Wildlife Image Generation from Scene Graphs

Yoshio Rubio, Marco A. Contreras-Cruz
Samsung Research America, Mountain View, United States
Samsung Research Tijuana, Tijuana, Mexico
{y.rubio, marco.cruz}@samsung.com

Abstract

Image generation from natural language descriptions is an exciting and challenging task in computer vision and natural language processing. In this work, we propose a novel method to generate synthetic images from scene graphs in the context of wildlife scenarios. Given a scene graph, our method uses a graph convolutional network to predict semantic layouts, and a semi-parametric approach based on a cascade refinement network to synthesize the final image. We test our approach on a subset of COCO dataset, which we call COCO-Wildlife. Our results outperform the baselines, both quantitatively and qualitatively, and the visual results show the ability of our approach to generate stunning images with natural interaction between the different objects. Our findings show the potential to expand the use case of the proposed method to other contexts where scale and realism is fundamental.

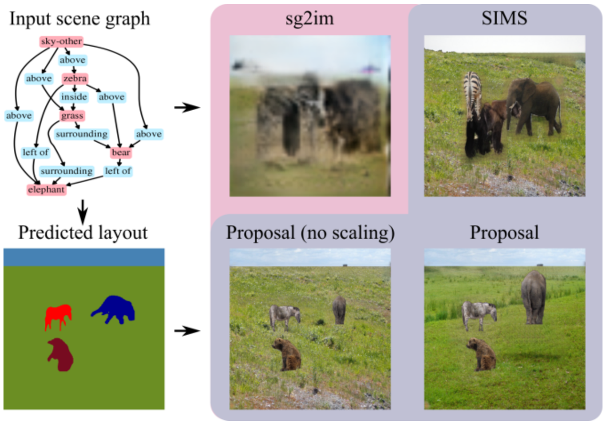


Figure 1. Current approaches for synthetic image generation, such as SIMS [19], require highly detailed semantic layouts to generate realistic images, which limits its use on T2I. We overcome this limitation by taking into consideration artifact removal and scaling to preserve realism in the composition.

1. Introduction

The automatic generation of photorealistic images from text (T2I) is a significant and frontier problem in natural language processing and computer vision [2]. It has received substantial attention due to its enormous potential applicability, as T2I methods can be useful for image editing, creating artistic images, computer-aided design, video games, and virtual reality [1, 2, 14, 21, 27]. Much progress in this area has been lead by the advances of generative adversarial networks (GANs) [3]. GANs have shown great success in generating realistic images that follow a known training distribution. They are composed of two models trained adversarially: the generator and discriminator. The generator tries to fool the discriminator by generating realistic synthetic images, while the discriminator tries to classify each image as real or synthetic.

For T2I, GANs architectures are conditioned on text to generate realistic synthetic images that match the text description. For instance, Reed *et al*. [22] proposed the first GAN model conditioned on text. They use a pre-trained text encoder that extracts features from text, and feeds the generator and discriminator with these features. This method can only generate low-resolution images (64×64). In order to address this issue, Zhang *et al*. [33] proposed StackGAN, where multiple generators are in charge of progressively refine the image. Most of the parameters of the generators are shared and used to produce higher resolution images (256×256). This approach significantly changes synthetic images even if only some words related to attributes change.

GAN models based on attention emerged to better control the image generation by producing regions associated with the most relevant words [14, 30], enabling the system to generate similar images when minor changes in the sentences occur. Another challenge is the semantic consistency between the generated image and the text description. Some advances in this problem use a cycle consistency approach to align the image with the input text by using the text description of the generated image [20]. Some other approaches have focused on improving the quality of the images. Yin *et al*. [32] proposed a Siamese architecture to generate images for a given text. Each branch of the architecture contains an array of generators and discriminators, and the input text for each branch is different but has the same meaning. Tao *et al.* [24] proposed a dynamic memory network that synthesizes higher resolution images with a single generator and discriminator. They gain visual details by using residual blocks. Other recent works have used a zero-shot approach with autoregressive transformers for T2I [1, 21].

Although the methods conditioned on text have shown impressive progress, they are still far from generating complex images where many objects and relationships exist. Some other works have proposed to use more explicit representations to deal with complex descriptions [10]. These methods use additional annotations during training, *e.g.*, captions [9], mouse traces [12], semantic layouts [19], and scene graphs [7]. Among these methods, the scene graphs are powerful structures to represent objects, attributes, and interactions between objects [8]. The use of scene graphs for automatic photo-realistic image generation is a relatively new research area, but its promising results show that it is a step forward to close the gap between text descriptions and image generation [6, 15, 17, 25, 26].

