Incremental Image Generation using Scene Graphs

Sushant Mehta[*]
Language Technologies Institute
Carnegie Mellon University
sushantm@andrew.cmu.edu

Shubham Agrawal*
Robotics Institute
Carnegie Mellon University
sagrawa1@andrew.cmu.edu

Anuva Agarwal*
Language Technologies Institute
Carnegie Mellon University
anuvaa@andrew.cmu.edu

Gaurav Mittal*
Robotics Institute
Carnegie Mellon University
gauravm@andrew.cmu.edu

Tanya Marwah*
Robotics Institute
Carnegie Mellon University
tmarwah@andrew.cmu.edu

Abstract

Recent years have witnessed some very exciting developments in the domain of generating images from scene-based text descriptions. These approaches have primarily focused on generating images from a static text description. They are limited to generating images in a single pass without allowing to generate an image incrementally based on an incrementally additive text description (something that is more intuitive and similar to the way we describe an image). To this end, we propose a method that enables the underlying model to generate an image incrementally based on a sequence of graph of scene descriptions (scene-graphs). We propose a recurrent network architecture such that the cumulative image generated at any point in the sequential generation is consistent with the previously generated images. Our model utilizes Graph Convolutional Networks (GCN) to cater to variable size scene graphs along with GAN based image translation networks to generate realistic multi-object images with high amount of variability. We demonstrate our model's capability to generate context preserving scene-graph based image sequence using multi-modal datasets such as Coco-Stuff which have multi-object images along with annotations describing the visual scene.

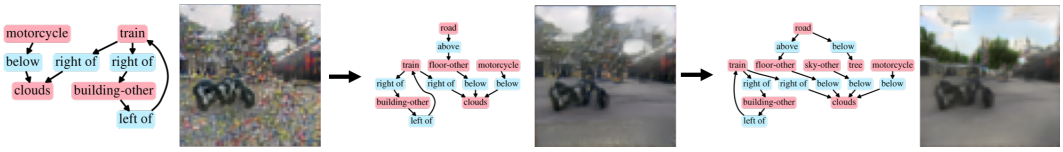

Figure 1: We propose an image generation framework capable of generating images from scene graphs, and then modify the image based on modifications to the scene graph, without losing previously generated image content.

1 Introduction

To truly understand the visual world our models should be able to not only recognize images but also generate them. Generative Adversarial Networks, proposed by [4] have proven immensely useful in generating real world images. GANs are composed of a generator and a discriminator that are trained with competing goals. The generator is trained to generate samples towards the true data distribution

[*]Authors contributed equally

Preprint. Work in progress.

to fool the discriminator, while the discriminator is optimized to distinguish between real samples from the true data distribution and fake samples produced by the generator.

The next step in this area is to generate customized images and videos in response to the individual tastes of a user. A grounding of language semantics in the context of visual modality has wide-reaching impacts in the fields of Robotics, AI, Design and image retrieval. To this end, there has been exciting recent progress on generating images from natural language descriptions. Conditioned on given text descriptions, conditional-GANs [13] are able to generate images that are highly related to the text meanings. Samples generated by existing text-to-image approaches can roughly reflect the meaning of the given descriptions, but they fail to contain necessary details and vivid object parts.

Leading methods for generating images from sentences struggle with complex sentences containing many objects. A recent development in this field has been to represent the information conveyed by a complex sentence more explicitly as a scene graph of objects and their relationships [8]. Scene graphs are a powerful structured representation for both images and language; they have been used for semantic image retrieval [9] and for evaluating [1] and improving [11] image captioning. In our work, we propose to leverage these scene graphs by incrementally expanding them into more complex structures and generating corresponding images.

A visualization of our framework's outputs with a progressively growing scene graph can be seen in Fig. 1. We can see how at each step new objects get inserted into the image generated so far without losing the context.

To summarize, we make the following contributions:

- The first framework for image generation that allows the image to be incrementally modified, such as adding an object in the image next to an existing one. The output images reflect the changes accurately, while preserving the previous image as much as possible.
- Improvement in image quality upon existing image generation baselines for complex datasets like MS COCO, while preserving object-relationship semantics

2 Related Work

Image generation from text. Generating images from text descriptions is of great interest, both from a computer vision perspective, and a broader artificial intelligence perspective. Since the advent of Generative Adversarial Networks (GANs), there have been many efforts in this direction [6].

