Auto-Encoding Scene Graphs with Image Reconstruction

Kevin Tan
Stanford University
kevintan@cs.stanford.edu
CS236 Fall 2019 Final Report

Abstract

Understanding a visual scene requires not only recognizing individual objects, but also inferring the relationships and interactions between them. Modern deep generative models have demonstrated remarkable ability in producing high-quality samples, yet they fail to capture the relationships between objects in a visual scene. In contrast, scene graphs explicitly model objects and their relationships by a visually-grounded graphical structure of an image. In this work, we propose an encoder-decoder method for scene graph generation analogous to autoencoder architectures. We additionally introduce image reconstruction as a supervisory signal to regularize the scene graph generator to address noisy and biased annotations. We validate our approach on the large-scale Visual Genome benchmark dataset, demonstrating that our approach can learn to reconstruct images using predicted scene graphs with similar qualitative results to an oracle relying on ground truth.

1 Introduction

To truly understanding the visual world, our models should be not only recognize images but also generate them. To this end, modern deep generative models have demonstrated remarkable ability in synthesizing high-quality, realistic-looking samples. In the lens of probability, generative models are able to learn an approximation of an underlying distribution p_{data} given a finite set of observed dataset $\mathcal{D}$. However, while these methods can give stunning results on limited domains, they still struggle in more complicated synthesis settings, e.g. on the task of *text-to-image synthesis* as demonstrated by Johnson et al. [9].

To address this issue, one idea is to endow generative models with explicit representations that capture visual context in a *compositional* manner, in order to (i) generalize effectively to long-tail instances from the data distribution, and (ii) enjoy better model interpretability compared to blackbox deep networks. *Scene graphs* [10] provide a way to represent scenes in this way – as directed graphs, where the nodes are objects and edges are relationships between objects. In the past, scene graphs have been used for image retrieval [10], image captioning [1], or predicting grounded scene graphs from language priors [15]. Most work on scene graphs have used the Visual Genome (VG) [13] benchmark for evaluation which is a large-scale dataset that provides human annotations for scene graphs. However, since VG is hand-crafted by humans, some of the scene graph annotations are noisy or biased.

In this work, we propose to address the problem of scene graph generation with an additional constraint of image reconstruction loss such that the learned scene graph generating procedure is robust to noisy or biased annotations in VG. Similar to autoencoders, our encoder-decoder approach aims to map an input image to a latent representation, i.e. its corresponding scene graph, and subsequently reconstruct the image in an end-to-end fashion. The image generator for reconstruction is conditioned on the scene graph and maps a noise vector to the output image. In this way, our approach leverages two sources of supervision – both at the scene graph level and at the image level.

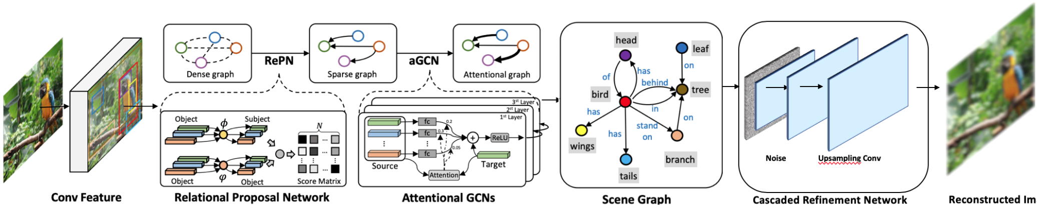

Figure 1: **Overview of our approach.** Given an input image, we first extract region proposals with an object detector i.e. Faster R-CNN. The relational proposal network (RPN) prunes the connections in the `<subject, relationship, object>` by computing a learned relatedness score matrix, ranking them in descending order, and choosing the top K pairs. Graph convolution is applied to aggregate contextual information from neighboring nodes to produce a scene graph. To reconstruct an image, we pass Gaussian noise conditioned on the scene layout into a Cascaded Refinement Network (CRN) which consists of a series of upsampling convolutions.

Our contributions are two-fold:

- We propose a method for scene graph generation in an end-to-end trainable fashion via image reconstruction supervision.
- We demonstrate that our method can achieve similar results with predicted scene graphs compared to an oracle (sg2im [9]) relying on ground truth.