Typically, scene graphs are used to generate intermediate semantic layouts that are used as a reference to generate the final synthetic image. These methods for image generation can be classified into non-parametric, semi-parametric, and parametric approaches. Parametric models have the advantage of end-to-end training but lose the ability of non-parametric and semi-parametric models that draw from source images to obtain detailed object appearance [19]. The biggest issue with the parametric and semi-parametric models, as indicated in Figure 1, is that they operate over detailed semantic layouts like GauGAN [18], SIMS [19], and pix2pixHD [28].

In this work, we present a novel method to generate wildlife images from scene graphs using a semi-parametric approach that mixes the strength of generative models to learn from examples, and the performance of non-parametric models in creating images with high-quality objects. Our proposal consists of two steps. First, our proposal uses an architecture based on the graph convolution network [7] to generate high-quality semantic layout from scene graphs. Then, it uses an adaptation of SIMS [19] to generate synthetic images.

The proposed approach does not require detailed semantic layout as most of the parametric approaches. Besides, it can generate realistic images with a low number of image artifacts compared to the semi-parametric methods. We perform experiments in a subset of COCO with images related to wildlife. We select the wildlife subset since the interaction of animals in natural scenarios mixes the simplicity of having a limited number of objects and the complexity of generating realistic images under a high number of variables such as scale, depth loss, illumination, textures, variability in scenarios, among others.

The rest of this paper is structured as follows. Section 2 describes the method to generate wildlife synthetic images from scene graphs. Section 3 presents the experimental setup, datasets, and implementation details. Section 4 shows the main findings and results of the research. Finally, Section 5 shares our conclusions and perspectives of future work.

2. Method

The proposed method can be seen in Figure 2 and is composed of two blocks: the semantic layout generator which creates semantic layouts from scene graphs; and the synthesis block generates realist synthetic images from the layouts. In the following subsections, we describe these blocks in detail.

2.1. Semantic layout generator from scene graphs

The input for our proposed method is a scene graph, which is a data structure that encodes objects and their interactions in a scene [8]. A scene graph can be defined as a tuple (O, E), where $O = \{o_1, ..., o_n\}$ is a set of objects belonging to set of categories C, and $E \subseteq O \times R \times O$ is a set of directed edges of the form (o_i, r, o_j) where $o_i, o_j \in O$ and r is part of relationship categories R [7]. In Figure 2 we show a scene graph as input of our system.

The system uses a shorter version of sg2im [7], which we called m-sg, to predict a semantic layout from a scene graph. The semantic layout is an image $S \in \{0, 1\}^{h \times w \times c}$, where $w \times h$ is its size, and c is the number of classes.

Our model removes the cascaded refinement network (CRN) and we only kept the box loss $\mathcal{L}_{box}$ (mean square error from predicted and ground-truth boxes) and mask loss $\mathcal{L}_{mask}$ (pixel-wise binary cross-entropy between ground-truth and predicted layouts).

2.2. Semantic layout enhancement

Since m-sg is based on sg2im, it provides good information about the position of the objects but it generalizes their shapes with blobs that are the average of the object examples.

Therefore, we propose a method to enhance the layout by replacing the blobs with realistic shapes of objects, and filling the empty background areas (black pixels in the map). This process uses a bank of shapes that stores animals in five different poses: *left*, *right*, *back*, *front*, and *laying down*. Then, the real shapes are selected at random from the bank to replace the original blobs in the semantic layout. Any empty region of the original layout is filled with the closest background class. For the construction of the bank of

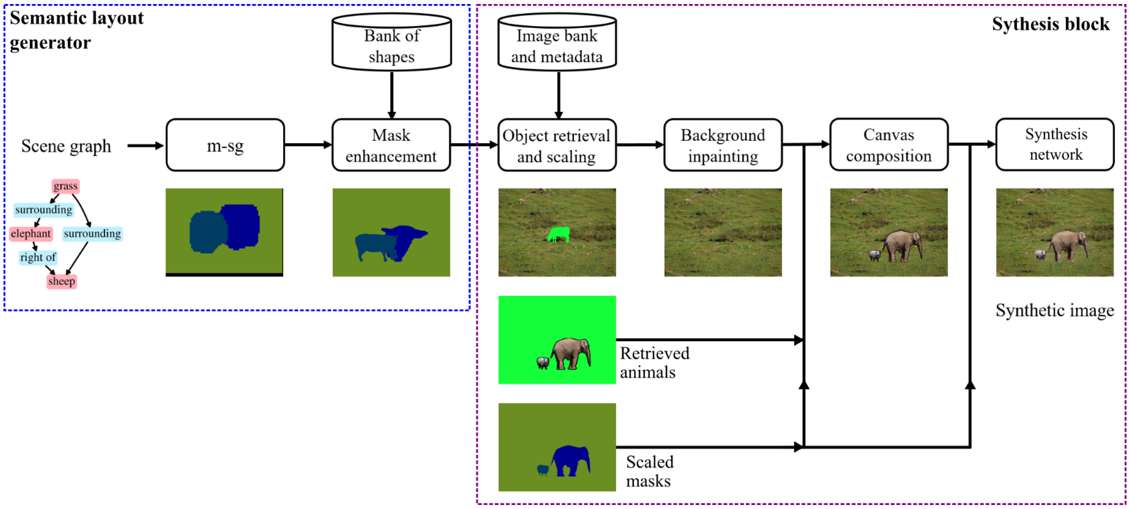


Figure 2. Proposed pipeline for wildlife image generation from scene graphs. The semantic layout generator takes a scene graph and creates a detailed mask, then the synthesis block makes a composition based on the mask of the proper background, scales the objects, fills any gaps using image painting, and removes artifacts and provide color balance with a cascade refinement network.