[12] proposed a framework based on an LSTM and a conditional GAN to incrementally generate an image using a sentence. The words in the sentence were encoded using word2vec, and passed through an LSTM. A skip-though vector representing the semantic meaning of an entire sentence is used as the conditioning for the GAN. However all of these works mostly focus on generating images with single objects (such as faces or flowers or birds). Even withing these objects, the avenues of variance is quite limited. Generating more complex scenes with multiple objects and specific relationships between those objects is an even harder research problem.

[20] proposed an architecture based on multiple GANs stacked together, generating images in a course-to-fine manner. They later also proposed arranging the generators in a tree like structure for improved results [21]. More recently [5] proposed an end-to-end pipeline for inferring scene structure and generating images based on text descriptions. A similar approach was taken by [17], where they used attention-based object and attribute decoders to infer bounding box locations of objects in the scene. However for images with several objects such as in COCO-Stuff, the captions are often not descriptive enough to capture all the objects. Furthermore, the captions don't describe the relations between the objects in the image effectively. AttnGANs also begin with a low-resolution image, and then improves it over multiple steps to come up with a final image. However, there's no mechanism to capture consistency during incremental image generation. A more detailed failure case analysis is done in Section 6.

Most recently, [8] proposed to use scene-graphs as a convenient intermediate for image synthesis. Scene-graphs provide an efficient and interpretable representation of the objects in an image and their relationships. The input scene graph is processed with a graph convolution network which passes information along edges to compute embedding vectors for all objects. These vectors are used to predict bounding boxes and segmentation masks for all objects, which are combined to form a coarse

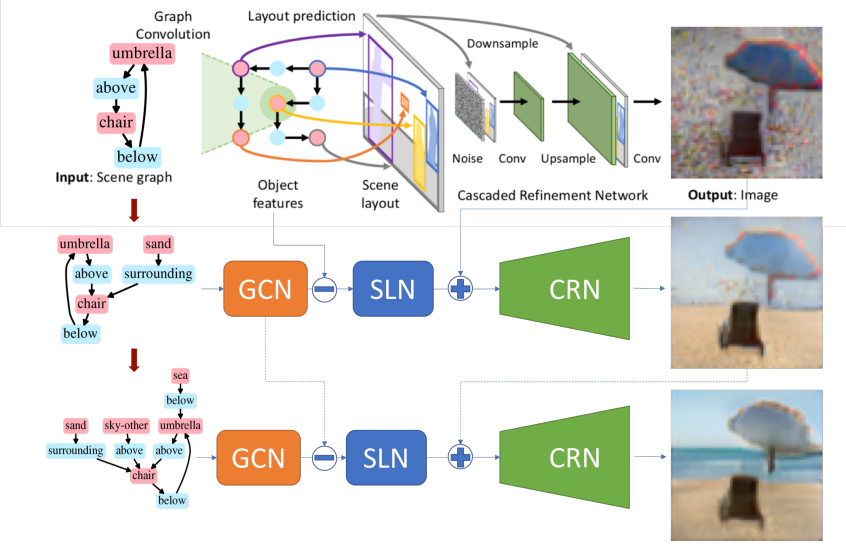

Figure 2: Model architecture for incremental image generation using scene graphs

scene layout. The layout is passed to a cascaded refinement network which generates an output image at increasing spatial scales. The model is trained adversarially against a pair of discriminator networks which ensure that output images look realistic. However this model does not account for object saliency or temporal consistency in the generated images. A more detailed failure case analysis is done in Section 6.

Image modification. One of the major avenues in research in computer vision has been the task of "inpainting", i.e. filling in a missing region in an image. Earlier approaches relied on finding nearest-neighbour matches within the image or in a dataset to complete missing regions. More recently, works such as [19, 7] have used architectures such as fully convolutional nets or GANs, coupled with discriminators, to complete images. [16] proposed a GAN based framework for removing objects in an image by generating a mask and inpainting the masked area. They used a combination of 3 adversarial losses based on object classification and image and mask realism.

Text to Scene graph. It is also of great research interest to be able obtain precise scene graphs from detailed text descriptions of an image. Earlier works [3] have proposed parsing the dependency structure of detailed text descriptions and converting them into query formats for a 3D cad model database. More recently, [15] proposed rule-based and classifier-based parsers for obtaining such graphs from descriptions.

3 Method

As in most modern conditional image-generation formulations, we follow a generative-adversarial approach to image generation. Here the adverserial network penalizes the network based on how realistic the generated images are, as well as whether the required objects are present in them. Furthermore, a key part of the task is to preserve the relations between objects as specified in the scene graph in the generated image as well.