2 Related Work

Scene Graph Generation. A number of approaches for scene graph generation have been proposed for the detection of objects and their relationships [3]. Several have noted that explicit reasoning over the quadratic number of pairwise relationships is intractable, and address this by heuristic or random methods. Other works have explored mechanisms for refining final relationship labels by aggregating contextual information, such as proposing sequential and parallel message passing strategies [21]. Others have noted strong regularities in visual scenes with motivate a formal definition of motifs [24]. Most similar to ours is Graph R-CNN [22] which offers efficient relation proposals by leveraging a learned relatedness kernel function to save on memory and computation. Our work is built upon Graph R-CNN but propose to additionally supervise scene graph prediction via an image reconstruction loss.

Graph Convolutional Networks (GCNs). Graph Neural Networks (GNNs) [5] are a family of neural network models that operate on arbitrary graph structures, and have been applied to a wide range of domains. Graph convolutional networks (GCNs) [12] reason over graphs by performing a series of localized operations typically involving only neighboring nodes for each node at each time step. Similar to convolutional neural networks, the structure and edge strengths are chosen a priori. In our work, we use GCNs to integrate contextual information informed by graph structure, with the intuition that neighboring nodes provide crucial contextual information.

Deep Generative Models. Recent methods for synthesizing images include generative adversarial networks (GANs) [6] and variational autoencoders (VAE) [11]. These methods belong to a family of deep generative models that can learn to synthesize images by learning an approximation of the underlying data distribution given a finite training set of images. GANs consist of a generator and a discriminator where the goal of the generator is to produce realistic images so that the discriminator cannot tell the synthesized images apart from the real ones. Our work is built on GAN aimed for the conditional image synthesis task where a synthesized image is produced via a series of upsampling convolution on some input noise conditioned by a scene layout.

Conditional Image Synthesis. There is a large body of work across various forms of input data for the task of conditional image synthesis. For example, class-conditional models learn to synthesize images given a class label [16] [17], and text-conditioned models aim to generate images from text [8]. Most related to our approach, [9] synthesizes images conditioned on an oracle scene graph with subpar qualitative results. Our work is built on this approach by adopting a scene generation module to infer scene graphs from images instead of relying on the ground truth. With an image reconstruction module, our network is able to be trained end-to-end with supervision from both scene graph and image levels.

3 Approach

3.1 Problem Statement

A *scene graph* [9] is a structured representation of an image, where nodes in a scene graph correspond to object bounding boxes with their object categories, and edges correspond to their pairwise relationships between objects. The task of *scene graph generation* is to generate a visually-grounded scene graph that most accurately correlates with an image.

More formally, a *scene graph* can be defined by a 3-tuple set $G = \{B, O, R\}$:

- $B = \{b_1, b_2, ..., b_n\}$ is the region candidate set, with elements $b_i \in \mathbb{R}^4$ denoting the bounding box of the i^{th} region
- $O = \{o_1, o_2, ..., o_n\}$ is the object set, with element $o_i \in \mathbb{N}$ denoting the corresponding class label of region b_i
- $R = \{r_1, r_2, ..., r_m\}$ of pairwise relationships between those objects, where r_k denotes a triplet of a start node $(b_i, o_i) \in B \times O$, an end node $(b_j, o_j) \in B \times O$, and a relationship label $x_{i \to j} \in \mathcal{R}$, where $\mathcal{R}$ is the set of all possible predicate types.

Given an image I, the goal is to decompose the probability distribution of the scene graph $P(G \mid I)$ into three components, as demonstrated previously by [24]:

$$Pr(G \mid I) = Pr(B \mid I)Pr(O \mid B, I)Pr(R \mid O, B, I) \tag{1}$$

The bounding box component $Pr(B \mid I)$ generates the set of candidate regions for the key objects in the input image, given by the output of an off-the-shelf Faster RCNN detector [19]. The object component $Pr(O \mid B, I)$ uses the detected regions to predict the class labels of each region. The relationship component $Pr(R \mid O, B, I)$ is conditioned on the predicted labels, and infers the pairwise relationships to generate the whole scene graph G.

