Supplementary: Segmentation-grounded Scene Graph Generation

Siddhesh Khandelwal*,1,2 Mohammed Suhail*,1,2 Leonid Sigal[1,2,3]
[1]Department of Computer Science, University of British Columbia
[2]Vector Institute for AI [3]CIFAR AI Chair
skhandel@cs.ubc.ca suhail33@cs.ubc.ca lsigal@cs.ubc.ca

A. Augmenting to Existing Approaches

As described in Section 2 of the main paper, our proposed approach can be easily integrated to any existing scene graph generation architecture. In the experimental analysis (Section 3), we demonstrate this by incorporating our approach with MOTIF [55] and VCTree [45]. In this section we present details on how these are implemented.

For an input image $\mathbf{x}^g \in \mathcal{D}^g$, existing approaches like [55, 45] use a pretrained object detector, like Faster R-CNN [39], to generate proposal bounding boxes $\mathbf{B}^g$. For each proposal $\mathbf{b}_j^g \in \mathbf{B}^g$, the pretrained object detector also outputs a feature representation $\mathbf{z}_j^g$ (computed using `RoIAlign`) and object label probabilities $\mathbf{l}_j^g$. As described in Section 2.3 of the main paper, our approach additionally computes per class segmentation masks $\mathbf{m}_j^g$ for each bounding box $\mathbf{b}_j^g$.

Our approach uses $\mathbf{m}_j^g$ as an additional input to the object and relation networks, which differ in implementation depending on the model architecture.

A.1. Integration with MOTIF

MOTIF [55] leverages recurring substructures in images to generate accurate scene graphs. Therefore, to keep track of global context, [55] instantiates the object and relation networks using bidirectional LSTM networks.

Object Network. For the purposes of refining object labels, MOTIF [55] constructs contextualized representations based on the set of bounding boxes $\mathbf{B}^g$. Boxes in $\mathbf{B}^g$ are first sorted, ideally based on their x-coordinate position, into a linear sequence $[(\mathbf{b}_1^g, \mathbf{z}_1^g, \mathbf{l}_1^g), \ldots, (\mathbf{b}_n^g, \mathbf{z}_n^g, \mathbf{l}_n^g)]$. This sorted linear sequence is passed into a bidirectional LSTM,

$$\mathbf{C}^g = \text{biLSTM}\left(\left[\mathbf{z}_j^g; \mathbf{W}_{l1}\mathbf{l}_j^g\right]_{j=1,\ldots,n}\right) \tag{A1}$$

where $\mathbf{C}^g = \{\mathbf{c}_1^g, \ldots, \mathbf{c}_n^g\}$ are the set of object context representations for each bounding box in $\mathbf{B}^g$, and $\mathbf{W}_{l1}$ is a learned parameter matrix. Each $\mathbf{c}_j^g \in \mathbf{C}^g$ corresponds to the concatenation of the final hidden states of the bidirectional LSTM for each element.

Our proposed approach, instead of using the segmentation agnostic representations $\mathbf{z}_j^g$, utilizes the per class masks $\mathbf{m}_j^g$ to compute a segmentation aware representation $\hat{\mathbf{z}}_j^g$ for each bounding box $\mathbf{b}_j^g$. These are obtained via a learned network $f_\mathbf{N}$ as described in Section 2.4 of the main paper. The segmentation aware context object representations $\hat{\mathbf{C}}^g = \{\hat{\mathbf{c}}_1^g, \ldots, \hat{\mathbf{c}}_n^g\}$ are then computed identically to Equation A1,

$$\hat{\mathbf{C}}^g = \text{biLSTM}\left(\left[\hat{\mathbf{z}}_j^g; \mathbf{W}_{l1}\mathbf{l}_j^g\right]_{j=1,\ldots,n}\right) \tag{A2}$$

A decoder LSTM then utilizes the object representations $\hat{\mathbf{C}}^g$ to sequentially generate labels for each bounding box as follows,

$$\begin{aligned}\hat{\mathbf{h}}_j^g &= \text{LSTM}_j\left(\left[\hat{\mathbf{c}}_j^g; \hat{\mathbf{o}}_{j-1}^g\right]\right) \\ \hat{\mathbf{o}}_j^g &= \text{argmax}\left(\mathbf{W}_o\hat{\mathbf{h}}_j^g\right)\end{aligned} \tag{A3}$$

where $\hat{\mathbf{o}}_j^g$ are one-hot class labels for object $\mathbf{b}_j^g$, and $\mathbf{W}_o$ is a learned parameter matrix.

Relation Network. Similar to the object network, MOTIF [55] generates a contextualized representation for objects for the purposes of relation prediction. Specifically, it computes the edge context representation $\mathbf{D}^g = \{\mathbf{d}_1^g, \ldots, \mathbf{d}_n^g\}$ using a bidirectional LSTM using the segmentation agnostic object context representations $\mathbf{C}^g$ and the subsequent object labels obtained from the decoding step of the object network.

