Object-Centric Image Generation from Layouts

Tristan Sylvain[1,2], Pengchuan Zhang[3], Yoshua Bengio[1,2,4], R Devon Hjelm[3,1], and Shikhar Sharma[3]

[1] Mila, Montréal, Canada
{tristan.sylvain,yoshua.bengio}@mila.quebec
[2] Université de Montréal, Montréal, Canada
[3] Microsoft Research
{penzhan,devon.hjelm,shikhar.sharma}@microsoft.com
[4] CIFAR Senior Fellow

Abstract. Despite recent impressive results on single-object and single-domain image generation, the generation of complex scenes with multiple objects remains challenging. In this paper, 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 quantitative evaluation and ablation studies demonstrate 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.

Keywords: Image generation, Adversarial methods, Scene Graphs

1 Introduction

Generative Adversarial Networks (GANs) [12] have been at the helm of significant recent advances [12,45,15,38,4] in image generation. Apart from unsupervised image generation, GAN-based image generation approaches have done well at conditional image generation from labels [45,63,4], captions [46,64,60,27,62], conversations [50,8,29], scene graphs [23,36,2], layouts [65,53], segmentation masks [41],

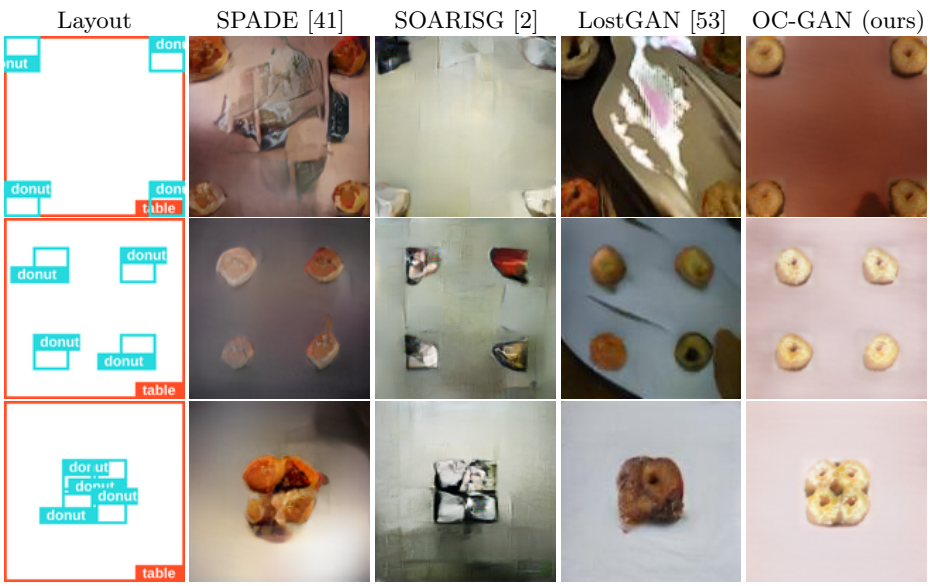


Fig. 1: Each row depicts a layout and the corresponding images generated by various models. Along each column, the donuts converge to the centre. In addition to more clearly defined objects, our method is the only one that maintains distinct objects for the final layout, for which bounding boxes slightly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

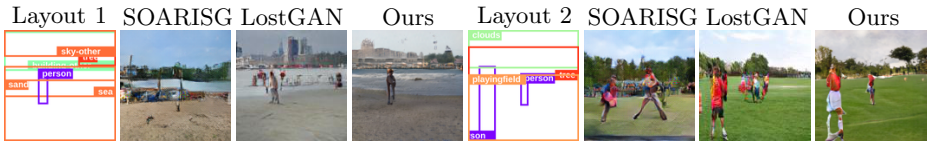


Fig. 2: Existing models introduce spurious objects not specified in the layout, a failure mode over which our model improves significantly.

etc. While the success in single-domain or single-object focused image generation has been remarkable, generating complex scenes with multiple objects is still challenging.

Generating realistic multi-object scenes is a difficult task owing to the large number of components constituting a given image (the Visual Genome [26] dataset as pre-processed by most methods can contain as many as 30 different objects in an image). Past methods focus on different input types, including scene graphs [23,2], pixel-level semantic segmentation [27], and bounding box-level segmentation [65,53]. In addition, some methods also consider multi-modal data, such as instance segmentation alongside pixel-wise semantic segmentation masks [41,58]. Orthogonal to considerations relating to the input, methods tend to rely on additional components to help with the complexity of scene gen-

eration, such as attention mechanisms [60,27] and explicit disentanglement of objects from the background [52].

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*.