shapes, we used the Panoptic-FPN network [11] in a set of images of animals from internet.

2.3. Image synthesis

After the semantic layout is generated, the layout is used as input for the synthesis block to generate realistic images. Our approach is semi-parametric similar to SIMS [19] and PasteGAN [15], but we address object scaling, occlusion, and image composition.

2.3.1 Image bank and metadata

The first step in the synthesis block is to construct an image bank from a dataset. This bank M is composed of individual segments extracted from the dataset, where each segment P_i is associated with its binary mask $P_i^{mask} \in \{0,1\}^{h \times w \times c}$ and color image $P_i^{color} \in \mathbb{R}^{h \times w \times 3}$.

2.3.2 Object retrieval and scaling

During testing, given a semantic layout S, the goal is to find the best matching segment in the image bank M for each element in S. Let $S_j \in S$ be a semantic segment of S, and $S_j^{mask} \in \{0,1\}^{h \times w \times c}$ be its corresponding binary mask. The objective is to find the segment in M with the maximum intersection-over-union (IoU) score,

$$\sigma(S_j) = \arg\max_i \; \mathrm{IoU}(P_i^{mask}, S_j^{mask}) \tag{1}$$

where i iterates over all the segments in M of the same class as S_j.

In contrast with [19], we do not discard segments in different positions and scales. Instead, we translate the candidate segment P_i to the position of S_j and resize P_i to the scale of S_j respecting the proportion of P_i. Then, we apply the IoU score.

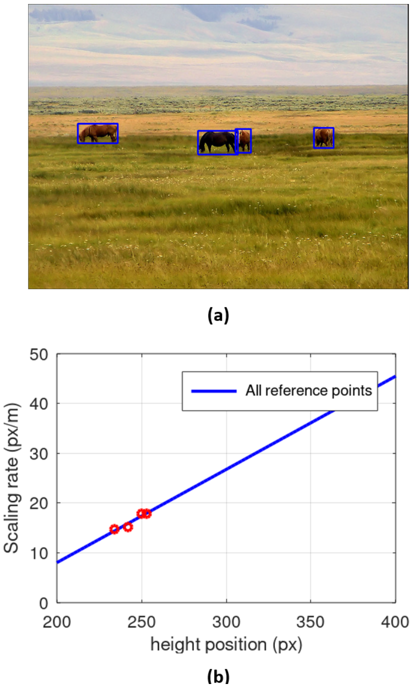
We estimate the scaling information by using the animals in the training images as reference. We used the Faster RCNN X101 network [29] to extract the bounding boxes of all these reference animals.

The reference animals per image and the information about their average sizes in the real world are used to construct a linear model that predicts the scaling rate given the height position of the animal. We define the scaling rate as the ratio between the height of the animal in pixels and its average height in meters. Figure 3 shows an example of the scaling rate model in a training image by comparing the height of the horses based on their position in the image.

The system searches for all the terrains in the image bank where the scaling rate prediction model balances the size of the animals, the overlap between animals, and the IoU score concerning the semantic segment of the terrain S_j^{mask} by calculating $\sigma_{terrain}(S_j)$,

$$\begin{aligned}\sigma_{terrain}(S_j) = \arg\max_i w_1 \mathrm{IoU}(P_i^{mask}, S_j^{mask}) \\ + w_2 h_{min} + w_3(1 - \mathit{overlap})\end{aligned} \tag{2}$$

where i iterates over all the terrains of the same class than S_j, $h_{min} \in [0,1]$ is the new normalized height of the smaller animal in the terrain, *overlap* $\in [0,1]$ is the estimated normalized overlapping between the bounding boxes of the rescaled animals, and w_k, $k = \{1,2,3\}$ are the



(a)

(b)

Figure 3. Scaling approach: using the bounding boxes of the animals (a), we create a linear regression model (b) with the height position of the animal in the image and the scaling rate (pixels of the bounding box divided by average height of real animals).

weights for each component. Terrains that generate hard to visualize animals (too big or small) are discarded before using Equation 2.