3.1 Image generation from scene graphs

For our baseline approach for generating images from scene graphs, we adopt the architecture proposed by [8]. The architecture consists of 3 main modules, a Graph Convolution network (GCN), Layout Prediction Network (LN) and a Cascade Refinement Network (CRN), which we describe in more detail below.

Graph Convolution Network. The Graph Convolution Network (GCN) is composed of several graph convolution layers, and can operate natively on graphs. GCN takes an input graph and computes new vectors for each node and edge. Each graph convolution layer propagates information along edges of the graph. The same function is applied to all graph edges, which ensures that a single convolution layer can work with arbitrary shaped graphs.

Layout Prediction Network. The GCN outputs an embedding vector for each object. These object embedding vectors are used by the layout prediction network to compute a scene layout. This layout is computed by predicting a segmentation mask and bounding box for each object. Mask regression network and a box regression network are used to predict a soft binary mask and a bounding box, respectively. The layout prediction network hence acts as an intermediary between the graph and image domanins.

Cascade Refinement Network. Given a scene layout, the Cascade Refinement Network (CRN) is responsible for generating an image which respects the object positions in the scene layout. The CRN consists of a series of convolutional refinement modules. The spatial resolution doubles between modules; which ensures that image generation is happening in a coarse-to-fine manner. The scene layout is first downsampled to the module input resolution and then fed to the module along with the previous module's output. Both these inputs are concatenated channelwise and then passed to a pair of 3×3 convolution layers. This output is upsampled using nearest-neighbor interpolation and then passed to the next module. The output from the last module is finally passed to 2 convolution layers to produce the output image.

3.2 Sequential generation of images with context preservation

Our method allows for preserving context across the sequentially generated images by conditioning subsequent steps of image generation over certain information from previous steps.

- We extend [8] with a recurrent architecture that generates images using incrementally growing scene graphs using the components discussed in previous section as shown in Figure 2.
- To ensure that the image generated in the current step preserves the visual context from the previous steps, we replace three channels of the noise passed to CRN with the RGB channels of the image generated in the previous step. This encourages the CRN to generate the new image as similar as possible to the previously generated image.
- Moreover, we want that SLN generates a layout corresponding to only the newly added objects as part of the scene graph. To this end, we remove the representations generated by GCN corresponding to the objects generated in previous steps.
- We do not have any ground truth for the intermediate generated images so we use perceptual loss for images generated in the intermediate steps to enforce the images to be perceptually similar to the ground truth final image. We do have L1 loss between the final image generated and the ground truth.

Concretely, we train the network with the following losses:

1. Adversarial losses : We use an image level and an object level discriminator to ensure realism of the images and presence of the objects. These are trained as in the regular GAN formulation : $\mathcal{L}_{GAN} = E_{x \sim p_{real}} logD(x) + E_{x \sim p_{fake}} log(1 - D(x))$
2. Box loss: Penalizes L1 distance between ground truth boxes from MS COCO vs the predicted labels as $\mathcal{L}_{box} = \sum_i^n ||b - b'||$
3. Mask loss: Penalized difference between the masks predicted vs the ground truth masks, using cross entropy loss.
4. L1 pixel loss : Penalizes the difference between the ground truth image from MS COCO and the final generated image at the end of the incremental generation. L1 pixel losses are also used to penalze the difference between the previous and current generated image. $\mathcal{L}_{pixel} = ||I - I'||$

Thus we can provide additional supervision on the coordinates of the bounding boxes predicted by the layout network, to explicitly ensure the relations are preserved.

3.3 Image quality enhancement

We also focus on improving the quality of the images generated by the baseline model. By default, the images are generated at a 64x64 resolution. We later show in our experiments that our recurrent network and modifications to the loss terms by itself lead to improvements in the image quality. However, we further seek to enhance the generated images by drawing on insights drawn from StackGAN [20]. The authors if StackGAN proposed a two-stage coarse-to-fine generation framework that generated an image conditioned on a text embedding. They proposed a conditioning augmentation, which augmented the size of the training data and lead to a smoother manifold of the GANs. For our purposes, we remove the first stage of the StackGAN, instead using the 64x64 image generated by our network, as the input to the second stage. For the conditioning, we use the layout matrix generated by the LN, with the same conditioning augmentation applied. We skip the spatial replication as the Layout already has spatial info and is of the same dimensions as the image. The output of the generator is a higher resolution 256x256 image.