Our main contributions can be summarized as follows:

- We introduce a set of novel components that are well-motivated and improve performance for complex scene generation. Our proposed scene-graph-based retrieval module, the SGSM, improves layout-fidelity. We also introduce other improvements, such as conditioning on instance boundaries, that help generating sharp objects and realistic scenes.
- Our model improves significantly on the previous state of the art in terms of a set of classical metrics. In addition to these, we introduce SceneFID, a novel metric that is conceptually better adapted to judge the image quality of complex scenes with many objects, and demonstrate large improvements *vs*. other models.
- Our ablation study demonstrates the usefulness of our different components in terms of a number of standard metrics. This justifies our model choices and provides insights for those who wish to build off our work to generate realistic and complex scenes from layouts.

2 Related Work

Conditional scene generation For some time, the image generation community has focused on complex scenes that contain multiple objects in the foreground [46,64,23]. Several conditional image generation tasks have been formulated using different subsets of annotations. Text-based image generation using captions [46,64,60,27,62] or even multi-turn conversations [50,8,29] have gained significant interest. However, with increasing numbers of objects and their relationships in the image, understanding long textual captions becomes difficult [23,50]. Text-based image generation approaches are also not immune to small perturbations in text leading to quite different images [62].

Layout-based synthesis Generating images from a given layout makes the analysis more interpretable by decoupling the language understanding problem from the image generation task. Another advantage of generating from layouts is more controllable generation: it is easy to design interfaces to manipulate layouts. Owing to these advantages, in this work we will focus on coarse layouts, where the scene to be generated is specified by bounding-box-level annotations. Layout-based approaches fall into 2 broad categories. Some methods take scene-graphs as inputs, and learn to generate layouts as intermediate representations [23,2]. In parallel, other approaches have focused on generating directly from coarse layouts [53,65]. Models that perform well on fine-grained pixel-level

semantic maps also can be easily applied to this setting [41,21,58]. Almost all recent approaches have in common the use of *patch* and *object discriminators* (to ensure whole image and object quality). In addition to this, image quality has been improved by the addition of *perceptual losses* [41,2,58], *multi-scale patch-discriminators* [41], which motivate some of our architecture choices. Finally, significant gains were realized recently by modulating the parameters of batch- or instance-normalization layers [20,57] with a function of the input condition. This is done per-channel in [40], and more recently per pixel [41,53]. As bounding box layouts are coarse for this task, it is common to introduce *unsupervised mask generators* [53,35] to provide estimated shapes for this conditioning.

Finally, there is a growing body of literature involving semi-parametric [43,28] models that use ground-truth training image to aid generation. We consider the case of such models in the Appendix.

Scene-graphs and image matching Scene-graphs have been used traditionally as intermediate representations in image captioning [61,1], reconstruction [14] and retrieval [24], as well as in sentence to scene graph [49] and image to scene graph [33,39] prediction.

When considering complex scenes, scene graphs are a potent object-centric representation, potentially leading to better supervision. Firstly, 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. Secondly, for scenarios that might require generating multiple diverse images, they provide more variability in reconstruction and matching tasks as the mapping from a scene graph to an image is one to many usually. These points explain their use in higher-level visual reasoning tasks such as visual question answering [56] and zero-shot learning [54], and also motivate the use of scene graph-based retrieval in our model. 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, which would be more difficult for a model to distill from pixel-level layouts. The higher layout-fidelity of our model can in a large part be attributed to the SGSM module that uses the scene-graphs.

There has been strong interest in image and caption similarity modules for retrieval [9,18] and for text-to-image generation, most recently with the DAMSM model proposed in [60]. Despite similar interest in scene graph to image retrieval [24,44], and the large improvements in text-to-image synthesis resulting from the DAMSM [60,27], our approach is the first to use a scene graph to image retrieval module when training a generative model.

3 Proposed Method

3.1 Scene-Graph Similarity Module

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

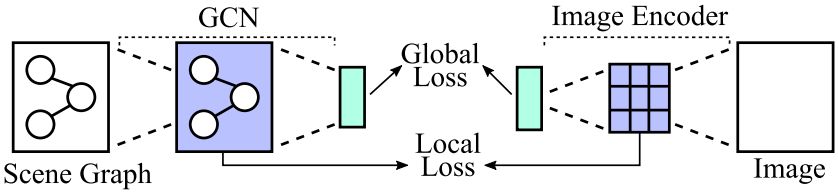


Fig. 3: Overview of the SGSM module. The SGSM module computes similarity between the scene-graph and the generated image and provides fine-grained matching-based supervision between the positional scene-graph and the generated image.

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 [55] 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) [11,13,48] 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 [60] 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 [60]. 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:

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

Finally we can define a global and local probability model in a similar way to e.g. [18]:

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

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):

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

Empirically, the SGSM resulted in large gains in performance as shown in Table 6. Our hypothesis is that the scene graph, in a similar way to a caption, provides easier, simpler to distil relational information contained in the layout, which results in stronger performance compared to generation using just the layout.

Architectural details of the SGSM and related data processing are described in the Appendix.