3.2 Model Components

Bounding Box Detector: We follow previous work [22] to obtain the bounding box localizations. Given an input image I, we utilize a Faster RCNN to generate the region set $B = \{b_1, b_2, ..., b_n\}$ of size $|B| = n$. Each region is associated with an additional feature vector f_i using the ROI pooling layer [4], which are subsequently fed into the relation proposal network.

Relation Proposal Network: For any two different objects $(o_i, o_j) \in O$, there are two possible relationships in opposite directions. Therefore, for N object proposals, there are $N \times (N - 1)$ potential relations. Storing more relationships results in a larger scene graph with more expressivity, but significantly increases the computational cost in the forward pass of the region proposal network. To overcome this, we follow a similar approach to Graph R-CNN [] to take advantage of a learned relatedness kernel function to reduce memory and computation costs [22].

Specifically, we consider the following kernel function:

$$f(o_i, o_j) = \langle \Phi(o_i), \Psi(o_j) \rangle, i \neq j \tag{2}$$

where $\Phi(\cdot)$ and $\Psi(\cdot)$ are projection functions for subjects and objects, respectively. We construct a score matrix $S = \{s_{ij}\}^{n \times n}$ for all object pairs that computes a measure of pairwise relatedness. To do so, we instantiate $\Phi(\cdot)$ and $\Psi(\cdot)$ as two multilayer perceptrons (MLPs) with identical architectures but different parameters. We apply a sigmoid as the final layer to output scores between 0 and 1. After obtaining the score matrix, we sort the scores in descending order and select the top K pairs.

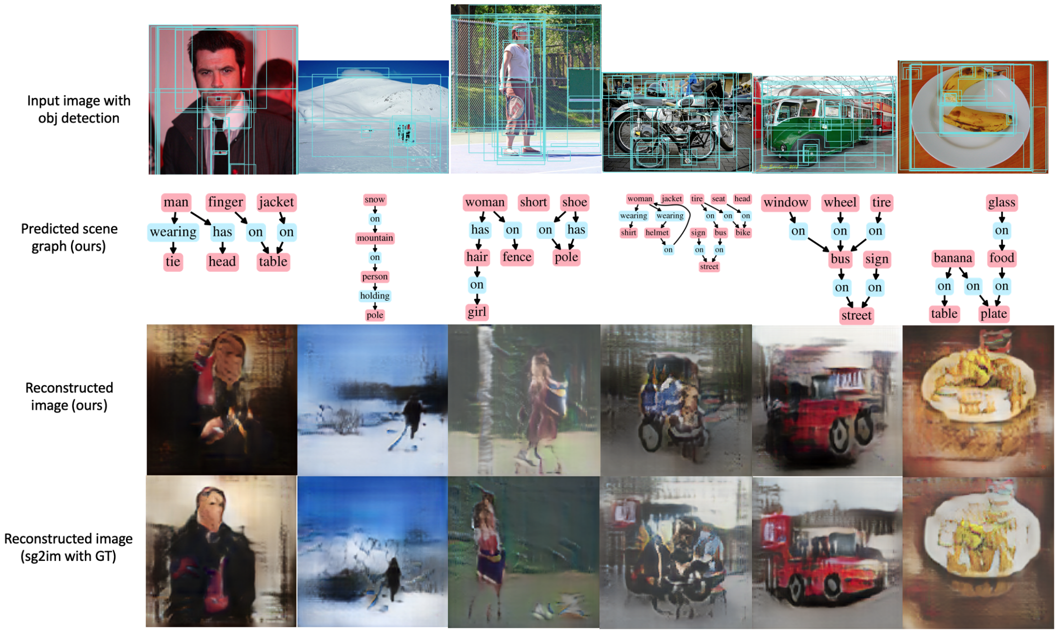

Figure 2: Examples of generated images from test set of Visual Genome. For each example we show the original input image with object region proposals, the predicted scene graph, the reconstructed image conditioned the predicted scene graph, and the reconstructed image from an oracle (sg2im) with access to ground truth. Our approach can achieve similar qualitative results without access to the annotated scene graph during image generation.