Our proposed approach, on the other hand, uses segmentation aware context representations $\hat{\mathbf{C}}^g$ and object labels $\hat{\mathbf{O}}^g = \{\hat{\mathbf{o}}_1^g, \ldots, \hat{\mathbf{o}}_n^g\}$ in the relation network. Specifically, we compute the segmentation aware edge context representation $\hat{\mathbf{D}}^g$ as follows,

$$\hat{\mathbf{D}}^g = \text{biLSTM}\left(\left[\hat{\mathbf{c}}_j^g; \mathbf{W}_{l2}\hat{\mathbf{o}}_j^g\right]_{j=1,\ldots,n}\right) \tag{A4}$$

where $\mathbf{W}_{l2}$ is a learned parameter matrix.

In order to predict a possible relation between a pair of boxes $(\mathbf{b}_j^g, \mathbf{b}_{j'}^g)$, MOTIF [55], in addition to the edge context representations, utilizes segmentation agnostic feature representation $\mathbf{z}_{j,j'}^g$ corresponding to the union of boxes $(\mathbf{b}_j^g, \mathbf{b}_{j'}^g)$.

To ground relations to pixel-level regions, our proposed approach instead relies on the segmentation aware feature representations $\hat{\mathbf{z}}_{j,j'}^g$, which are computed using a novel Gaussian attention mechanism, as described in Section 2.5

			PredCls						SGCls						SGDet					
			R@K			mR@K			R@K			mR@K			R@K			mR@K		
Model	Detector	Method	@20	@50	@100	@20	@50	@100	@20	@50	@100	@20	@50	@100	@20	@50	@100	@20	@50	@100
MOTIF†	VGG [41]	Baseline	**58.2**	**64.9**	**66.8**	13.7	17.5	18.9	**32.0**	**35.2**	**36.0**	7.5	9.2	9.8	21.1	26.9	**30.0**	5.2	6.8	7.9
		Seg-Ground	57.0	64.0	65.9	**14.6**	**18.7**	**20.3**	30.6	34.1	35.0	**7.9**	**9.8**	**10.5**	**22.2**	**27.5**	29.9	**5.6**	**7.3**	**8.1**
	ResNeXt-101-FPN [34, 49]	Baseline	**57.8**	**64.8**	**66.6**	14.1	18.0	19.5	**35.0**	38.5	39.4	8.0	9.9	10.6	23.8	30.2	33.6	5.8	7.7	9.0
		Seg-Ground	55.2	62.0	64.0	**14.5**	**18.5**	**20.2**	34.9	**38.5**	**39.6**	**8.9**	**11.2**	**12.1**	**25.0**	**31.2**	**33.7**	**6.4**	**8.3**	**9.2**
VCTree†	VGG [41]	Baseline	**57.9**	**64.9**	**66.8**	14.4	18.4	19.8	32.0	35.7	36.7	8.1	9.9	10.7	19.4	24.3	26.5	4.4	5.7	6.4
		Seg-Ground	56.7	63.6	65.6	**14.8**	**18.9**	**20.5**	**33.2**	**36.9**	**37.9**	**8.7**	**10.8**	**11.6**	**21.6**	**26.9**	**29.1**	**5.3**	**7.0**	**7.8**
	ResNeXt-101-FPN [34, 49]	Baseline	**58.1**	**64.8**	**66.7**	13.7	17.4	19.0	35.4	38.9	39.8	8.1	9.9	10.6	22.9	29.1	32.3	5.3	6.9	7.9
		Seg-Ground	55.3	62.2	64.3	**15.0**	**19.2**	**21.1**	**37.5**	**41.2**	**42.2**	**9.3**	**11.6**	**12.3**	**24.7**	**30.8**	**33.5**	**6.3**	**8.1**	**9.0**

Table A1. **Quantitative Results.** Table shows the **R@K** and **mR@K** comparison the baseline model and models augmented with the proposed segmentation grounding.