3.2 Instance-Aware Conditioning

As in [41,53], 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 [6]). In our case, these parameters are determined by three concatenated inputs: *masked object embeddings, bounding-box layouts and instance boundaries*. Masked object embeddings [35,53] 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.

3.3 Architecture

Our OC-GAN model is based on the GAN framework. The generator module generates the images conditioned on the ground-truth layout. The discriminator has the task of predicting whether the input image is generated or real. The discriminator has an additional object discriminator component which has to discriminate objects present in the input image patches corresponding to the ground-truth layout object bounding boxes. We present an overview of the model in Fig. 4 and describe the components below.

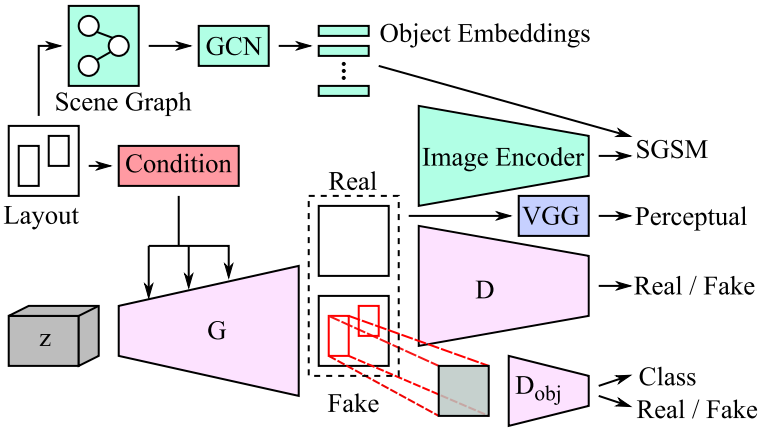


Fig. 4: Overview of our OC-GAN model. The GCN and Image Encoder modules are trained separately and then frozen. The condition for the Generator's normalization and the Scene Graph encoding the spatial relationships between objects are both derived from the input layout. The SGSM and the instance-aware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

Generator As a means of disentangling our model's performance from a specific choice of generator architecture, we used a classical residual [16] architecture consisting of 4 layers for 64×64 inputs, and 5 layers for 128×128 inputs, as used recently in [41,53,58]. The residual decoder G takes as input image-level noise. As described in Sec. 3.2, we further condition the generation by making the normalization parameters of the batch-norm layers of the decoder dependent on the layout and instance boundaries.

Discriminator We use two different types of discriminators, an object discriminator, and a set of patch-wise 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) [40] 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 [41,58].

Further details of the architectures are provided in Section 4.2 and the Appendix.

3.4 Loss Functions

In the following, x denotes a real image, l a layout, and z noise. We also denote objects with o and their labels y_o.

Perceptual loss We found that adding a perceptual loss [7,10,22] to our model improved results slightly. We extract features using a VGG19 network [51]. The loss has expression: $\mathcal{L}_P = \mathbb{E}_{x,l,z} \sum_{i=1}^{N} \frac{1}{D_i} ||F^{(i)}(x) - F^{(i)}(G(l,z)||_1$ where $F^{(i)}$ extracts the output at the i-th layer of the VGG and D_i is the dimension of the flattened output at the i-th layer.

Generator and Discriminator losses We train the generator and patch discriminators using the adversarial hinge loss [30]:

$$\mathcal{L}_G^{\text{GAN}} = -\mathbb{E}_{l,z}\Big[D_1^p(G(l,z),l) + D_1^p(G(l,z),l)\Big] \tag{7}$$

$$\mathcal{L}_{D^p} = \sum_{i=1}^{2} -\mathbb{E}_{x,l}\Big[\min(0,-1+D_i^p(x,l))\Big] - \mathbb{E}_{l,z}\Big[\min(0,-1-D_i^p(G(l,z),l)\Big] \tag{8}$$

The object discriminator follows the AC-GAN framework [40], leading to $\mathcal{L}_G^{AC}$ and $\mathcal{L}_{D_{obj}}^{AC}$. The final expression is:

$$\mathcal{L}_G = \mathcal{L}_G^{\text{GAN}} + \lambda_P \mathcal{L}_{\text{P}} + \lambda_{\text{SGSM}} \mathcal{L}_{\text{SGSM}} + \lambda_{\text{AC}} \mathcal{L}_G^{AC} \tag{9}$$

$$\mathcal{L}_D = \mathcal{L}_{D^p} + \lambda_o \mathcal{L}_{D_{obj}}^{AC} \tag{10}$$

We fix $\lambda_P = 2, \lambda_o = 1, \lambda_{\text{SGSM}} = 1, \lambda_{\text{AC}} = 1$ in our experiments.