Graph Convolution Network: We utilize graph convolution networks (GCN) to aggregate contextual spatial information over the output of the relation proposal network. The purpose of the GCN is to transform the original node embeddings into a new set of context-aware embeddings. Using the contextual embeddings, we can predict the label based on the feature vectors of two object nodes given their corresponding subgraph contextual feature map. The prediction procedure is formulated as follows [23]:

$$R_{i,j} = softmax(f_{rel}([o_i \otimes s_k; o_j \otimes s_k; s_k)])) \tag{3}$$

$$O_i = softmax(f_{node}(o_i)) \tag{4}$$

where f_{rel} and f_{node} denote the relation and node embedding mappings respectively, $\otimes$ denotes the graph convolution operation, and ; denotes concatenation.

Image Generator: To generate an output image from the scene graph, we first need to convert the object embeddings into a spatial scene layout. For each object embedding $o_i \in \mathbb{R}^D$, we expand the embedding vector to size $D \times 8 \times 8$ and wrap it to position of the bounding box via bilinear interpolation to give an object layout o_i^{layout} of size $D \times H \times W$, and sum over all i to obtain the scene layout $S^{layout} = \sum_i o_i^{layout}$.

Given the scene layout, the purpose of the image generator G is to then synthesize an image that respects the object positions and relations. We adopt a Cascaded Refinement Network [2] which consists of a series of convolutional refinement modules to generate the image. Each convolutional refinement module doubles the spatial resolution of the image in a coarse-to-fine manner. Each module takes two inputs: (i) the hidden feature output of the previous module (where the first module takes as input some Gaussian noise), and (ii) the scene layout S^{layout} which is downsampled to fit the input size of the particular module.

Specifically, the two inputs are concatenated via the channel dimension and passed onward to a pair of 3×3 convolution layers. The outputs at each module are upsampled via nearest-neighbor interpolation before being passed as input to the next module. The output of the last module is passed to two final convolution layers to produce the output image.

Discriminator: We adopt a conditional GAN loss [18] to supervise the quality of the reconstructed image. In particular, we train the discriminator $D(\cdot)$ and generator $G(\cdot)$ by alternatively optimizing the following objectives:

$$\mathcal{L}_D = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}}[\log D(\boldsymbol{x})] \tag{5}$$

$$\mathcal{L}_G = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{fake}}}[\log(1 - D(\boldsymbol{x})] + \lambda \mathcal{L}_{\text{pixel}} \tag{6}$$

where $\boldsymbol{x} \sim p_{\text{fake}}$ are outputs from the generator, λ is the tuning parameter, and $\mathcal{L}_{\text{pixel}}$ is given by the l_1 distance between the real image x and a corresponding fake image $\hat{x}$ as $||x - \hat{x}||_1$.

3.3 Training Procedure:

During training we optimize for two levels of supervision: scene graph level and image level.

The scene graph level loss is given by:

$$\mathcal{L}_{sg} = \lambda_{rel}\mathcal{L}_{rel} + \lambda_{obj}\mathcal{L}_{obj} + \lambda_{bb}\mathcal{L}_{bb} \tag{7}$$

where $\mathcal{L}_{rel}$ denotes the relation classification loss, $\mathcal{L}_{obj}$ denotes the object classification loss, and $\mathcal{L}_{bb}$ denotes the bounding box regression loss. The relation classification loss is defined as the cross entropy loss with softmax over the total vocabulary of candidate relations. Similarly, the object classification loss is over all the possible object categories. The bounding box regression loss is the smooth l_1 loss.

The image level loss is given by Eq. (5) and (6). One can view the image reconstruction module as a regularizer to improve the performance of scene graph generation. During backpropagation, we pass the gradient from the losses (5), (6), (7) to update model parameters.

4 Experiments

Dataset. We evaluate our approach on the Visual Genome (VG) benchmark [13]. VG contains 108,077 images, 5.4M region descriptions, 1.7M visual question answers, 3.8M object instances, 2.8M attributes, and 2.3M relationships. On average, VG contains annotations of 38 objects and 22 relationships for each image. It is currently one of the most widely used and challenging benchmarks for evaluating scene graph generation. In the experiments, we follow the procedure of prior work [21], using the most frequent 150 object categories and 50 relationships. As a result, each image has a scene graph of around 11.5 objects and 6.2 relationships. We also follow the same train/test split of 70%/30% respectively.