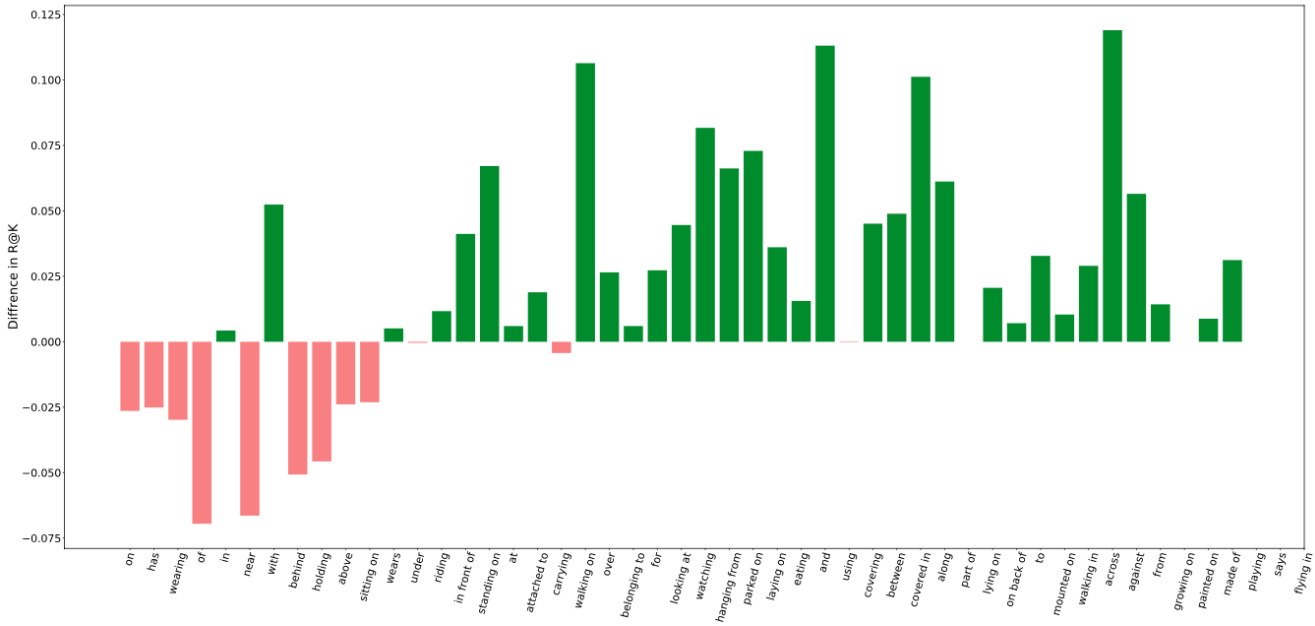

Figure A1. **Relation Wise Recall.** Plot shows the relative difference in the recall of individual relations for a VCTree model trained with the baseline method and one trained with the proposed methodology. A green(red) bar denote improvement(decrease) in performance when using the segmentation grounding method. The x-axis is sorted by the sampling fraction of the relations in Visual Genome. We node that our model performs better on relations with less annotation but suffer on generic annotations such as `on`, `has` *etc*. This explains why our model has superior mR@K performance in PredCls but experience a drop in R@K.

of the main paper. Formally, the probability that the edge will have a relation label $r_{j\to j'}$ is computed as follows,

$$\begin{aligned} \hat{\mathbf{g}}^g_{j,j'} &= \left(\mathbf{W}_h \hat{\mathbf{d}}^g_j\right) \odot \left(\mathbf{W}_t \hat{\mathbf{d}}^g_{j'}\right) \odot \hat{\mathbf{z}}^g_{j,j'} \\ \Pr\left(r_{j\to j'}\right) &= \text{softmax}\left(\mathbf{W}_r \hat{\mathbf{g}}^g_{j,j'} \mathbf{w}_{o_j,o_{j'}}\right) \end{aligned} \tag{A5}$$

where $\mathbf{W}_h$, $\mathbf{W}_t$, and $\mathbf{W}_r$ are learned parameters, and $\mathbf{w}_{o_j,o_{j'}}$ is a bias vector specific to the labels o_j and $o_{j'}$.

A.2. Integration with VCTree

VCTree [45], for a given image, dynamically generates a binary tree, where each node corresponds to an object within the image. The construction of this tree involves running the Prim's algorithm for maximum spanning tree over a symmetric adjacency matrix $\mathbf{S}$. For a particular pair of nodes (j, j'), each element of $\mathbf{S}$ is defined using as the product of the object correlation and the pairwise task-dependency scores. Please see Section 3.1 in [45] for a detailed explanation on how $\mathbf{S}$ is computed. Once this VCTree is generated, a bidirectional TreeLSTM [43] is then used

Model	Method	PredCls			SGCls			SGDet		
		@20	@50	@100	@20	@50	@100	@20	@50	@100
MOTIF†	Baseline	$13.9^{-0.2}$	$17.8^{-0.2}$	$19.3^{-0.1}$	$7.9^{-0.1}$	$9.5^{-0.4}$	$10.3^{-0.3}$	$5.7^{-0.1}$	$7.1^{-0.6}$	$8.2^{-0.6}$
	Ours	$14.6^{+0.1}$	$18.5^{+0.0}$	$20.1^{-0.1}$	$8.7^{-0.2}$	$11.0^{-0.2}$	$11.8^{-0.3}$	$6.3^{-0.1}$	$8.2^{-0.1}$	$9.1^{-0.1}$
VCTree†	Baseline	$13.7^{+0.0}$	$17.3^{-0.1}$	$18.9^{-0.1}$	$8.0^{-0.1}$	$9.9^{+0.0}$	$10.5^{-0.1}$	$5.1^{-0.2}$	$6.9^{+0.0}$	$7.8^{-0.1}$
	Ours	$15.0^{+0.0}$	$19.3^{+0.1}$	$21.3^{+0.3}$	$9.1^{-0.3}$	$11.5^{-0.1}$	$12.2^{-0.1}$	$6.2^{-0.1}$	$7.9^{-0.2}$	$8.8^{-0.2}$

Table A2. **Performance on Visual Genome Subset.** Table shows the **mr@K** values computed on a Visual Genome test subset that does not contain images from the MS-COCO training set. The results assume the ResNeXt-101-FPN backone [34, 49] trained models described in Table 1 of the main paper. The relative deviation from the corresponding values in Table 1 of the main paper is shown using a red / green superscript. A superscript $+x$ implies a positive relative deviation from the values reported in Table 1 by x. Similarly, a superscript $-x$ a negative relative deviation by x.