4 Experiments

4.1 Datasets

Microsoft Common Objects in Context (MS-COCO) [31] and Visual Genome (VG) [26] datasets have been the popular choice for layout- and scene-to-image tasks as they provide diverse and high-quality annotations. In the case of MS-COCO, the annotation comprises image-captions, object bounding boxes, and instance segmentation masks. In the case of VG, the annotation comprises of object bounding boxes, object attributes, relationships, region descriptions, and segmentation. Building upon MS-COCO, the COCO-Stuff [5] dataset augments the MS-COCO dataset with pixel level *stuff* annotations. In keeping with recent approaches, we ran experiments on both the COCO-Stuff [31] and Visual Genome [26] datasets. These datasets represent complex scenes often featuring more than 1 object. We apply the same pre-processing and use the same splits as [23,65]. The summary statistics of the two datasets are presented in Table 1.

Our OC-GAN model takes three different inputs:

- The spatial layout *i.e.* object bounding boxes and object class annotations.
- Instance boundary maps computed directly from the layout. While they appear redundant once the bounding boxes are provided, they aid the model in better differentiating different objects especially different instances of the same object class.
- Scene-graphs. These are constructed from the objects and spatial relations inferred from the bounding box positions following the setup in [23]. While VG provides more complex scene graphs, we restricted ourselves to spatial relations only for compatibility between the two datasets.

Table 1: Statistics of the COCO-Stuff and Visual Genome dataset.

Dataset	# Train Images	# Valid Images	# Test Images	# Objects	# Objects in Image
COCO-Stuff	24 972	1 024	2 048	171	$3 \sim 8$
VG	62 565	5 506	5 088	178	$3 \sim 30$

4.2 Implementation and Training Details

Our code is written in PyTorch [42]. We apply Spectral Normalization [37] to all the layers in both the generator and discriminator networks. Each experiment ran on 4 V100 GPUs in parallel, using float 16 precision (we verified empirically that float 16 precision had no measurable impact on results compared to float 32). We use synchronized BatchNorm (all summary statistics are shared across GPUs).

We used the Adam [25] solver, with $\beta_1 = 0.5$, $\beta_2 = 0.999$. The global learning rate for both generator and discriminators is 0.0001. 128×128 models were trained for up to 300 000 iterations, 64×64 models were trained for up to 200 000 iterations (early stopping on a validation set).

4.3 Baselines

We consider all recent methods that allow layout-to-image generation (Layout2Im [65], LostGAN [53]) or scene-graph-to-image generation (SG2Im [23], SOARISG [2]) as the two fields are closely related. We also consider SPADE [41] and Pix2PixHD [58] as related baselines. While the latter two were not initially designed for layout-based generation (they generate images based on pixel-level semantic segmentation maps), they can be readily adapted to this new context.

SOARISG uses semantic segmentation maps during training. As a result, it cannot be applied to the VG dataset which does not provide such maps.

In their publicly available implementation, LostGAN [53] uses data augmentation during training – they add flips of the layout. We found that this increased results by a small margin, and for a fair comparison, we provide results when training with and without this data-augmentation for our OC-GAN method.

4.4 Evaluation

Evaluation of GANs is a complex issue, and the subject of a vast body of literature. In this paper, we focus on three existing evaluation metrics: Inception Score (IS) [47], Fréchet Inception Distance (FID) [17] and Classification Accuracy Score (CAS). For the CAS score, a ResNet-101 [16] network is trained on object crops obtained from the real images of the train set of the corresponding dataset, as suggested by [2]. The FID metric computes the 2-Wasserstein distance between the real and generated distributions, and therefore serves as an efficient proxy for the diversity and visual quality of the generated samples. While the FID metric focuses on the whole image, the CAS metric allows us to demonstrate the ability of our model to generate realistic-looking objects within a scene. Finally, we include the Inception Score as a legacy metric.

Our proposed metric: SceneFID We note that there exist many concerns in the literature regarding the use of metrics that are not designed or adapted to the task at hand. The Inception Score has been criticised [3], notably due to issues caused by the mismatch between the domain it was trained on (the ImageNet dataset comprising single objects of interest) and the domain of VG and COCO-Stuff images (comprising multiple objects in complex scenes), making it a potentially poor metric to evaluate generative ability of models in our setting. While the FID metric was introduced in response to Inception Score's criticisms, and was shown empirically to alleviate some of the concerns with it [19,59,34], it stills suffers from problems in the layout-to-image setting. In particular, the single manifold assumption behind FID was found in [32] to be problematic in a multi-class setting. This is *a fortiori* the case in a multi-object setting as in VG and COCO. While [32] introduce a class-aware version of FID, this is not applicable to our setting. We introduce the *SceneFID* metric, where we compute the FID on the crops of all objects, resized to same size (224×224), instead of on the whole image. Thus, the SceneFID metric measures FID in the single manifold assumption it was designed for and extends it to the multi-object setting.

The use of a diverse set of metrics has allowed us to evaluate in more detail the relative advantages of the different models considered. In addition to the above quantitative metrics, we also perform qualitative assessment of the model, notably by considering the effect of modifying the input layout on the output image.

