Object-Centric Image Generation from Layouts

Tristan Sylvain
Mila
Montréal, QC, Canada
tristan.sylvain@gmail.com

Pengchuan Zhang
Microsoft Research AI
Redmond, WA, USA
penzhan@microsoft.com

Yoshua Bengio
Mila
Montréal, QC, Canada
yoshua.bengio@mila.quebec

R Devon Hjelm
Microsoft Research
Redmond, WA, USA
devon.hjelm@microsoft.com

Shikhar Sharma
Microsoft Turing
Montréal, QC, Canada
shikhar.sharma@microsoft.com

Abstract

Despite recent impressive results on single-object and single-domain image generation, the generation of complex scenes with multiple objects remains challenging. We start with the idea that a model must be able to understand individual objects and relationships between objects in order to generate complex scenes well. Our layout-to-image-generation method, which we call Object-Centric Generative Adversarial Network (or OC-GAN), relies on a novel Scene-Graph Similarity Module (SGSM). The SGSM learns representations of the spatial relationships between objects in the scene, which lead to our model's improved layout-fidelity. We also propose changes to the conditioning mechanism of the generator that enhance its object instance-awareness. Apart from improving image quality, our contributions mitigate two failure modes in previous approaches: (1) spurious objects being generated without corresponding bounding boxes in the layout, and (2) overlapping bounding boxes in the layout leading to merged objects in images. Extensive evaluation demonstrates the impact of our contributions, with our model outperforming previous state-of-the-art approaches on both the COCO-Stuff and Visual Genome datasets. Finally, we address an important limitation of evaluation metrics used in previous works by introducing SceneFID – an object-centric adaptation of the popular Fréchet Inception Distance metric, that is better suited for multi-object images.

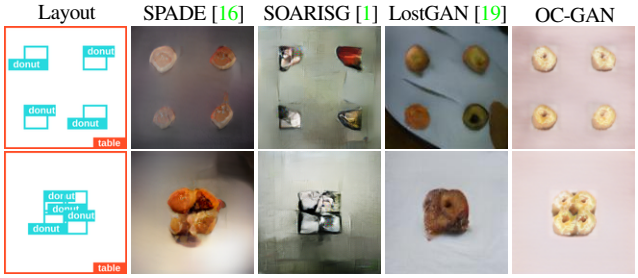

Figure 1. Each row depicts a layout and the corresponding images generated by various models. Along each column, the donuts converge to the centre. In addition to more clearly defined objects, our method OC-GAN is the only one that maintains distinct objects for the final layout, for which bounding boxes slightly overlap.

Figure 2. Existing models introduce spurious objects not specified in the layout, a failure mode our model OC-GAN improves significantly over.

1. Motivation

Generative Adversarial Networks (GANs) [6] have been at the helm of significant recent advances [2] in image generation.While the success in single-domain or single-object focused image generation has been remarkable, generating complex scenes with multiple objects is still challenging. Complex scenes have been generated in the past conditioned on text or even conversations [5, 18]. In this work we will focus on coarse layouts, where the scene to be generated is specified by bounding-box-level annotations.

When considering complex scenes, scene graphs are a potent object-centric representation. By virtue of being a simpler and more distilled abstraction of the scene than a layout, they emphasize *instance awareness* more than layouts that focus on pixel-level class labels, and allow for downstream analysis [20]. In our work, we generate scene graphs depicting positional relationships (such as "to the left of", "above", "inside", *etc.*) from given spatial layouts and leverage them to learn the relationships between objects.

There has been strong interest in image and caption similarity modules for text-to-image generation, most recently with the DAMSM model proposed in [22] resulting in large improvements in performance. Despite this, our approach is the first to use a scene graph to image module when training a generative model.

Despite these advances, models still have difficulty creating realistic scenes. As shown in Figs. 1 and 2, even simple layouts can result in merged objects, spurious modes, and more generally images that do not match the given layout (low layout-fidelity). To counter this, we propose OC-GAN, an architecture to generate realistic images with *high layout-fidelity* and *sharp objects*.