Method	PredCls			SGCls			SGDet		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
No Refine	$31.8^{+0.3}$	$64.3^{+0.5}$	$28.3^{+0.2}$	$32.4^{-0.1}$	$58.6^{-0.3}$	$31.8^{+0.0}$	$22.9^{-0.3}$	$44.1^{-0.6}$	$21.3^{-0.3}$
MOTIF† + Refine	$42.8^{+0.4}$	$78.4^{+0.3}$	$41.3^{+0.4}$	$37.5^{+0.0}$	$63.3^{-0.2}$	$38.9^{+0.1}$	$24.4^{-0.3}$	$45.3^{-0.5}$	$23.6^{-0.3}$
VCTree† + Refine	$42.3^{+0.4}$	$78.0^{+0.4}$	$40.6^{+0.3}$	$37.3^{-0.1}$	$63.0^{-0.4}$	$38.7^{+0.1}$	$24.6^{-0.3}$	$45.6^{-0.5}$	$23.8^{-0.3}$

Table A3. **Performance on MS-COCO Subset.** Table shows the standard **AP** values computed on a MS-COCO val subset that does not contain images from the Visual Genome training set. The results assume the VGG-16 backbone [41] trained models described in Table 1 of the main paper. The relative deviation from the corresponding values in Table 2 of the main paper is shown using a red / green superscript. A superscript $+x$ implies a positive relative deviation from the values reported in Table 2 by x. Similarly, a superscript $-x$ a negative relative deviation by x.

to capture global context, leading to improved scene graph predictions.

Object Network. VCTree [45] utilizes a bidirectional TreeLSTM to obtain contextualized representations for object label refinement. Specifically, for a set of bounding boxes $\mathbf{B}^g$, the set of object context representations $\mathbf{C}^g = \{\mathbf{c}_1^g, \ldots, \mathbf{c}_n^g\}$ is computed as,

$$\mathbf{C}^g = \text{BiTreeLSTM}\left(\left[\mathbf{z}_j^g; \mathbf{W}_{l1}\mathbf{l}_j^g\right]_{j=1,\ldots,n}\right) \tag{A6}$$

where $\mathbf{W}_{l1}$ is a learned parameter matrix. The bidirectional TreeLSTM involves parsing the tree in a top-down and bottom-up direction using two TreeLSTMs [43]. Therefore, each $\mathbf{c}_j^g \in \mathbf{C}^g$ is computed as the concatenation of the hidden states obtained from these two TreeLSTMs.

Similar to Section A.1, our proposed approach, utilizes the per class masks $\mathbf{m}_j^g$ to compute a segmentation aware representation $\hat{\mathbf{z}}_j^g$ for each bounding box $\mathbf{b}_j^g$. These are obtained via a learned network $f_\mathbf{N}$ as described in Section 2.4 of the main paper. The segmentation aware context object representations $\hat{\mathbf{C}}^g = \{\hat{\mathbf{c}}_1^g, \ldots, \hat{\mathbf{c}}_n^g\}$ are then computed identically to Equation A6,

$$\hat{\mathbf{C}}^g = \text{BiTreeLSTM}\left(\left[\hat{\mathbf{z}}_j^g; \mathbf{W}_{l1}\mathbf{l}_j^g\right]_{j=1,\ldots,n}\right) \tag{A7}$$

A decoder TreeLSTM [43] utilizes these $\mathbf{C}^g$, and processes the tree in a top-down fashion to sequentially generate labels for each bounding box as follows,

$$\begin{aligned} \hat{\mathbf{h}}_j^g &= \text{TreeLSTM}_j\left(\left[\hat{\mathbf{c}}_j^g; \hat{\mathbf{o}}_p^g\right]\right) \\ \hat{\mathbf{o}}_j^g &= \text{argmax}\left(\mathbf{W}_o\hat{\mathbf{h}}_j^g\right) \end{aligned} \tag{A8}$$

where $\hat{\mathbf{o}}_j^g$ are one-hot class labels for object $\mathbf{b}_j^g$, $\mathbf{W}_o$ is a learned parameter matrix, and $\mathbf{c}_p^g$ is the contextual embedding corresponding to the j's parent in the tree.

Relation Network. For relation prediction, VCTree [45] first generates a contextualized representation for each object. The edge context representation $\mathbf{D}^g = \{\mathbf{d}_1^g, \ldots, \mathbf{d}_n^g\}$ is computed using a bidirectional TreeLSTM, using the segmentation agnostic object context representations $\mathbf{C}^g$.

Our proposed approach instead uses segmentation aware context representations $\hat{\mathbf{C}}^g$ as input to the relation network.