4.5 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. SOARISG [2] depends on semantic segmentation maps being available, and therefore it was not feasible to include results on VG for this method. Some papers introduced additional data-augmentation, such as LostGAN [53] which introduced flips of the real images during training. Where

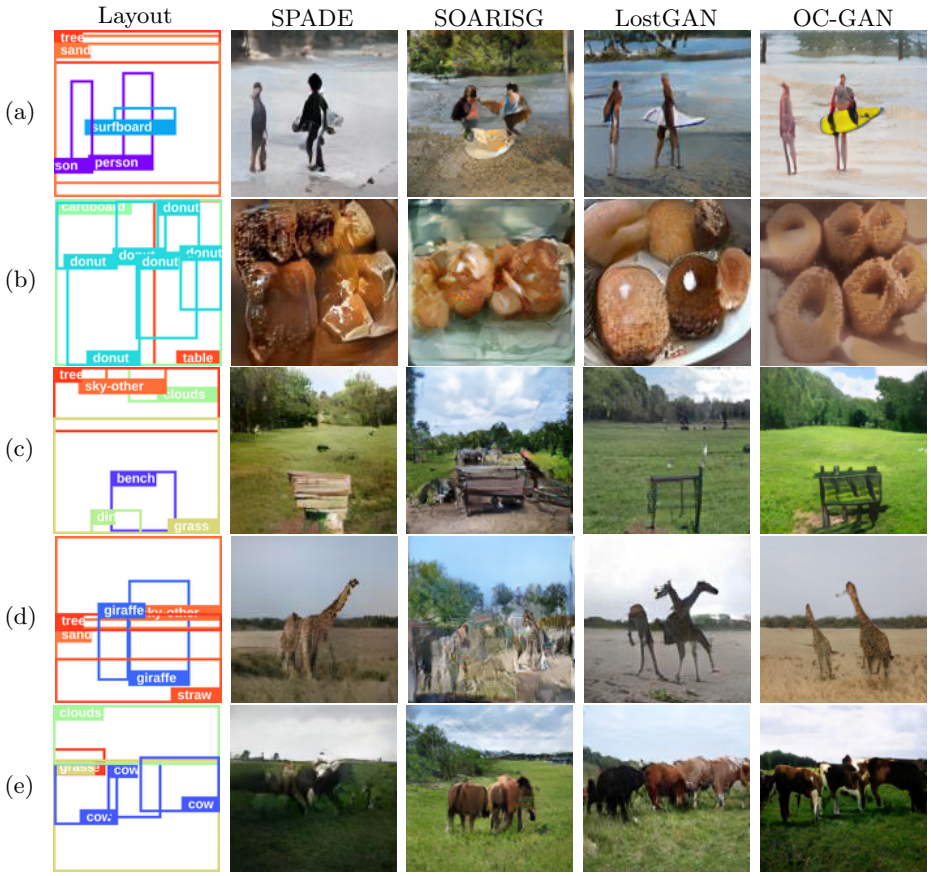


Fig. 5: 128 × 128 COCO-Stuff test set images, taken from our method (OC-GAN), and multiple competitive baselines. Note the overall improved visual quality of our samples. In addition, for (d, e) many baselines introduce spurious objects, and for (b, d, e) spatially close objects are poorly defined and sometimes fused for the baselines.

applicable, we report results using the same experimental setup as the authors, and highlight it in the results table. For all models that do not report CAS scores, we evaluate them using images generated with the pre-trained models provided by their authors.

Tables 2 and 3 show that our model consistently outperforms the baselines in terms of IS, FID and CAS, often significantly. We note that for some models, the CAS score is above that reported for ground-truth images. This is due to the fact that a sufficiently capable generator will start to generate objects that are both realistic, and of the same distribution as the training distribution, rather than the test one.

Table 2: Performance on 64×64 images. All models use ground-truth layouts. We use † to denote results taken from the original paper. $*$ denotes a model that uses pixel-level semantic segmentation during training. N/A denotes a result that cannot be computed (SOARISG cannot be trained on VG due to the absence of pixel-level semantic segmentation). The best results in each category are in bold. Our method outperforms the baselines across the evaluation metrics considered.

	Inception Score ↑		FID ↓		CAS ↑	
Methods	COCO	VG	COCO	VG	COCO	VG
Real Images	16.3 ± 0.4	13.9 ± 0.5	0	0	54.48	49.57
SG2Im [23]†	7.3 ± 0.1	6.3 ± 0.2	67.96	74.61	30.04	40.29
Pix2PixHD [58]	7.2 ± 0.2	6.6 ± 0.3	59.95	47.71	20.82	16.98
SPADE [41]	8.5 ± 0.3	7.3 ± 0.1	43.31	35.74	31.61	23.81
Layout2Im [65]†	9.1 ± 0.1	8.1 ± 0.1	38.14	31.25	50.84	48.09
SOARISG [2]∗ †	10.3 ± 0.1	N/A	48.7	N/A	46.1	N/A
OC-GAN (ours)	10.5 ± 0.3	8.9 ± 0.3	33.10	22.61	56.88	57.73
LostGAN [53] (flips) †	9.8 ± 0.2	8.7 ± 0.4	34.31	34.75	37.15	27.10
OC-GAN (ours w/ flips)	**10.8 ± 0.5**	**9.3 ± 0.2**	**29.57**	**20.27**	**60.39**	**60.79**