2.3.3 Image inpainting

The segments of the background classes (terrains and sky) typically have missing regions. These missing regions are undesired for the image synthesis, because they can generate image artifacts. Hi-Fill [31] is a recent light-weight inpainting model with excellent results for irregular holes of considerable sizes, which are the types of holes that appear regularly in our image segments. Hi-Fill uses a convolutional network to predict a low-resolution inpainted image, and it up-samples this image to generate a large blurry version of the image, then it uses a contextual residual aggregation mechanism to produce an inpainted image with high resolution.

In this work, we use Hi-Fill trained with Places [34] dataset to fill all the missing regions in the selected background. Also, Hi-Fill is used to stretch the sky and the terrain to reduce the missing regions in the frontiers between them.

2.3.4 Canvas composition

Once we retrieved the animals (with proper scale) and applied the image inpainting to the background, we merge these elements into a canvas. The canvas uses the semantic layout as reference to blend all the objects into the image. To naturally blend the object boundaries, we implemented inside and outside boundary elision as suggested by [19].

2.3.5 Synthesis network

Typically, the recovered segments in the canvas have different illuminations, color temperature, and artifacts between the borders. Therefore, we use a CRN proposed on [19] to balance the color of the recovered object and to smooth the transitions between them as a final step. The SIMS-Synthesis network receives a semantic layout and its canvas and generates the final synthetic image. We adapted the architecture of the original network to fit the new image size (640×480), with most of the upsampling and convolutional filters adjusted to the new resolution, and trained with the same perceptual loss based on feature activations as in [19].

We decided to use this network as a refiner instead of the original implementation because the network is prone to generate artifacts in missing regions. This behavior in CRNs is originated due to the loss of semantic information by normalization layers as studied in [18]. .

In Figure 2, we can observe that after segment retrieval both animals have black borders and the sheep has a colder color that does not match the background. This is fixed after the final synthesis step.

3. Experimental setup

In the following sections, we describe the datasets, the baseline methods, the metrics, and the details of the experiments.

3.1. COCO-Wildlife subset

We selected from the Common Objects in COntext (COCO) dataset [16] a subset that we called COCO-Wildlife. In this subset, we selected images that contain at least one object of the context categories (bush, dirt, grass, hill, leaves, mountain, mud, river, rock, sand, sea, stone, tree, sky-other, plant-other, branch, clouds, fog, and ground-other) or the animal categories (bird, horse, sheep, cow, elephant, bear, zebra, and giraffe). Our animal classes from COCO do not include pets to prevent the presence of urban scenarios. In total, we selected 20K images to train m-sg and 850 for the validation set. We selected this specific set of classes, since it has semantically similar classes with high variability between their objects, which prevents having images with very similar composition.

From the COCO-Wildlife subset, we created a class-balanced subset where the corresponding semantic layouts of the images had more than 50% of the pixels different from black. We used this subset (around 4K images) to train the SIMS-Synthesis network. In the synthesis phase, we curated the subset of COCO-Wildlife to remove images with sky or land sections with big segments missing. This curated subset contains around 3K images.

3.2. Open Images V6 subset

Open Images V6 [13] contains around 1.9M images with 16M bounding boxes and 600 object classes. It provides high-quality segmentation masks for the objects. We selected a subset of Open Images V6 with the animal categories, and manually curated the subset to obtain the examples with the best-looking masks. These animals replaced the animal segments from COCO-Wildlife in the synthesis phase.

3.3. Baseline methods

Sg2im [7] is widely used as benchmark in the task of image generation from scene graphs [6, 26]. We adopted sg2im for the comparison because it is the closer method to our proposed pipeline. We also used SIMS [19] for the comparison because our method is inspired by some of its elements. As SIMS does not have a layout generator, we used m-sg with the mask enhancement block to perform a fair comparison.

3.4. Evaluation metrics

To measure the performance of the proposed method, we use five metrics. The first metric is the relation score proposed in [25], which measures the agreement between the relations specified by the scene graph and the relations presented in the generated layouts. The second metric is the widely adopted Inception score (IS) [23] that uses the pre-trained Inception V3 network to predict the class probability and computes the score that penalizes bad quality images with lower diversity. The third metric, called Fréchet Inception Distance (FID) extracts the visual features of the last pooling layer in Inception V3 and computes the distance between sets of real and synthetic images. The authors [4] claimed that this metric correlates better to human judgments compared to IS. As IS and FID do not consider the specified objects in the synthetic images, we used the Semantic Object Accuracy (SOA) metric [5] to alleviate this issue. This metric uses a pre-trained object detector to identify if the images contain the desired objects (animals). We used the Faster RCNN X101 network trained on COCO as the object detector, and applied the two variants of SOA, *i.e*. the recall as a class average (SOA-C) and the image average (SOA-I).