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
MOTIF†	VGG-16 [41]	Baseline	13.7	17.5	18.9	7.5	9.2	9.8	5.2	6.8	7.9
		Seg-Grounded	**14.4**$^{-0.2}$	**18.5**$^{-0.2}$	**20.1**$^{-0.2}$	**7.9**$^{+0.0}$	**9.9**$^{+0.1}$	**10.7**$^{+0.2}$	**5.5**$^{-0.1}$	**7.2**$^{-0.1}$	**8.1**$^{+0.0}$
	ResNeXt-101-FPN [34, 49]	Baseline	14.1	18.0	19.4	8.0	9.9	10.6	5.8	7.7	9.0
		Seg-Grounded	**15.0**$^{+0.4}$	**19.0**$^{+0.3}$	**20.6**$^{+0.3}$	**9.0**$^{+0.1}$	**11.1**$^{-0.1}$	**12.0**$^{-0.1}$	**6.4**$^{+0.0}$	**8.3**$^{+0.0}$	**9.3**$^{+0.1}$
VCTree†	VGG-16 [41]	Baseline	14.4	18.4	19.8	8.1	9.9	10.7	4.4	5.7	6.4
		Seg-Grounded	**14.9**$^{+0.1}$	19.2$^{+0.3}$	**20.9**$^{+0.4}$	**8.7**$^{+0.0}$	**10.7**$^{-0.1}$	**11.5**$^{-0.1}$	**5.5**$^{+0.2}$	**7.1**$^{+0.1}$	**7.8**$^{+0.0}$
	ResNeXt-101-FPN [34, 49]	Baseline	13.7	17.4	19.0	8.1	9.9	10.6	5.3	6.9	7.9
		Seg-Grounded	**15.1**$^{+0.1}$	**19.1**$^{-0.1}$	**20.9**$^{-0.2}$	**9.3**$^{+0.0}$	**11.4**$^{-0.2}$	**12.2**$^{-0.1}$	**6.2**$^{-0.1}$	**8.1**$^{+0.0}$	**9.0**$^{+0.0}$

Table A4. **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], and trained using the MS-COCO subset described in Section B. The relative deviation from the corresponding values in Table 1 of the main paper is shown using a red / green superscript. A superscript $+x$ implies a positive relative deviation from the values reported in Table 1 by x. Similarly, a superscript $-x$ a negative relative deviation by x. † denotes our re-implementation of the methods.

Detector	Method	Predicate Classification			Scene Graph Classification			Scene Graph Generation		
		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
VGG-16 [41]	No Refine	31.8$^{+0.3}$	64.3$^{+0.5}$	28.3$^{+0.2}$	32.4$^{-0.1}$	58.6$^{-0.3}$	31.8$^{+0.0}$	22.9$^{-0.3}$	44.1$^{-0.6}$	21.3$^{-0.3}$
	MOTIF† + Refine	**42.7**$^{+0.3}$	**78.3**$^{+0.2}$	**41.2**$^{+0.3}$	**37.4**$^{-0.1}$	63.0$^{-0.5}$	**39.0**$^{+0.2}$	24.5$^{-0.2}$	45.4$^{-0.4}$	23.6$^{-0.3}$
	VCTree† + Refine	42.5$^{+0.6}$	78.0$^{+0.4}$	41.1$^{+0.8}$	37.2$^{-0.2}$	**63.1**$^{-0.3}$	38.5$^{-0.1}$	**24.5**$^{-0.4}$	**45.5**$^{-0.6}$	**23.7**$^{-0.4}$
ResNeXt-101-FPN [34, 49]	No Refine	55.0$^{+0.2}$	88.1$^{+0.5}$	58.7$^{+0.4}$	51.4$^{-0.2}$	76.5$^{-0.2}$	56.4$^{-0.5}$	38.7$^{-0.5}$	**60.6**$^{-0.6}$	41.9$^{-0.5}$
	MOTIF† + Refine	59.3$^{+0.0}$	**90.9**$^{+0.3}$	64.5$^{-0.2}$	54.0$^{-0.6}$	77.6$^{-0.6}$	60.1$^{-1.0}$	**38.7**$^{-0.5}$	60.5$^{-0.7}$	**41.9**$^{-0.5}$
	VCTree† + Refine	**59.3**$^{+0.3}$	90.8$^{+0.4}$	**64.7**$^{+0.5}$	**54.1**$^{-0.2}$	**77.7**$^{-0.2}$	**60.1**$^{-0.3}$	38.7$^{-0.5}$	60.5$^{-0.7}$	41.9$^{-0.5}$

Table A5. **Segmentation Refinement on MSCOCO.** Standard COCO precision metrics are reported across three tasks and two detector backbones. Task formulation is identical to Table A4. 'No Refine' is the baseline where the segmentation masks are obtained from the pre-trained detector. Evaluation is performed on the MS-COCO subset that does not contain images from the Visual Genome training set. The relative deviation from the corresponding values in Table 2 of the main paper is shown using a red / green superscript. A superscript $+x$ implies a positive relative deviation from the values reported in Table 2 by x. Similarly, a superscript $-x$ a negative relative deviation by x.

Specifically, the segmentation aware edge context representations $\mathbf{D}^g$ are computed as follows,

$$\hat{\mathbf{D}}^g = \text{BiTreeLSTM}\left(\left[\hat{\mathbf{c}}_j^g\right]_{j=1,\ldots,n}\right) \tag{A9}$$

To predict a relation between a pair of objects $(\mathbf{b}_j^g, \mathbf{b}_{j'}^g)$, VCTree [45] generates three pairwise features, which are computed using segmentation agnostic edge features $(\mathbf{d}_j^g, \mathbf{d}_{j'}^g)$ and the union box features $\mathbf{z}_{j,j'}^g$.