Table 3: Performance on 128×128 images. All models use ground-truth layouts. We use † to denote results taken from the original paper. $*$ denotes a model that uses pixel-level semantic segmentation during training. N/A denotes a result that cannot be computed (SOARISG cannot be trained on VG due to the absence of pixel-level semantic segmentation). $\diamond$ denotes models for which the openly available source code was not adapted to 128×128 generation. We altered the code to allow this and ran a hyperparameter search on the new models. The best results in each category are in bold. Our method outperforms the baselines across most evaluation metrics considered.

	Inception Score ↑		FID ↓		CAS ↑	
Methods	COCO	VG	COCO	VG	COCO	VG
Real Images	22.3 ± 0.5	20.5 ± 1.5	0	0	60.71	56.25
Pix2PixHD [58]	10.4 ± 0.3	9.8 ± 0.3	62.00	46.55	26.67	25.03
SPADE [41]	13.1 ± 0.5	11.3 ± 0.4	40.04	33.29	41.74	34.10
Layout2Im [65] $\diamond$	12.0 ± 0.4	10.1 ± 0.3	43.21	38.21	49.06	51.13
SOARISG [2] †∗	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 [53] †	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

On the proposed SceneFID metric, Table 4 shows that our method outperforms the others significantly. Thus, our model is significantly better at generating realistic objects compared to the baselines.

Note that the LostGAN model obtains better FID compared to our model exceptionally on 128×128 COCO-Stuff images but our OC-GAN model outperforms it on the SceneFID metric which is more appropriate in this multi-class setting.

Table 4: SceneFID scores on object crops resized to size 224×224, extracted from the 128×128 outputs of the different models, for both datasets. All models use ground-truth layouts. $*$ denotes a model that uses pixel-level semantic segmentation during training. N/A denotes a result that cannot be computed (SOARISG cannot be trained on VG due to the absence of pixel-level semantic segmentations). $\diamond$ denotes models for which the openly available source code was not adapted to 128×128 generation. Note the large improvement in SceneFID for our method.

	SceneFID ↓	
Methods	COCO	VG
Pix2PixHD [58]	42.92	42.98
SPADE [41]	23.44	16.72
Layout2Im [65] $\diamond$	22.76	12.56
SOARISG [2]$*$	33.46	N/A
LostGAN [53] (flips)	20.03	13.17
OC-GAN (ours w/ flips)	**16.76**	**9.63**

4.6 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 from the COCO-Stuff test set and corresponding 128×128 images generated by the SOARISG, LostGAN, and our OC-GAN models. They were also shown 100 layouts from the VG test set and corresponding 128×128 images generated by LostGAN and our OC-GAN 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 5 and demonstrate that our model has higher layout-fidelity than previous state-of-the-art methods.

Table 5: Results of our user study. 10 computer-science professionals were shown 100 COCO-Stuff and 100 VG test set layouts and corresponding images generated by various models, shuffled randomly. Users were asked to select the highest layout-fidelity image for each layout at 128×128 resolution. SOARISG cannot be trained on VG, so is marked N/R, non-rated. Our method is consistently found to have the highest layout-fidelity.

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

Table 6: Quantitative comparison of different ablated versions of our model on the COCO-Stuff dataset (64×64 images). These results highlight the importance of the SGSM (and its positive interaction with the perceptual loss) in the bottom row block, as well as the impact of removing some of the discriminators (middle row block).

	FID ↓	CAS ↑
Full	**29.57**	60.27
Single patchD	30.54	59.86
No patchD	33.85	**62.48**
No objectD	31.62	48.03
No SGSM	34.32	52.57
No objectD, no SGSM	33.15	41.50
No perceptual loss	31.14	57.22
No perceptual loss, no SGSM	36.54	47.94

4.7 Ablation Study

In Table 6, we present an ablation study performed by removing certain components of our model. The effect of adding another patch discriminator is measurable, both in terms of FID and CAS. Removing the patch discriminator significantly lowers FID (the model has no more supervision in terms of matching the distribution of the real full images. This actually improves the CAS, as the generator will use more capacity to focus on generating realistic objects.

We also find that removing either the object discriminator or the SGSM results in a significant drop in performance. This does not however prevent the model from generating realistic objects (the CAS score remains above some of the baselines), meaning that the roles of the two components are to some extent complementary. As soon as both are removed, the CAS score drops sharply.

Removing the perceptual loss has little effect in itself, but it greatly helps the SGSM when present. Removing the SGSM altogether strongly impairs results, highlighting its importance.