3.5. Experiments

The proposed method aims to generate synthetic images of high-visual quality and realism. Therefore, we designed a qualitative experiment to compare the generated synthetic images against a set of real images. In a second experiment, we investigated the composition and realism of the images. To achieve this, we randomly generated scene graphs to create synthetic images and evaluate them with IS, FID, relation score, and SOA.

3.5.1 Synthetic scene graphs from images

As described by Johnson et al. [7], we began by extracting scene graphs from a set of real images. The generator extracts six possible excluding geometric relationships between two objects: *right of*, *left of*, *above*, *below*, *inside*, and *surrounding*. It discards objects in the images with less than 2% of the image size, and selects images with 2 to 8 objects. Besides, it discards images without terrain. For the experiments, we used 141 images with the above characteristics from the COCO-Wildlife validation set.

Once we generated the scene graphs for the subset, we executed the systems to create the synthetic images from those scene graphs.

3.5.2 Randomly generated scene graphs

Given a set of animals $\mathcal{A} = \{$*horse*, *sheep*, *cow*, *elephant*, *bear*, *zebra*, *giraffe*$\}$ and terrains $\mathcal{T} = \{$*dirt*, *grass*, *mud*, *sand*, *ground-other*$\}$, we select one terrain at random and a maximum number of five animals per scene graph. We decide if the scene graph contains sky with a given probability. For the relationships, we always specify that the sky is *above* the rest of the elements. Regarding animal-animal relationships, we use an animal as reference and then we randomly select one relation pair between the following: *right of*-*above*, *right of*-*below*, *left of*-*above*, and *left of*-*below*. Then, in the same scene graph, only the two options of the selected pair are used to avoid contradictions. We also randomly select the relationship between the terrain and the animal. The animal can be *above* or *inside* the terrain, or the terrain can *surround* the animals. We randomly generated 2250 scene graphs for the experiment.

3.6. Implementation details

We performed all the experiments in a workstation with an i7-6850K, 32GB of RAM, and a Titan Xp with 12GB of VRAM. For sg2im, we used the original Torch implementation proposed by the authors [7]. Since m-sg is based on sg2im, it inherits most of its configurations but we set the mask loss weight to 0.1. From the 27 classes of the COCO-Wildlife subset, we reduced them to 19 by merging the classes that were semantically similar. As an example we merged *bush*, *grass* and *plant-other* into the *vegetation* class. We did the same for *tree*, *sky*, *rock*, *ground*, and *water* related classes. The training batch for all the experiments is 32, and the size of the layouts is 64 × 64. We trained both models for 410 epochs on the COCO-Wildlife.

For the SIMS-Synthesis, we changed the size of the convolutional layers in the encoder and decoder to match the input size of 640 × 480, as well as the number of classes to fit those of m-sg. For training , we used the same training parameters as the authors for 100 epochs. The rest of the blocks of the pipeline were implemented directly in Python.

4. Results

As previously indicated, we extracted the relationships between objects in the validation set of COCO-Wildlife and used them as input for the baseline methods and our proposed method. Figure 4 shows some synthetic images generated for the validation set by using sg2im (trained on COCO), sg2im-wildlife (trained on COCO-Wildlife), m-sg + SIMS (SIMS with the m-sg as a semantic layout generator), and the proposed method. To better contrast the visual results, we also show the reference images and their corresponding scene graphs. We sorted the scene graphs from simple to complex (top to bottom in the figure).

We can observe that the proposed method not only preserves the relationships specified by the scene graph but also generates detailed images. Even if we designed the proposed system for simple scenes (images with one terrain, sky, and animals), in some cases the method composes complex scenes with context objects such as mountains and trees (see rows 2, 3, and 4 in Figure 4). The baseline methods generated some interesting synthetic images with blurred regions. Although there are blurred image regions, the synthetic images appear to preserve the composition of the ground truth.

Now, we explore the performance of the baseline methods against the proposed method in a set of 2250 randomly generated scene graphs. We compared the performance in terms of the IS, FID, relation score, and SOA. Given that IS and FID resize the images to 299 × 299, we computed the metrics using input images of low-resolution (64 × 64) and images of higher resolution (640 × 480) to prevent any issues since sg2im and our proposal have different output sizes. We compared the performance of sg2im in two versions, *i.e.*, the version trained with COCO and COCO-Wildlife. For IS and FID, we used real images from the training and validation sets of COCO-Wildlife with at least one animal to compute the reference value.

Table 1 summarizes the results for each approach in the aforementioned metrics. We marked in bold the best method for each metric. We can observe that IS for real images of 64 × 64 is 7.29 and 8.67 for images of 640 × 480. Our proposed method is around 2 points below compared to the reference value. However, it outperforms sg2im by 1.5 points, and it is just 0.1 points below m-sg+SIMS in images of 64 × 64, and it outperforms sg2im by 2.4 points in images of higher resolution. The m-sg+SIMS on 640 × 480 images has a higher IS than our proposed approach by 1.2 points, and it is 1 point below compared to the reference set. We can also observe that the performance of sg2im is deteriorated with the model trained on COCO-Wildlife, which is explained by the reduction of the training instances compared to the COCO-trained model.