To ground relations to pixel-level regions, our proposed approach instead utilizes segmentation aware feature representations $\hat{\mathbf{d}}_j^g, \hat{\mathbf{d}}_{j'}^g)$ and union box features $\hat{\mathbf{z}}_{j,j'}^g$. The union box features $\hat{\mathbf{z}}_{j,j'}^g$ are computed using a novel Gaussian attention mechanism, as described in Section 2.5. Specifically, the probability that the edge will have a relation label $r_{j\to j'}$ is computed as follows,

$$\begin{aligned} \mathbf{g}_{j,j'}^g &= \hat{\mathbf{d}}_{j,j'} \odot \hat{\mathbf{z}}_{j,j'}^g \odot \mathbf{b}_{j,j'}^g \\ \Pr\left(r_{j\to j'}\right) &= \text{softmax}\left(\mathbf{W}_r \hat{\mathbf{g}}_{j,j'}^g \mathbf{w}_{o_j,o_{j'}}\right) \end{aligned} \tag{A10}$$

where $\mathbf{W}_r$ is a learned parameter matrix, and $\mathbf{w}_{o_j,o_{j'}}$ is a bias vector specific to the labels o_j and $o_{j'}$. Additionally,

$$\begin{aligned} \hat{\mathbf{d}}_{j,j'}^g &= f_{\mathbf{D}}\left(\left[\hat{\mathbf{d}}_j^g; \hat{\mathbf{d}}_{j'}^g\right]\right) \\ \mathbf{b}_{j,j'}^g &= f_{\mathbf{B}}\left(\left[\mathbf{b}_j^g; \mathbf{b}_{j'}^g; \mathbf{b}_j^g \cup \mathbf{b}_{j'}^g; \mathbf{b}_j^g \cap \mathbf{b}_{j'}^g\right]\right) \end{aligned} \tag{A11}$$

where $f_{\mathbf{D}}$ and $f_{\mathbf{B}}$ are learned networks, $\mathbf{b}_j^g \cup \mathbf{b}_{j'}^g$ corresponds to the union box, and $\mathbf{b}_j^g \cap \mathbf{b}_{j'}^g$ corresponds to the intersection box.

A.3. Segmentation Refinement

Our proposed approach uses multi-task learning to simultaneously improve performance on both scene graph and segmentation generation. To this end, as described in Section 2.6 of the main paper, our proposed incorporates an additional segmentation refinement head $f_{\mathbf{M}'}$ to improve on the inferred masks. As segmentation annotations are unavailable in $\mathcal{D}^g$, we use the dataset $\mathcal{D}^m$ to train $f_{\mathbf{M}'}$. Specifically, as described in Section 2.6, the refined masks $\hat{\mathbf{m}}_j^m$ for a particular bounding box $\mathbf{b}_j^m \in \mathbf{B}^b$ is computed as,

$$\hat{\mathbf{m}}_j^m = \mathbf{m}_j^m + f_{\mathbf{M}'}\left(\mathbf{z}_j^{o,m}\right) \tag{A12}$$

Class	AP	Class	AP	Class	AP	Class	AP	Class	AP
Airplane	$35.5^{+1.3}$	Apple	$11.4^{+1.1}$	Backpack	$7.6^{+0.7}$	Banana	$10.5^{+1.2}$	Baseball bat	$12.0^{-0.1}$
Baseball glove	$27.2^{-0.0}$	Bear	$56.6^{+0.9}$	Bed	$22.0^{+1.6}$	Bench	$10.1^{+0.7}$	Bicycle	$11.8^{+1.5}$
Bird	$17.4^{+1.0}$	Boat	$12.5^{+1.2}$	Book	$2.6^{+0.2}$	Bottle	$24.1^{+1.6}$	Bowl	$28.4^{+2.2}$
Broccoli	$15.5^{+1.0}$	Bus	$53.0^{+3.3}$	Cake	$21.7^{+2.4}$	Car	$26.8^{+4.6}$	Carrot	$9.0^{+1.3}$
Cat	$55.3^{+1.6}$	Cell phone	$21.8^{+0.8}$	Chair	$8.4^{+1.6}$	Clock	$42.4^{+0.1}$	Couch	$23.5^{+1.8}$
Cow	$29.3^{+5.6}$	Cup	$31.1^{+2.3}$	Dining table	$10.9^{+0.9}$	Dog	$46.9^{+1.8}$	Donut	$25.4^{+5.0}$
Elephant	$37.5^{+6.1}$	Fire hydrant	$51.7^{+1.0}$	Fork	$3.8^{+0.2}$	Frisbee	$46.4^{+0.5}$	Giraffe	$35.0^{+4.1}$
Hair drier	$0.0^{+0.0}$	Handbag	$5.8^{+0.1}$	Horse	$30.1^{+1.7}$	Hot dog	$11.0^{+0.3}$	Keyboard	$35.1^{+1.8}$
Kite	$14.3^{+0.4}$	Knife	$1.6^{+0.2}$	Laptop	$43.5^{+2.8}$	Microwave	$42.6^{+0.3}$	Motorcycle	$22.5^{+2.8}$
Mouse	$44.4^{+0.4}$	Orange	$20.2^{+2.7}$	Oven	$19.8^{+1.1}$	Parking meter	$39.6^{+2.0}$	Person	$33.4^{+5.8}$
Pizza	$40.0^{+1.3}$	Potted plant	$14.8^{+0.7}$	Refrigerator	$35.8^{+2.0}$	Remote	$12.6^{+0.4}$	Sandwich	$21.3^{+2.1}$
Scissors	$7.5^{+2.0}$	Sheep	$26.0^{+8.3}$	Sink	$24.7^{+1.5}$	Skateboard	$19.5^{+1.8}$	Skis	$0.3^{-0.0}$
Snowboard	$9.3^{+0.3}$	Spoon	$1.8^{+0.1}$	Sports ball	$28.2^{+0.2}$	Stop sign	$52.9^{+0.9}$	Suitcase	$17.3^{+3.2}$
Surfboard	$17.7^{+0.7}$	Teddy bear	$26.2^{+5.2}$	Tennis racket	$40.2^{+0.3}$	Tie	$15.8^{+0.4}$	Toaster	$25.1^{+0.1}$
Toilet	$47.3^{+2.5}$	Toothbrush	$2.8^{-0.1}$	Traffic light	$16.0^{+0.3}$	Train	$53.2^{+1.3}$	Truck	$24.9^{+2.0}$
Tv	$45.8^{+1.8}$	Umbrella	$30.2^{+3.4}$	Vase	$23.6^{+1.9}$	Wine glass	$18.4^{+1.6}$	Zebra	$42.3^{+7.7}$