5 Conclusion

We observed that current state-of-the-art layout-to-image generation methods exhibit low layout-fidelity and tend to generate low quality objects especially in cases of occlusion. We proposed a novel Scene-Graph Similarity Module that mitigated the layout-fidelity issues 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. The addition of the proposed components to the image generation pipeline led to our model outperforming previous state-of-the-art approaches on a variety of quantitative metrics. A comprehensive ablation study was performed to analyse the contribution of the proposed and existing components of the model. Human users also rated our approach higher on generating better-suited images for the layout over existing methods.

Evaluation metrics for GAN popularized in the single-object-class setting have been criticized as inappropriate in the multi-class setting in literature. Our proposed SceneFID metric addresses those concerns and presents a useful metric for the image generation community which will increasingly deal with multi-class settings in the future. Our proposed OC-GAN model also showed a large improvement over existing approaches on the SceneFID evaluation criteria which further highlights the impact of our contributions.

Acknowledgements

We acknowledge Emery Fine, Adam Ferguson, Hannes Schulz for their insightful suggestions and valuable assistance. We also thank the many researchers who contributed to the human evaluation study.

References

1. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Spice: Semantic propositional image caption evaluation. In: European Conference on Computer Vision. pp. 382–398 (2016)
2. Ashual, O., Wolf, L.: Specifying object attributes and relations in interactive scene generation. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
3. Barratt, S., Sharma, R.: A note on the inception score. arXiv preprint arXiv:1801.01973 (2018)
4. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019)
5. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
6. De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., Courville, A.C.: Modulating early visual processing by language. In: Advances in Neural Information Processing Systems. pp. 6594–6604 (2017)
7. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Advances in neural information processing systems. pp. 658–666 (2016)
8. El-Nouby, A., Sharma, S., Schulz, H., Hjelm, D., El Asri, L., Ebrahimi Kahou, S., Bengio, Y., Taylor, G.W.: Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
9. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., et al.: From captions to visual concepts and back. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1473–1482 (2015)
10. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
11. Goller, C., Kuchler, A.: Learning task-dependent distributed representations by backpropagation through structure. In: Proceedings of International Conference on Neural Networks (ICNN'96). vol. 1, pp. 347–352 (1996)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems 27, pp. 2672–2680 (2014)
13. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. vol. 2, pp. 729–734 (2005)
14. Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., Ling, M.: Scene graph generation with external knowledge and image reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1969–1978 (2019)
15. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems 30, pp. 5767–5777 (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems. pp. 6626–6637 (2017)
18. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2333–2338 (2013)
19. Im, D.J., Ma, A.H., Taylor, G.W., Branson, K.: Quantitatively evaluating GANs with divergences proposed for training. In: International Conference on Learning Representations (2018)
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
21. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
22. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision. pp. 694–711 (2016)
23. Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
24. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 3668–3678 (2015)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
26. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision (2017)
27. Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., Gao, J.: Object-driven text-to-image synthesis via adversarial training. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
28. Li, Y., Ma, T., Bai, Y., Duan, N., Wei, S., Wang, X.: Pastegan: A semi-parametric method to generate image from scene graph. In: Advances in Neural Information Processing Systems 32, pp. 3948–3958 (2019)
29. Li, Y., Gan, Z., Shen, Y., Liu, J., Cheng, Y., Wu, Y., Carin, L., Carlson, D., Gao, J.: Storygan: A sequential conditional gan for story visualization. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
30. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
31. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context". In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) The European Conference on Computer Vision (ECCV). pp. 740–755 (2014)
32. Liu, S., Wei, Y., Lu, J., Zhou, J.: An improved evaluation framework for generative adversarial networks. arXiv preprint arXiv:1803.07474 (2018)
33. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: European Conference on Computer Vision. pp. 852–869. Springer (2016)
34. Lucic, M., Kurach, K., Michalski, M., Gelly, S., Bousquet, O.: Are gans created equal? a large-scale study. In: Advances in Neural Information Processing Systems. pp. 700–709 (2018)