Implementation Details. We adopt a two-phase training procedure. First, we train the image reconstruction module using the ground truth object annotations in the training set of VG. The output size of the generator is $64 \times 64 \times 3$ as well as the resized real image before inputting to the discriminator. In each mini-batch step, we first update the generator G_i and then update the discriminator D_i.

Second, we jointly train the scene graph generator with the image reconstruction module. We use a pretrained Faster R-CNN [19] with a VGG-16 [20] backbone. The number of object proposals is 256. For each proposal, we perform ROI align [7] pooling to get object and subgraph feature maps. The subgraph regions are pooled to size 5×5 and the hidden dim size is chosen to be 512. We use stochastic gradient descent (SGD) as the optimized with weight decay and dropout to help avoid overfitting, using initial learning rate of 0.01 and decay rate 0.1.

Metrics. We evaluate our model on the following metrics:

- Visual phrase detection (PhrDet): the task of detecting the triplets (o_i, r_{ij}, o_j) denoting the object-relation-object phrases
- Scene Graph Generation (SGGen): task of detecting the objects in the image and recognizing the correct pairwise relation

We follow [15] in reporting Top-K recall (Rec@K) as the performance metric, which measures how many labelled relationships are hit among the top K predictions.

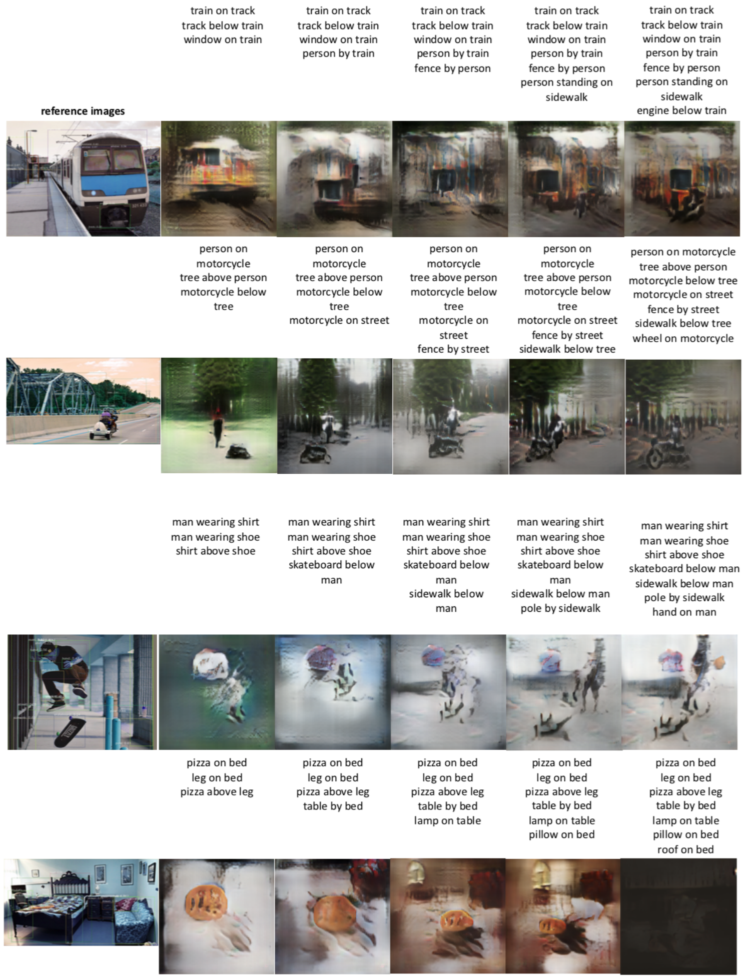

Figure 3: Qualitative results for conditional image synthesis on Visual Genome (VG) demonstrating increasing scene graph complexity. The first column depicts a reference image sampled from VG with the predicted bounding boxes. Each row begins with a simple scene graph and gradually adds more objects and relationships to gain complexity. Here, note that scene graphs in the first column are hand-crafted manually. Generated images demonstrate contextual knowledge and respect object relationships (e.g. pizza on bed).