According to the FID, the proposed method outperforms the baseline methods. The proposed method scores 1.9 times less than sg2im trained on COCO-Wildlife in images of 64 × 64 and 3.9 times less in images of 640 × 480. Compared against m-sg+SIMS, the proposal is slightly better for both resolutions. Sg2im trained with COCO-Wildlife gives better performance than sg2im trained on COCO in this metric.

It is worth mentioning that both FID and IS fail to measure the realism of images. FID can measure the quality of synthetic images against artifacts or noise due to the generation process, and it may be able to measure some level of relationships between the objects, but it is blind to the composition based on scale and perspective. This factor is highly important for human evaluators when judging the realism of an image.

Regarding the relation score, the proposed method achieved better results than the baseline methods. Although our model is based on sg2im for the layout prediction, the improvement in the semantic layout and the scaling approach benefit the performance as shown in Table 1. For example, the proposed method improves 3.26% the relation score of sg2im, and 2.72% the trained model on COCO-Wildlife. No relation score was calculated from SIMS given that the input of this method does not generate a semantic layout, and it is fed directly by m-sg.

Finally, we compare the baseline method against our proposed method with the SOA. As sg2im generates images of low resolution, the pre-trained network used to detect the objects is unable to identify any object, so we discarded this metric for this model. For the semi-parametric approaches, we can observe that our proposal improved around 27% the two variants of the metric concerning m-sg + SIMS.

Figure 5 shows some good examples of generated synthetic images by using our proposed method. We present the scene graphs from simple to complex for each row, showing that the more complex scene graph contains at most five animals. When only one animal exists in the image, the proposed method tries to find a terrain that maximizes the size of the animal, generating synthetic images where the animal grabs attention, see rows 1 and 2. It is also clear that the proposed method can scale the animals correctly when multiple animals interact in the image, see rows 3 and

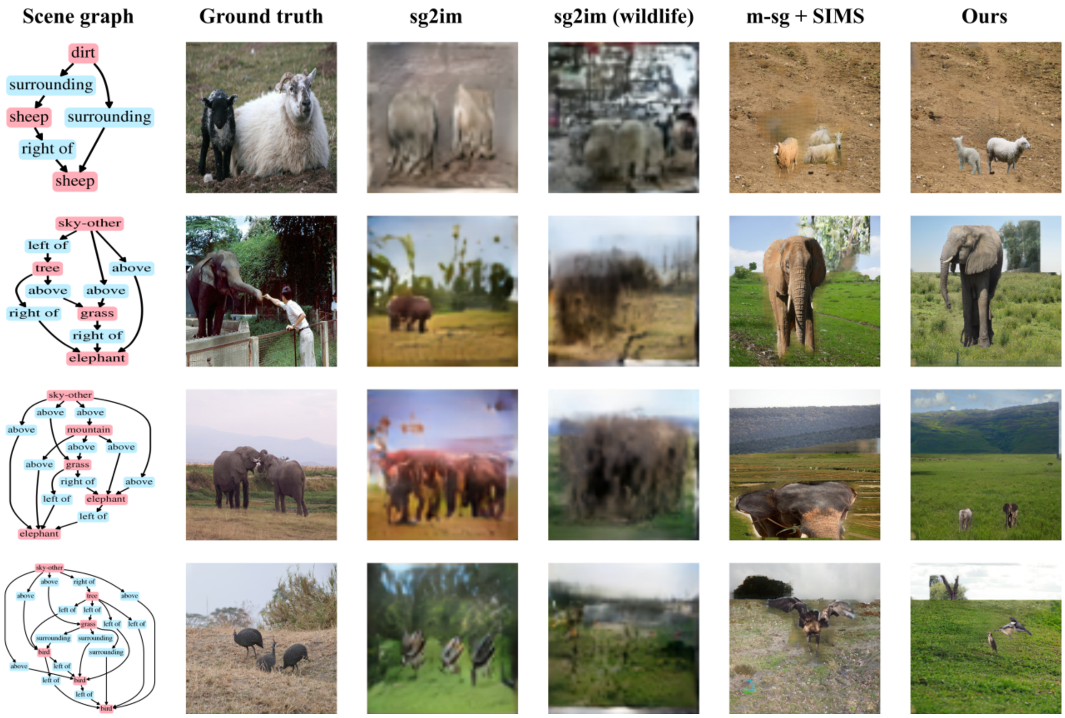


Figure 4. Visual comparison between sg2im (trained with COCO and COCO-Wildlife), m-sg+SIMS, and the proposed method.