2. Proposed Method

2.1. Scene-Graph Similarity Module

We introduce the Scene Graph Similarity Module (SGSM) as a means of increasing the *layout-fidelity* of generated images.

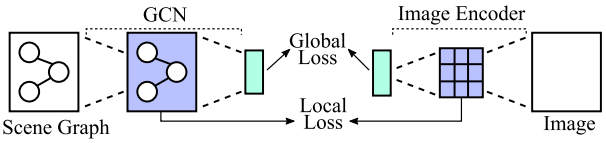

Figure 3. The SGSM module computes similarity between the scene-graph and the generated image and provides fine-grained matching-based supervision between the scene-graph and the generated image.

This multi-modal module, described summarily in Fig. 3, takes as input an image and a scene-graph (nodes corresponding to objects, and edges corresponding to spatial relations). We extract *local visual features* $\boldsymbol{v}_i$ from the *mixed_6e* layer in an Inception-V3 network pre-trained on the Imagenet dataset. We extract *global visual features* $\boldsymbol{v}^G$ from the final pooling layer. We encode the graph using a Graph Convolutional Network (GCN) to obtain *local graph features* $\boldsymbol{g}_j$ and apply a set of graph convolutions followed by a graph pooling operation to obtain *global graph features* $\boldsymbol{g}^G$. Note that each local and global feature is extracted and linearly projected to a common semantic space. In what follows, cos is the cosine similarity, and the γ_ks are normalization constants. We use L/G when the local and global terms are interchangeable. We use the modified dot-product attention mechanism of [22] to compute the *visually attended local graph embeddings* $\tilde{\boldsymbol{g}}_j$:

$$\boldsymbol{s}_{ij} = \gamma_1 \frac{\exp\left(\boldsymbol{g}_j{}^T \boldsymbol{v}_i\right)}{\sum_{i'} \exp\left(\boldsymbol{g}_j{}^T \boldsymbol{v}_{i'}\right)}, \qquad \tilde{\boldsymbol{g}}_j = \frac{\sum_i \exp(\boldsymbol{s}_{ij}) \boldsymbol{v}_i}{\sum_i \exp(\boldsymbol{s}_{ij})} \tag{1}$$

Then we can define a *local similarity metric* between the source graph embedding $\boldsymbol{g}_j$ and the visually aware local embedding $\tilde{\boldsymbol{g}}_j$ analogously to [22]. Intuitively, the similarity will be strong when the source graph embedding is close to the visually aware embedding. This local similarity will encourage different patches of the image to match the objects expected from the scene graph. The *global similarity metric* is classically the cosine distance between embeddings:

$$\begin{cases} \text{Sim}^L(S,I') = \log\left(\sum_j \exp\left(\gamma_2 \cdot \cos(\tilde{\boldsymbol{g}}_j, \boldsymbol{g}_j)\right)\right)^{\frac{1}{\gamma_2}} & (2) \\ \text{Sim}^G(S,I') = \cos\left(\boldsymbol{v}^G, \boldsymbol{g}^G\right) & (3) \end{cases}$$

Finally we can define a global and local probability model in a similar way to e.g. [8]:

$$\mathbb{P}^{L/G}(S,I') \propto \exp\left(\gamma_3 \cdot \text{Sim}^{L/G}(S,I')\right) \tag{4}$$

Normalizing over the images or scenes in the batch B (negative examples are selected by mis-matching the image and scene-graph pairs in the batch) leads to e.g.: $\mathbb{P}^{L/G}(S|I) = \frac{\mathbb{P}^{L/G}(S,I)}{\sum_{I' \in B} \mathbb{P}^{L/G}(S,I')}$. We define the loss terms as the log posterior probability of matching an image I and *the corresponding* scene graph (and vice-versa):

$$\begin{cases} \mathcal{L}_{L/G} = -\log \mathbb{P}_{L/G}(S|I) - \log \mathbb{P}_{L/G}(I|S) & (5) \\ \mathcal{L}_{\text{SGSM}} = \mathcal{L}_L + \mathcal{L}_G & (6) \end{cases}$$