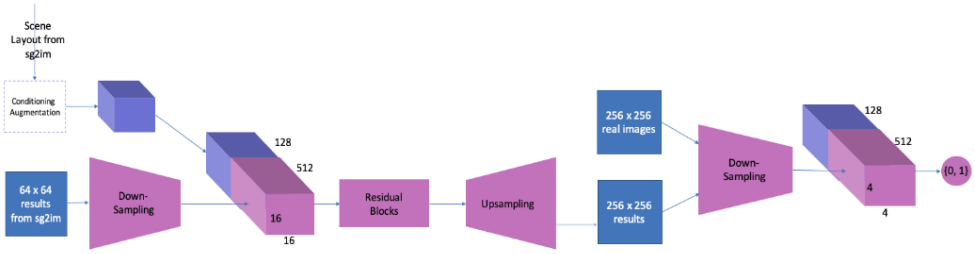

Figure 3: Our proposed architecture for image quality enhancement based on [20]. The 64x64 image generated by the main architecture is concatenated with a conditioning augmented layout map generated by the SLN. This is fed to the upsampling generator, which generates a 256x256 output. The images are sent to the discriminator along with the layout embedding.

4 Experiments

4.1 Dataset

We perform experiments on the 2017 COCO-Stuff dataset [2] which augments a subset of the COCO dataset [10] with additional stuff categories. The dataset annotates 40K train and 5K val images with bounding boxes and segmentation masks for 80 thing categories (people, cars, etc.) and 91 stuff categories (sky, grass, etc.).

We follow the procedure described in [8] to construct synthetic scene graphs from these annotations based on the 2D image coordinates of the objects, using six mutually exclusive geometric relationships: left of, right of, above, below, inside, and surrounding. We create three splits for each image based on the number of objects in it. We randomly select 50% of the objects for the first split and incrementally add 25% objects for the next two splits. We then synthetically create separate scene graphs for each split. Note that we train three steps for incremental generation, but this can be easily extended to more number of steps.

To enable comparison against [8], we follow their dataset preprocessing steps. They ignore objects covering less than 2% of the image, and use images with 3 to 8 objects. They divide the COCO-Stuff 2017 val set into their own val and test sets, which contain 24,972 train, 1024 val, and 2048 test images. For fair comparison, we do the same.

Detailed qualitative and quantitative analysis of our results and comparisons against the baseline models follow in the next subsections.

Figure 4: Comparison of our results with baseline approaches–sg2im and AttnGAN

4.2 Qualitative Results

We compare the performance of our model against two baselines [8, 18] as can be seen in Figure 5. The scene graph in the first step contains two objects and the relationships between them. An additional object and its relationship with other objects is added to the scene graph at each of the next two steps. As can be seen, the first baseline [8] is able to capture the semantic context provided by the graph. However (i) it fails to preserve consistency over multiple passes and generates a completely new image for each scene graph, agnostic of what it had generated at the previous step and (ii) the images generate are of poor quality. The second baseline [18] produces visually pleasing and high resolution images but completely fails to capture any semantic context provided from the graph. Our model, on the other hand, is capable of incrementally adding new objects to the image created in the previous step in accordance with the relationships defined in the scene graph. Additionally, the quality of generated images is significantly improved, since at each step the model has to generate only a few objects rather than generating cluttered scenes with multiple objects, hence enabling it to better generate scene semantics.

4.3 Quantitative Evaluation

We use Inception Score [14] for evaluating the quality of the images generated from our models. Inception Score uses an ImageNet based classifier to provide a quantitative evaluation of how realistic generated images appear. Inception Scores were originally proposed with the following goals:

- The images generated should contain clear objects (i.e. the images are sharp rather than blurry), (or, for image x and label y, $p(y|x)$ should be low entropy). In other words, the Inception Network should be highly confident there is a single object in the image.
- The generative algorithm should output a high diversity of images from all the different classes in ImageNet, or $p(y)$ should be high entropy.

The inception scores are reported in Table 1. We compare the inception score of the images generated from the baseline model sg2im with the full scene graph of the ground truth images. For our sequential generation model, we report the scores over three steps of generation, where at the third step the scene graph is the full scene graph corresponding to the ground truth image. We observe that due to our modified loss formulation and incremental generation, our model performs better in inception scores from Stage 0 itself. We also note that Stage 1 performs the best. From our observations this is because the vividness of the image colors and object definitions is the best at the stage 1, and begin to fade out stage 2 onwards.