Method	IS ↑		FID ↓		RS ↑	SOA-C ↑	SOA-I ↑
	64 × 64	640 × 480	64 × 64	640 × 480			
Real images	(7.2970, 0.4084)	(8.6722, 0.8862)	- - - -	- - - -	- - - -	- - - -	- - - -
sg2im	(4.0431, 0.1709)	(4.0390, 0.1874)	165.0762	298.1081	0.7499	- - - -	- - - -
sg2im-wildlife	(2.8169, 0.1403)	(2.8061, 0.1446)	157.0717	277.1845	0.7553	- - - -	- - - -
m-sg + SIMS	**(5.7984, 0.1938)**	**(7.6099, 0.4182)**	95.1993	76.5319	- - - -	0.6120	0.6137
Ours	5.6355, 0.2435)	(6.4558, 0.2617)	**81.7314**	**75.2954**	**0.7825**	**0.7785**	**0.7779**

Table 1. Quantitative comparison between the baseline methods and the proposed method by using Inception score (mean, std), Fréchet Inception Distance (FID), relation score (RS), and semantic object accuracy (SOA) at two different image resolutions.

4. Furthermore, we can note that our proposal has the exciting characteristic of generating images rarely found in search engines, *e.g.*, interactions between sheep and elephant, sheep and zebra, cow and zebra, cow and giraffe, among other combinations. Additionally, we can observe in Figure 5 the SIMS-Synthesis network effect: all the examples have an accurate color balance between objects and background with no visible artifacts.

Although the system can generate good-quality images under more complex situations, the method still presents issues in interactions between animals and context objects (hills, rocks, mountains, and trees) due to the quality of the predicted layouts. Besides, the quality of the image bank plays and essential role in our approach. Some issues are directly related to the limited number of well-segmented objects in the image bank. Other problems are related to scaling, since occlusions and different poses of the reference animals affect the scaling approach. Figure 6 presents some failure cases of the proposed pipeline.

5. Conclusions

Image generation from scene graphs is a complex topic that merges computer vision and natural language processing solutions. Some advances in generative adversarial networks have assisted in generating better solutions to the task. In this work, we proposed a novel methodology for image generation from scene graphs in the wildlife context based on a semi-parametric approach. Our model uses a graph convolutional network to predict semantic layouts and a cascade refinement network to synthesize the final image.

In this work, we proposed three solutions to improve the quality of the synthetic images: (1) a step to improve the

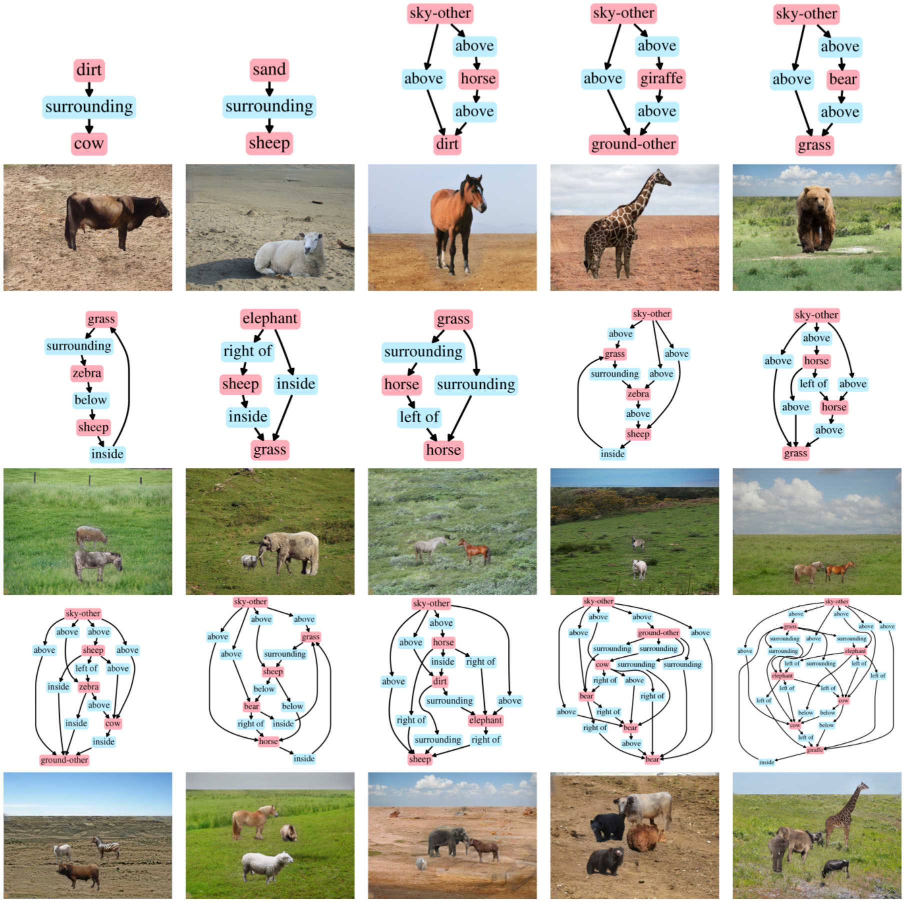


Figure 5. Sample results, scene graphs and their corresponding synthetic image (from top to bottom).