4.1 Qualitative Results.

Figure 2 shows examples of generated images from a held-out test set on Visual Genome. Each column demonstrates a separate example. We show that our approach can learn to reconstruct the image using the predicted scene graph from the input image. Crucially, our generated images are roughly similar to that of an oracle (i.e. sg2im [9] that relies on a ground truth annotated scene graph. We note that both approaches can roughly recreate the high-level semantics of the scene, such as positions of objects and colors, but fail to faithfully reproduce a more fine-grained image at the pixel level.

Figure 3 demonstrates the effect of increasing the complexity of scene graph during conditional image generation. Here, we use hand-crafted scene graphs from the reference image to produce a series of five reconstructed images, with increasing complexity from left to right. We see that generated images demonstrate contextual knowledge of the scene and respect object relationships as complexity increases. One interesting outlier is the last column of the last row which, upon the additional of the triple "roof on bed", severly darkens the image of the bedroom.

Table 1: Results on PhrDet and SGGen Tasks on Visual Genome

Model	**PhrDet**			**SGGen**		
	Rec@20	Rec@50	Rec@100	Rec@20	Rec@50	Rec@100
IMP [21]	-	15.87	19.45	-	8.23	10.88
MotifNet [24]	-	23.8	27.2	-	23.5	27.6
Graph R-CNN [22]	-	-	-	19.4	25.5	28.5
FactorizableNet [14]	-	26.03	30.77	-	18.32	21.20
ours	**24.1**	**30.5**	**33.1**	**23.8**	**28.2**	**32.6**

4.2 Quantitative Results.

Table 1 presents our quantitative comparisons on our approach against numerous recent models, including Iterative Messaging Passing (IMP) [21], MotifNet [24], Graph R-CNN [22]. We can see that our approach outperforms all the existing methods in the recall on both PhrDet and SGGen tasks. Compared to existing methods, our end-to-end approach utilizes an image reconstruction loss as an additional supervisory signal to update the scene graph generation weights more robustly.

We posit that low-frequency categories of objects in the long tail distribution without many labels add extraneous noise to the training loss. Therefore, with explicitly image level supervision, the model has the option to not only rely on the class label of the training target but also more fine-grained pixel loss.

5 Conclusion

In this work, we have proposed a method for scene graph generation that utilizes image-level supervision allowing for end-to-end training from images to reconstructed images. In this manner, one can interpret the scene graph level representation to be analogous to latent representation in autoencoder models. Using this additional image-level supervision, our experiments show that our approach is able to outperform recent state-of-the-art methods on scene graph generation on Visual Genome. Our work demonstrates a step in the direction of explicitly incorporating common-sense, discrete, and compositional structure as a latent representation in understanding visual scenes.

We hope that future work can explore a number of directions (in no particular order): investigating a decoupling of scene graph generation and image synthesis modules in a more elegant manner; exploring additional reasoning steps leveraging the structured graph representation for downstream tasks such as visual question answering; applications to videos instead of images; leveraging 3D structure as opposed to 2D.

The code is available at `https://github.com/kevinstan/auto_encoding_sg`

Acknowledgements

We thank Kaidi Cao, Jingwei Ji, Ehsan Adeli, and Juan Carlos Niebles for the helpful discussions. We also thank Jianwei Yang and Justin Johnson for the PyTorch implementions of Graph R-CNN and sg2im, respectively.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. *CoRR*, abs/1607.08822, 2016.

[2] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. pages 1520–1529, 10 2017.

[3] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3298–3308, 2017.

[4] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA, 2015. IEEE Computer Society.

[5] C. Goller and A. Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352 vol.1. IEEE, June 1996.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2017. cite arxiv:1703.06870Comment: open source; appendix on more results.

[8] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7986–7994, 2018.

[9] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018.

[10] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. cite arxiv:1312.6114.

[12] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[13] 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. 2016.

[14] 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 *ECCV*, 2018.

[15] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016.

[16] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. cite arxiv:1411.1784.

[17] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *ArXiv*, abs/1802.05637, 2018.

[18] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1060–1069. JMLR.org, 2016.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[21] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] 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.

[23] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. *CoRR*, abs/1812.02378, 2018.

[24] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Conference on Computer Vision and Pattern Recognition*, 2018.