independent classifiers. Another line of work involves explicitly modelling the the relation between tasks, either by grouping [17, 19, 25] or in the form of task-covariance [6, 8, 56].

## 6. Conclusion

We present a novel model-agnostic framework for segmentation grounded scene graph generation. Contrary to tradition scene graph generation frameworks that grounds object in a scene graph to bounding boxes, our proposed methodology allows for a more granular pixel-level grounding, obtained via a zero-shot transfer mechanism. Our proposed framework leverages these groundings to provide significant improvements across various scene graph prediction tasks, irrespective of the architecture it is augmented to. Finally, we highlight the benefits of simultaneously optimizing the tasks of scene graph and segmentation generation, which leads to improved performance on both.

## Acknowledgments and Disclosure of Funding

This work was funded, in part, by the Vector Institute for AI, Canada CIFAR AI Chair, NSERC CRC and an NSERC DG and Accelerator Grants. This material is based upon work supported by the US Air Force Research Laboratory (AFRL) under the DARPA Learning with Less Labels (LwLL) program (Contract No.FA8750-19-C-0515)<sup>6</sup>. Hardware resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute<sup>7</sup>. Additional support was provided by JELF CFI grant and Compute Canada under the RAC award. Finally, we sincerely thank Bicheng Xu for valuable feedback on the paper draft.

<sup>6</sup>Views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

<sup>7</sup>[www.vectorinstitute.ai/#partners](http://www.vectorinstitute.ai/#partners)



## References

- [1] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What’s the point: Semantic segmentation with point supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 549–565, 2016.
- [2] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [3] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, July 1997.
- [4] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6163–6171, 2019.
- [5] Xiao Chu, Wanli Ouyang, Wei Yang, and Xiaogang Wang. Multi-task recurrent neural network for immediacy prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3352–3360, 2015.
- [6] Carlo Ciliberto, Youssef Mroueh, Tomaso Poggio, and Lorenzo Rosasco. Convex learning of multiple tasks and their structure. In *International Conference on Machine Learning (ICML)*, pages 1548–1557, 2015.
- [7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, Jan. 2015.
- [8] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117, 2004.
- [9] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.
- [10] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10323–10332, 2019.
- [11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.
- [12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [13] Ping Hu, Stan Sclaroff, and Kate Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.
- [14] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2018.
- [15] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
- [16] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [17] Laurent Jacob, Francis Bach, and Jean-Philippe Vert. Clustered multi-task learning: A convex formulation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.
- [18] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10236–10247, 2020.
- [19] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning (ICML)*, 2011.
- [20] Naoki Kato, Toshihiko Yamasaki, and Kiyoharu Aizawa. Zero-shot semantic segmentation via variational mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.
- [21] Siddhesh Khandelwal, Raghav Goyal, and Leonid Sigal. Unit: Unified knowledge transfer for any-shot object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5951–5961, 2021.
- [22] A. Khoreva, A. Rohrbach, and B. Schiele. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision (ACCV)*, 2018.
- [23] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Graph density-aware losses for novel compositions in scene graph generation. *British Machine Vision Conference (BMVC)*, 2020.
- [24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)*, 123(1):32–73, 2017.
- [25] Abhishek Kumar and Hal Daume III. Learning task grouping and overlap in multi-task learning. *International Conference on Machine Learning (ICML)*, 2012.
- [26] Yikang Li, Wanli Ouyang, Zhou Bolei, Shi Jianping, Zhang Chao, and Xiaogang Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [27] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–351, 2018.
- [28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

- [29] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.
- [30] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [31] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [32] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. Learning multiple tasks with multilinear relationship networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [33] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 852–869, 2016.
- [34] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. <https://github.com/facebookresearch/maskrcnn-benchmark>, 2018.
- [35] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [36] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2171–2180, 2017.
- [37] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [38] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3957–3966, 2019.
- [39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015.
- [40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.
- [42] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13936–13945, 2021.
- [43] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics.
- [44] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3716–3725, 2020.
- [45] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6619–6628, 2019.
- [46] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [47] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8256–8265, 2019.
- [48] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning—the good, the bad and the ugly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4582–4591, 2017.
- [49] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017.
- [50] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5419, 2017.
- [51] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.
- [52] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10685–10694, 2019.
- [53] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 606–623, 2020.

- [54] Alireza Zareian, Zhecan Wang, Haoxuan You, and Shih-Fu Chang. Learning visual commonsense for robust scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 642–657, 2020.
- [55] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5831–5840, 2018.
- [56] Yi Zhang and Jeff Schneider. Learning multiple tasks with a sparse matrix-normal penalty. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 6, page 2, 2010.
- [57] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 94–108, 2014.
- [58] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2002–2010, 2017.