2.2. Instance-Aware Conditioning

As in [16, 19], the parameters γ, β of our batch-normalization layers are *conditional* and determined on a per-pixel level (as opposed to classical conditional batch-normalization [3]). In our case, these parameters are determined by three concatenated inputs: *masked object embeddings, bounding-box layouts and instance boundaries*. Masked object embeddings [14, 19] and bounding-box layouts (using 1-hot embeddings) have been previously used in the layout to image setting. A shortcoming of these conditioning inputs is that they do not provide any way to distinguish between objects of the same class if their bounding boxes overlap. We use the layout's bounding-box boundaries as additional conditioning information. The addition of the instance boundaries helps the model in mapping overlapping conditioning semantic masks to separate object instances, the absence of which led previous state-of-the-art methods to generate merged outputs as shown in the donut example in Fig. 1.

2.3. Architecture

Our OC-GAN model is based on the GAN framework. We present an overview of the model in Fig. 4. We used a classical residual architecture consisting of 5 layers.

We use two different types of discriminators. We discriminate objects using an *object discriminator* D_{obj} taking as input crops of the objects (as identified by their input bounding boxes) in real and fake images resized to size 32×32. It is trained using the Auxiliary-Classifier (AC) [15] framework, resulting in a classification and an adversarial loss. We discriminate whole images using a set of two *patch-wise discriminators* D_1^p, D_2^p. These output estimates of whether a given patch is consistent with the input layout. We apply them to the original image and the same image down-sampled by a factor of 2 (no weight sharing) in a similar fashion to [16, 21]. We also include a perceptual loss [4]. The generator and patch discriminators are trained using the adversarial hinge loss [12].

3. Experiments

3.1. Training Details and Implementation

We ran experiments on both the COCO-Stuff [13] and Visual Genome [11] datasets. These datasets represent complex scenes often featuring more than 1 object. We apply the same

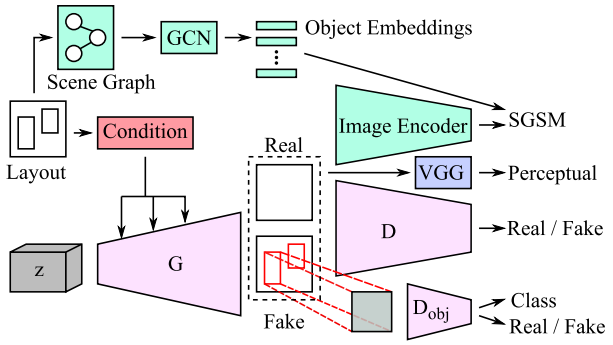

Figure 4. Overview of our OC-GAN model. The GCN and Image Encoder modules are trained separately and then frozen. The condition for the Generator's normalization and the Scene Graph encoding the spatial relationships between objects are both derived from the input layout. The SGSM and the instance-aware normalization lead our model to generate images with higher layout-fidelity and sharper, distinct objects.

pre-processing and use the same splits as [9, 23]. Our OC-GAN model takes as input the spatial layout *i.e.* object bounding boxes and object class annotations.

We use synchronized BatchNorm (all summary statistics are shared across GPUs). We used the Adam [10] solver, with $\beta_1 = 0.5$, $\beta_2 = 0.999$. The global learning rate for both generator and discriminators is 0.0001. Models were trained for up to 300 000 iterations (early stopping on a validation set).

3.2. Baselines

We consider all recent methods that allow layout-to-image generation (Layout2Im [23], LostGAN [19]) or scene-graph-to-image generation (SG2Im [9], SOARISG [1]) as the two fields are closely related, and adapt SPADE [16] and Pix2PixHD [21] to this setting as well. SOARISG cannot be applied to Visual Genome (VG) which has no pixel-level semantic segmentations. LostGAN uses flips when training their model, we report results with and without flips for our method.