Table A6. **Per Class Segmentation Improvement.** Per class AP metrics are reported for our approach augmented to VGG-16 [41] based VCTree [45] method. The table mentions the performance on the Scene Graph Generation task, and highlights the relative per-class improvement from the baseline 'No Refine' approach described in Table 2. A superscript $+x$ implies that our proposed approach is better than the baseline on that class by x AP. A superscript $-x$ implies that our proposed approach is worse than the baseline on that class by x AP.

where $\mathbf{z}_j^{o,m} = \hat{\mathbf{c}}_j^m$. Note that, for an image $\mathbf{x}^m \in \mathcal{D}^m$, $\hat{\mathbf{c}}_j^m$ is obtained by first extracting the proposal boxes $\mathbf{B}^m$ using the object detector, and then passing the required inputs in to the object network. This does not induce an additional memory constraints as the parameters are shared between both datasets $\mathcal{D}^g$ and $\mathcal{D}^m$.

B. Information Leakage Analysis

As described in Section 3 of the main paper, we use MS-COCO [29] as the auxiliary dataset $\mathcal{D}^m$ in our experiments. As Visual Genome [24] and MS-COCO [29] have images in common, there is a possibility of information across the two datasets. In this section we empirically show that such leakage is non-existent and provide a detailed analysis.

To highlight that there is no information leakage from MS-COCO to Visual Genome, we evaluate the models desribed in Section 4 of the main paper on a Visual Genome test subset that does not contain images from the MS-COCO training set. Assuming the ResNeXt-101-FPN detector [34, 49], Table A2 shows the mean recall (mR) on this reduced set, and also mentions the relative deviation from the corresponding mR values in Table 1 of the main paper.

Similarly, to show there is no leakage from Visual Genome to MS-COCO, we evaluate trained models mentioned in Table 2 of the main paper on a MS-COCO val subset that does not contain images from the Visual Genome training set. To show that backbone choices do not induce leakage either, results shown in Table A3 assume a VGG-16 [41] detector.

Finally, to further highlight that our joint training approach does not introduce any information leakage, we re-train the models with a modified auxiliary MS-COCO dataset to ensure there is no overlap between images. Specifically, we remove any images contained in the Visual Genome test set from the MS-COCO training set. Similarly, any images contained in the Visual Genome train set is removed from the MS-COCO validation set. As a consequence, the modified MS-COCO train set contains $104,723$ images and the modified MS-COCO val set contains $3,742$ images. Table A4 reports the mean recall values comparing the baseline and our proposed method trained on this modified dataset, and also highlights the relative deviation from the values reported in Table 1 of the main paper. Similarly, Table A5 reports the standard COCO evaluation metrics and relative deviation from the numbers reported in Table 2.

In all the above experiments, the variations in the performance of our proposed model is similar to that of the baselines on the modified evaluation and training subsets. Additionally, the variations from the experiments done without removing image overlap is minimal, confirming that the leakage from one dataset to another is non-existent.

C. Additional Results

We report the complete results (Regular Recall and mean Recall) of the baseline model and the proposed segmenta-

tion grounding framework in Table A1. For the SGCls and SGDet mode we observe an almost consistent improvement in the both the R@K and mR@K for both models across both detectors. This improvement in performance can be attributed to the improvement in the representations learned from exploiting segmentation predictions and improvement in the overall performance of the detector. For the PredCls task, we note that the use of our proposed framework leads to an improvement in the mR@K but a drop in R@K. This behaviour is due to the long-tail distribution of annotations in the Visual Genome dataset [44]. The use of our method leads to models prediction more granular predicates such a `standing on`, `parked on` as opposed to a generic relation `on`. This effect can be seen observed in the qualitative results in Figure A2 and relation wise recall in Figure A1.