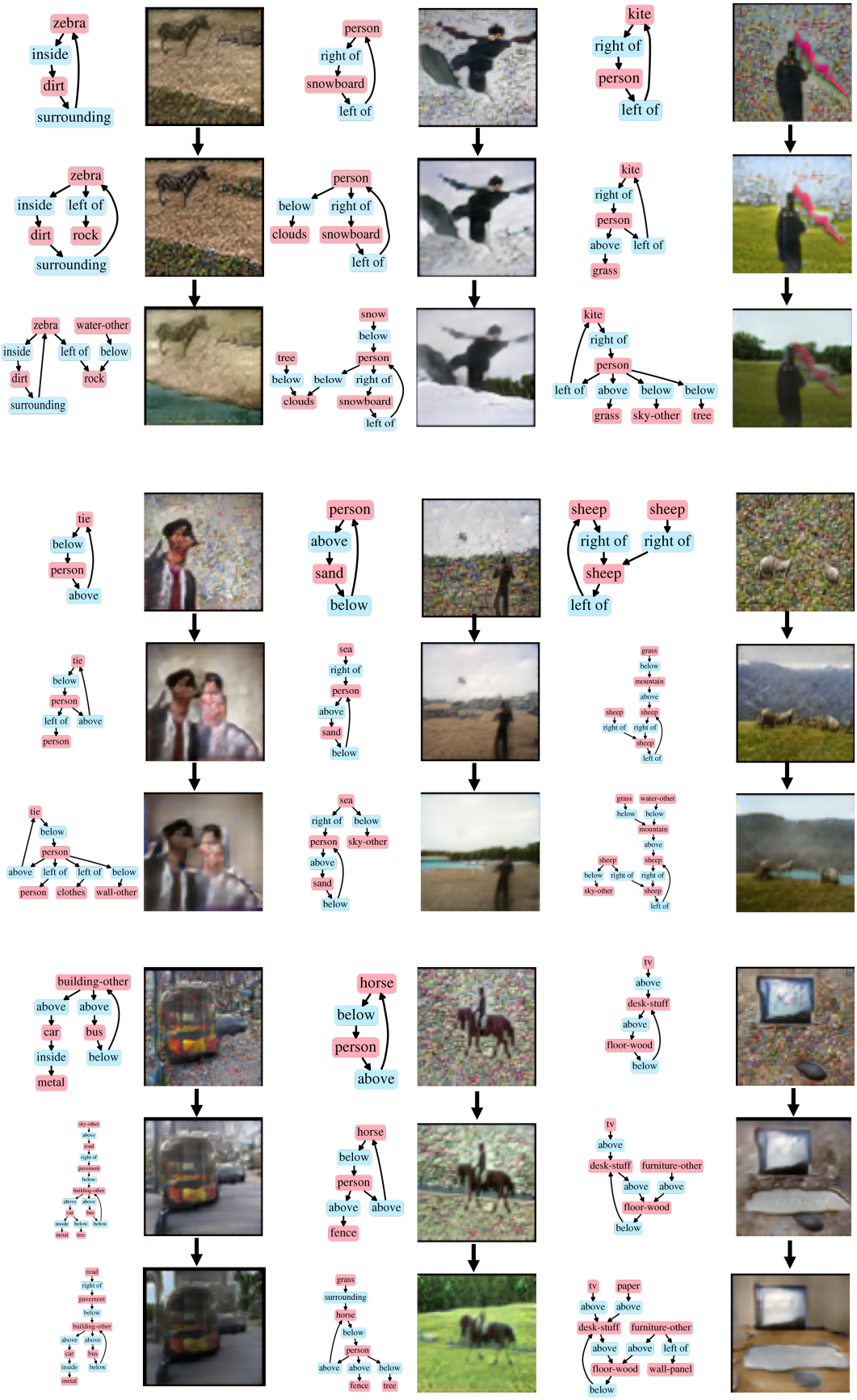

Figure 5: Sample outputs of our pipeline, visualized over 3 steps of generation.

Table 1: Inception Scores for Ground Truth Images, images generated from Sg2im and the three steps from our model

Ground Truth	Sg2im	Step-0 (Ours)	Step-1 (Ours)	Step-2 (Ours)
6.13	3.05	3.68	**5.02**	4.14

4.4 Qualitative results of image quality enhancement

We visualize the results of our StackGAN based image quaity enhancement in Fig. 6. We observe a considerable improvement in image quality with our StackGAN based refinement network approach. For would like to note that for training this in parallel, we used the images generated by our baseline, sg2im, as the coarse 64x64 input. We expect an even further quality improvement by retraining using the images generated by our model.