3.3. Evaluation and SceneFID

Evaluation of GANs is a complex issue, and the subject of a vast body of literature. In this paper, we report results on three existing evaluation metrics: Inception Score (IS) [17], Fréchet Inception Distance (FID) [7] and Classification Accuracy Score (CAS). For the CAS, we use the setup of [1].

In addition to these metrics, we also introduce the SceneFID metric which corresponds to the FID computed on each object (as identified by bounding boxes), resized to size 224 × 224. This metric has the advantage of (1) using the FID in the setting it was inteded for, single mode distributions corresponding to one object (2) providing a measure of actual object visual quality (completing the analysis provided by the CAS).

3.4. Quantitative Results

We report comparisons of our model's performance to the set of all recent state-of-the-art methods. Where applicable and possible, we use metric values reported by the authors of the papers, and report results using the same experimental setting when training our method.

Table 1 shows that our model consistently outperforms the baselines in terms of IS, FID and CAS, often significantly.

On the proposed SceneFID metric, Table 2 shows that our method outperforms the others significantly. Thus, our model is significantly better at generating realistic objects.

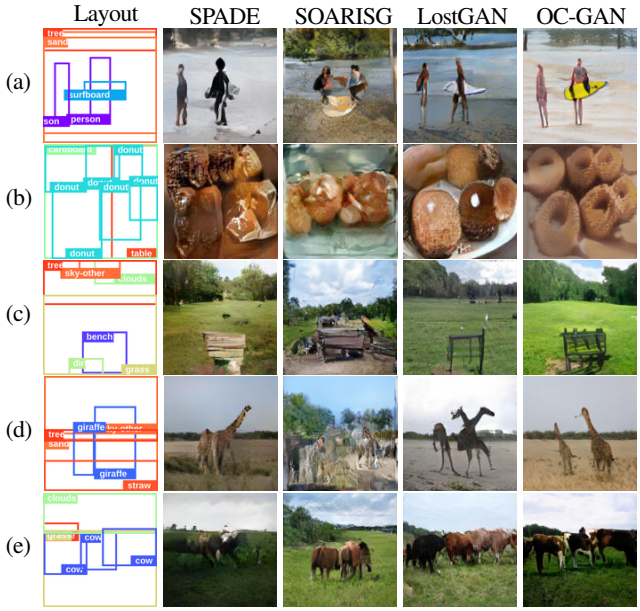

Figure 5. 128 × 128 COCO-Stuff test set images, taken from our method (OC-GAN), and multiple competitive baselines.

Methods	Inception Score ↑		FID ↓		CAS ↑	
	COCO	VG	COCO	VG	COCO	VG
Real Images	22.3 ± 0.5	20.5 ± 1.5	0	0	60.71	56.25
Pix2PixHD [21]	10.4 ± 0.3	9.8 ± 0.3	62.00	46.55	26.67	25.03
SPADE [16]	13.1 ± 0.5	11.3 ± 0.4	40.04	33.29	41.74	34.10
Layout2Im [23]	12.0 ± 0.4	10.1 ± 0.3	43.21	38.21	49.06	51.13
SOARISG [1]	12.5 ± 0.3	N/A	59.5	N/A	44.6	N/A
OC-GAN (ours)	14.0 ± 0.2	11.9 ± 0.5	36.04	28.91	60.32	58.03
LostGAN [19]	13.8 ± 0.4	11.1 ± 0.6	**29.65**	29.36	41.38	28.76
OC-GAN (ours w/ flips)	**14.6±0.4**	**12.3±0.4**	36.31	**28.26**	59.44	59.40

Table 1. Performance on 128 × 128 images.

Methods	SceneFID ↓	
	COCO	VG
Pix2PixHD [21]	42.92	42.98
SPADE [16]	23.44	16.72
Layout2Im [23] ◇	22.76	12.56
SOARISG [1]∗	33.46	N/A
LostGAN [19] (flips)	20.03	13.17
OC-GAN (ours w/ flips)	**16.76**	**9.63**

Table 2. SceneFID scores on object crops resized to size 224 × 224. Note the large improvement in SceneFID for our method.