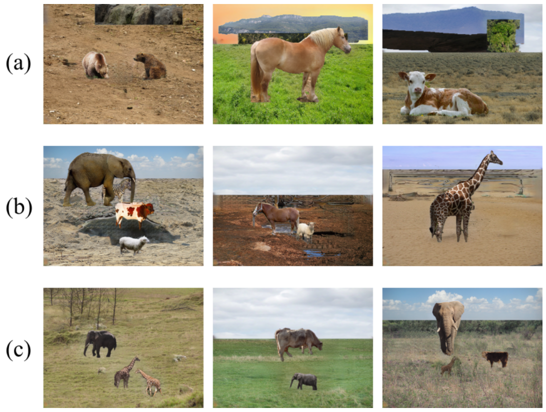


Figure 6. Failure cases: (a) poor quality of the predicted layouts, (b) low quality in the retrieved segments, and (c) scaling affected by occlusions or different poses in the selected terrain.

semantic layout, (2) a scaling approach to resize the animals according to their actual sizes in the real world, and (3) the use of an inpainting network to reduce the missing regions in the image bank segments. We tested our proposed method in the wildlife context with some promising results in simple scenarios with one terrain, sky, and interactions between animals. Additionally, we compared our approach against sg2im and SIMS, achieving outstanding results that outperformed the baseline methods regarding Inception score, Fréchet Inception distance, relation score, and semantic object accuracy.

Although we performed experiments in a selected subset of classes in the wildlife context, the results indicated potential in the sense that we can translate our proposal to other contexts.

For future work, we plan to explore generative adversarial networks to create synthetic landscapes with higher control over the details in the image. Another interesting avenue is to improve the quality of the image bank and include more metadata to match rich text descriptions with the synthesized images.

References

[1] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao,

Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *arXiv preprint arXiv:2105.13290*, 2021. 1, 2

[2] Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, and Andreas Dengel. Adversarial text-to-image synthesis: A review. *arXiv preprint arXiv:2101.09983*, 2021. 1

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017. 5

[5] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 5

[6] Maor Ivgi, Yaniv Benny, Avichai Ben-David, Jonathan Berant, and Lior Wolf. Scene graph to image generation with contextualized object layout refinement. *arXiv preprint arXiv:2009.10939*, 2020. 2, 5

[7] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018. 2, 5

[8] 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. 2

[9] KJ Joseph, Arghya Pal, Sailaja Rajanala, and Vineeth N Balasubramanian. C4synth: Cross-caption cycle-consistent text-to-image synthesis. In *IEEE Winter Conference on Applications of Computer Vision*, pages 358–366. IEEE, 2019. 2

[10] Gal S Kenigsfield and Ran El-Yaniv. Transtextnet: Transducing text for recognizing unseen visual relationships. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1955–1964, 2021. 2

[11] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 3

[12] Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Text-to-image generation grounded by fine-grained user attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 237–246, 2021. 2

[13] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 5

[14] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Controllable text-to-image generation. *arXiv preprint arXiv:1909.07083*, 2019. 1

[15] Yikang Li, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. Pastegan: A semi-parametric method to generate image from scene graph. *Advances in Neural Information Processing Systems*, 32:3948–3958, 2019. 2, 3

[16] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 4

[17] Gaurav Mittal, Shubham Agrawal, Anuva Agarwal, Sushant Mehta, and Tanya Marwah. Interactive image generation using scene graphs. *arXiv preprint arXiv:1905.03743*, 2019. 2

[18] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 4

[19] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8808–8816, 2018. 1, 2, 3, 4, 5

[20] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019. 1

[21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 1, 2

[22] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016. 1

[23] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. 5

[24] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020. 2

[25] Subarna Tripathi, Anahita Bhiwandiwalla, Alexei Bastidas, and Hanlin Tang. Using scene graph context to improve image generation. *arXiv preprint arXiv:1901.03762*, 2019. 2, 5

[26] Duc Minh Vo and Akihiro Sugimoto. Visual-relation conscious image generation from structured-text. In *European Conference on Computer Vision*, pages 290–306. Springer, 2020. 2, 5

[27] Tianren Wang, Teng Zhang, and Brian Lovell. Faces à la carte: Text-to-face generation via attribute disentanglement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3380–3388, 2021. 1

[28] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[29] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017. 3

[30] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018. 1

[31] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020. 4

[32] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. Semantics disentangling for text-to-image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2327–2336, 2019. 1

[33] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1947–1962, 2018. 1

[34] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 4