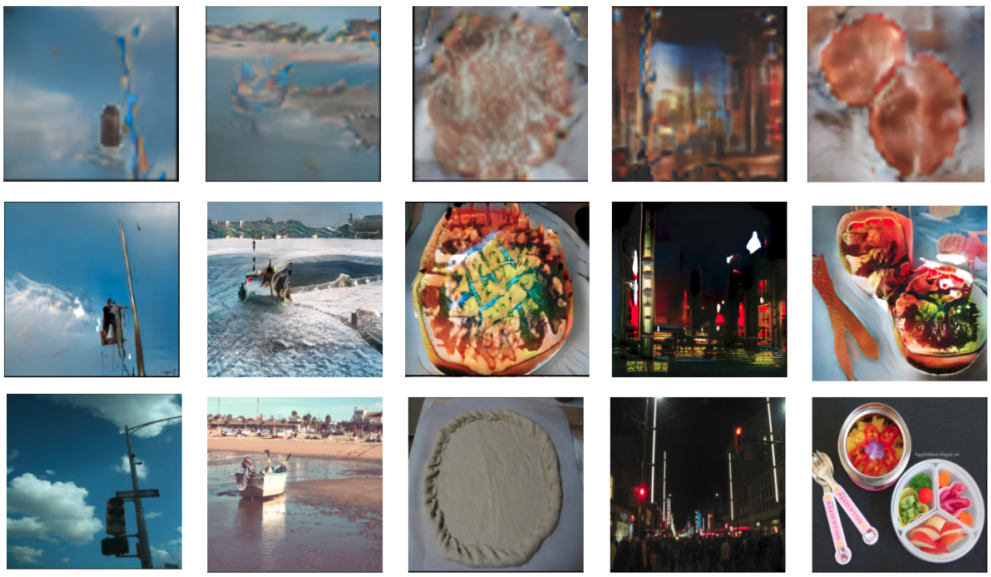

Figure 6: Sample images generated using our StackGAN based refinement network proposed in Section 3.3. Top row: 64x64 images generated from sg2im, our baseline model. Middle row: 256x256 outputs generated by our StackGAN based enhancement. Bottom row: Ground truth images in MS COCO

5 Discussion and Future Work

In this paper, we proposed an approach to sequentially generate images using incrementally growing scene graphs with context preservation. Through extensive evaluation and qualitative results, we demonstrate that our approach is indeed able to generate an image sequence that is consistent over time and preserves the context in terms of objects generated in previous steps. To the best of our knowledge, this is the first attempt in the direction of incremental image generation using scene composition. In future, we plan to explore generating end-to-end with text description by augmenting our methodology with module to generate scene graphs from language input. While scene-graphs provide a very convenient modality to capture image semantics, we would like to explore ways to take natural sentences as inputs to modify the underlying scene graph. The current baseline method does single shot generation by passing the entire layout map through the Cascade Refinement Net for the final image generation. We plan to investigate whether the quality of generation can be improved by instead using attention on the GCN embeddings during generation. This could also potentially make the task of only modifying certain regions in the image easier. Further, we plan to explore better architectures for image generation through layouts for higher resolution image generation.

6 Acknowledgement

We would like to thank Dr L.P. Morency and the 11-777 TAs for their valuable guidance throughout the project. We also thank Justin Johnson for sharing his implementation of sg2im.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. *CoRR, abs/1612.03716*, 5:8, 2016.

[3] Bob Coyne and Richard Sproat. Wordseye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 487–496. ACM, 2001.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

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

[6] He Huang, Phillip S Yu, and Changhu Wang. An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469*, 2018.

[7] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

[8] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs.

[9] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[10] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[11] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *Proc. IEEE Int. Conf. Comp. Vis*, volume 3, page 3, 2017.

[12] Xu Ouyang, Xi Zhang, Di Ma, and Gady Agam. Generating image sequence from description with lstm conditional gan. *arXiv preprint arXiv:1806.03027*, 2018.

[13] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[14] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[15] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015.

[16] Rakshith Shetty, Mario Fritz, and Bernt Schiele. Adversarial scene editing: Automatic object removal from weak supervision. *arXiv preprint arXiv:1806.01911*, 2018.

[17] Fuwen Tan, Song Feng, and Vicente Ordonez. Text2scene: Generating abstract scenes from textual descriptions. *arXiv preprint arXiv:1809.01110*, 2018.

[18] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *arXiv preprint*, 2017.

[19] Raymond A Yeh, Chen Chen, Teck-Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, volume 2, page 4, 2017.

[20] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint*, 2017.

[21] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1710.10916*, 2017.