Dataset	SOARISG	LostGAN	Ours
COCO-Stuff	16.8%	36.8%	**46.4%**
VG	N/R	31.4%	**68.6%**

Table 3. Results of our user study. SOARISG cannot be trained on VG, so is marked N/R, non-rated.

3.5. Qualitative Results

We compare and analyse image samples generated by our method and competitive baselines in Fig. 5. In addition to generating higher quality images, our OC-GAN model does not introduce spurious modes *i.e*. objects not specified in the layout but present in the generated image. This can be attributed to the SGSM module which, by virtue of the retrieval task and the scene-graph being a higher-level abstraction than pixels, aids the model in learning a better mapping from the spatial layout to the generated image. Our model also keeps object instances identifiable even when bounding boxes of objects of the same class overlap slightly or are in close proximity. This can be attributed to the addition of instance-level information and leads to sharper, more realistic objects.

To further validate the previous observations, in Fig. 1, we consider the effect of generating from artificial layouts of gradually converging donuts, to tease out the model's ability to correctly generate separable object instances. Our model generates distinct donuts even when occluded, whereas the other models generate realistic donuts when the bounding boxes are far apart, but fail to do so when they overlap.

We also conducted a user study to evaluate the model's layout-fidelity. The study surveyed 10 users who were each shown 100 layouts each from the COCO-Stuff and VG test sets and corresponding 128×128 images generated by various models. The images were shuffled in a random order. For each layout, the users were asked to select the model which generates the best corresponding image. The results from the user study are presented in Table 3 and demonstrate that our model has higher layout-fidelity than previous state-of-the-art methods.

4. Conclusion

We proposed a novel Scene-Graph Similarity Module that mitigated the layout-fidelity issues in existing models, aided by an improved understanding of spatial relationships derived from the layout. We also proposed to condition the generator's normalization layers on instance boundaries which led to sharper, more distinct objects compared to other approaches. Our model OC-GAN outperforms previous state-of-the-art approaches on a variety of quantitative metrics. Human users also rated our approach higher on generating better-suited images for the layout over existing methods. Our proposed SceneFID metric addresses the concerns around using Inception Score and FID for multi-object images and presents a useful metric for the image generation community which will increasingly deal with multi-class settings in the future.

References

[1] O. Ashual and L. Wolf, "Specifying object attributes and relations in interactive scene generation," in *ICCV*, 2019. 1, 3

[2] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *ICLR*, 2019. 1

[3] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *NeurIPS*, 2017. 2

[4] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *NIPS*, 2016. 2

[5] A. El-Nouby, S. Sharma, H. Schulz, D. Hjelm, L. El Asri, S. Ebrahimi Kahou, Y. Bengio, and G. W. Taylor, "Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction," in *ICCV*, 2019. 1

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014. 1

[7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NIPS*, 2017. 3

[8] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013. 2

[9] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *CVPR*, 2018. 3

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015. 3

[11] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: connecting language and vision using crowdsourced dense image annotations," *IJCV*, 2017. 2

[12] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017. 2

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context"," in *ECCV*, 2014. 2

[14] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool, "Exemplar guided unsupervised image-to-image translation with semantic consistency," *arXiv preprint arXiv:1805.11145*, 2018. 2

[15] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*, 2017. 2

[16] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019. 1, 2, 3

[17] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NIPS*, 2016. 3

[18] S. Sharma, D. Suhubdy, V. Michalski, S. E. Kahou, and Y. Bengio, "ChatPainter: Improving text to image generation using dialogue," in *ICLR Workshop*, 2018. 1

[19] W. Sun and T. Wu, "Image synthesis from reconfigurable layout and style," in *ICCV*, 2019. 1, 2, 3

[20] T. Sylvain, L. Petrini, and D. Hjelm, "Locality and compositionality in zero-shot learning," in *ICLR*, 2020. 1

[21] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *CVPR*, 2018. 2, 3

[22] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *CVPR*, 2018. 2

[23] B. Zhao, L. Meng, W. Yin, and L. Sigal, "Image generation from layout," in *CVPR*, 2019. 3