We additionally report per class segmentation refinement improvements on MSCOCO in Table A6 for our approach augmented to VGG-16 [41] based VCTree [45] method on the Scene Graph Generation task. It can be seen that our proposed joint training significantly improves segmentation performance across all classes. To analyse this further, we also qualitatively visualize segmentation refinement improvements on MSCOCO in Figure A3. It can be seen that our proposed multi-task learning visually improves segmentation mask quality.

Figure A2. **Qualitative Results.** Visualizations of scene graphs generate using a VCTree baseline (in purple) and a VCTree model augmented with proposed segmentation grounding framework (in green). The zero-shot triplets are indicated in yellow. We omit some nodes and edges in the visualization where the baseline and the proposed model predicts the same relation for clarity.

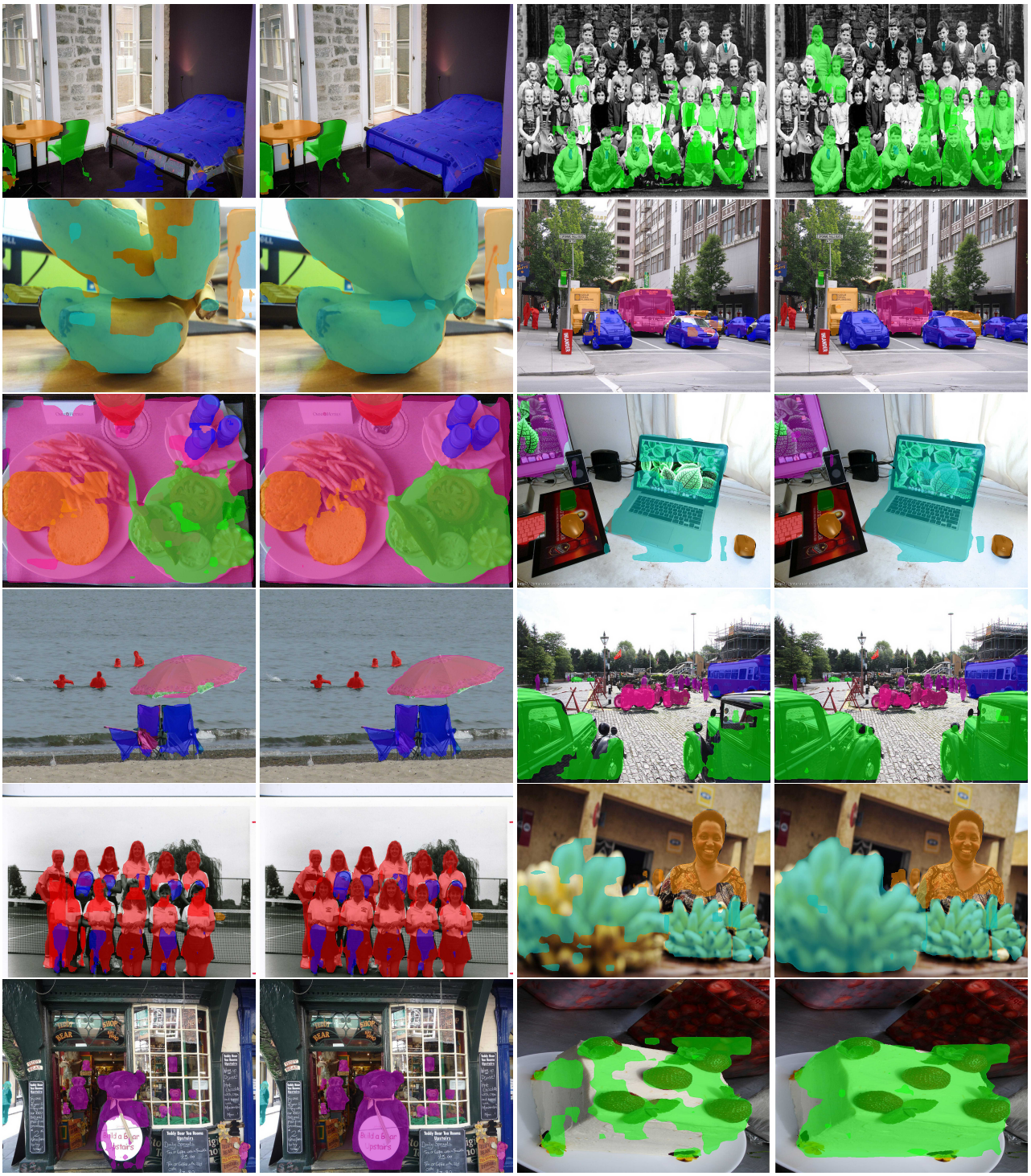

Figure A3. **Segmentation Refinement Qualitative Results.** Visualizations show the improvements in segmentation masks by using our proposed segmentation refinement on MSCOCO. For each pair of images, the one on the left shows the masks obtained by using segmentation head f_{M} (Section 2.3), and the one on the right shows the masks obtained after refinement by using $f_{\mathrm{M}'}$ (Section 2.6). For a particular image, color is indicative of class label.

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.