35. Ma, L., Jia, X., Georgoulis, S., Tuytelaars, T., Van Gool, L.: Exemplar guided unsupervised image-to-image translation with semantic consistency. arXiv preprint arXiv:1805.11145 (2018)
36. Mittal, G., Agrawal, S., Agarwal, A., Mehta, S., Marwah, T.: Interactive image generation using scene graphs. In: International Conference on Learning Representations (ICLR): Deep Generative Models for Highly Structured Data Workshop (2019)
37. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)
38. Miyato, T., Koyama, M.: cGANs with projection discriminator. In: International Conference on Learning Representations (2018)
39. Newell, A., Deng, J.: Pixels to graphs by associative embedding. In: Advances in Neural Information Processing Systems. pp. 2171–2180 (2017)
40. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 2642–2651 (2017)
41. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
42. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035 (2019)
43. Qi, X., Chen, Q., Jia, J., Koltun, V.: Semi-parametric image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8808–8816 (2018)
44. Quinn, M.H., Conser, E., Witte, J.M., Mitchell, M.: Semantic image retrieval via active grounding of visual situations. In: 2018 IEEE 12th International Conference on Semantic Computing (ICSC). pp. 172–179 (2018)
45. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (2016)
46. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1060–1069 (2016)
47. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. pp. 2234–2242 (2016)
48. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2008)
49. Schuster, S., Krishna, R., Chang, A., Fei-Fei, L., Manning, C.D.: Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In: Proceedings of the fourth workshop on vision and language. pp. 70–80 (2015)
50. Sharma, S., Suhubdy, D., Michalski, V., Kahou, S.E., Bengio, Y.: ChatPainter: Improving text to image generation using dialogue. In: International Conference on Learning Representations (ICLR) Workshop (2018)
51. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)

52. Singh, K.K., Ojha, U., Lee, Y.J.: Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
53. Sun, W., Wu, T.: Image synthesis from reconfigurable layout and style. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
54. Sylvain, T., Petrini, L., Hjelm, D.: Locality and compositionality in zero-shot learning. In: International Conference on Learning Representations (2020)
55. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2818–2826 (2016)
56. Teney, D., Liu, L., van Den Hengel, A.: Graph-structured representations for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2017)
57. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
58. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
59. Xu, Q., Huang, G., Yuan, Y., Guo, C., Sun, Y., Wu, F., Weinberger, K.: An empirical study on evaluation metrics of generative adversarial networks. arXiv preprint arXiv:1806.07755 (2018)
60. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
61. Yang, X., Tang, K., Zhang, H., Cai, J.: Auto-encoding scene graphs for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10685–10694 (2019)
62. Yin, G., Liu, B., Sheng, L., Yu, N., Wang, X., Shao, J.: Semantics disentangling for text-to-image generation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
63. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 7354–7363 (2019)
64. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
65. Zhao, B., Meng, L., Yin, W., Sigal, L.: Image generation from layout. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

A Comparison with Semi-Parametric Methods

Recently, semi-parametric methods have been proposed in the field of layout-to-image generation [28]. We excluded a comparison with these methods in the main paper due to the fact that (1) they are structurally different (they incorporate real images when generating images) leading to difficulties in making a fair comparison and (2) they function in diverse ways, not all of which can be applied to our setting [43].

We include a comparison with the state-of-the art semi-parametric model, PasteGAN [28] in Table 7. This method outperforms most of the other baselines, but still performs worse than our method.

Table 7: Comparison of our method with the semi-parametric method PasteGAN [28]. We use † to denote results taken from the original paper. The best results in each category are in bold. Our method outperforms this baseline across the evaluation metrics considered.

	Inception Score ↑		FID ↓	
Methods	COCO	VG	COCO	VG
PasteGAN [28] †	10.2 ± 0.2	8.2 ± 0.2	38.29	35.25
OC-GAN (ours)	$\mathbf{10.5 \pm 0.3}$	$\mathbf{8.9 \pm 0.3}$	**33.10**	**22.61**

B Spatial Relationships used for Generating the Scene-Graph

We used 6 spatial relationships to generate the scene-graphs from layouts. All of the spatial relationships are derived from the bounding box coordinates specified in the layouts. If an edge in the scene-graph is represented as <subject, relationship, object>, then the possible relationships we consider are:

- "left of": subject's centre is to the left of object's centre
- "right of": subject's centre is to the right of object's centre
- "above": subject's centre is above object's centre
- "below": subject's centre is below object's centre
- "inside": subject contained inside object
- "surrounding": object contained inside subject

C A Note on Evaluation

Inception Score and FID were computed using the official Tensorflow implementations [5][6] (the most commonly available PyTorch implementations give slightly

[5] `https://github.com/openai/improved-gan` for Inception Score
[6] `https://github.com/bioinf-jku/TTUR` for FID

different but close values), to ensure compliance with the literature. In the past, papers considering layout and scene graph to image generation have used different values for the number of splits when computing the Inception score, ranging usually from 3 to 5 (as shown in the different official implementations and via contacting some of the authors). Empirically, we found that lowering the split size results in better numerical values for the inception score, for all methods relevant to this work. Out of fairness considerations, we opted for splits of size 5 and note that in addition to this issue, the size of the evaluation set for Inception score computation is very low compared to recommended sizes. This impacts the relevance of this metric.

In addition to the above concerns, some models used different network architectures to compute the inception score (e.g. [65] uses a VGG net as opposed to the standard Inception-V3 network as noted in their paper). We used the official Inception-V3-based evaluation on all models.

Some models introduce non-standard data-augmentation (e.g. [53] uses image flips during training). Out of fairness considerations, we compared our approach to the official reported values, and used the same data-augmentation as the compared methods, when applicable.

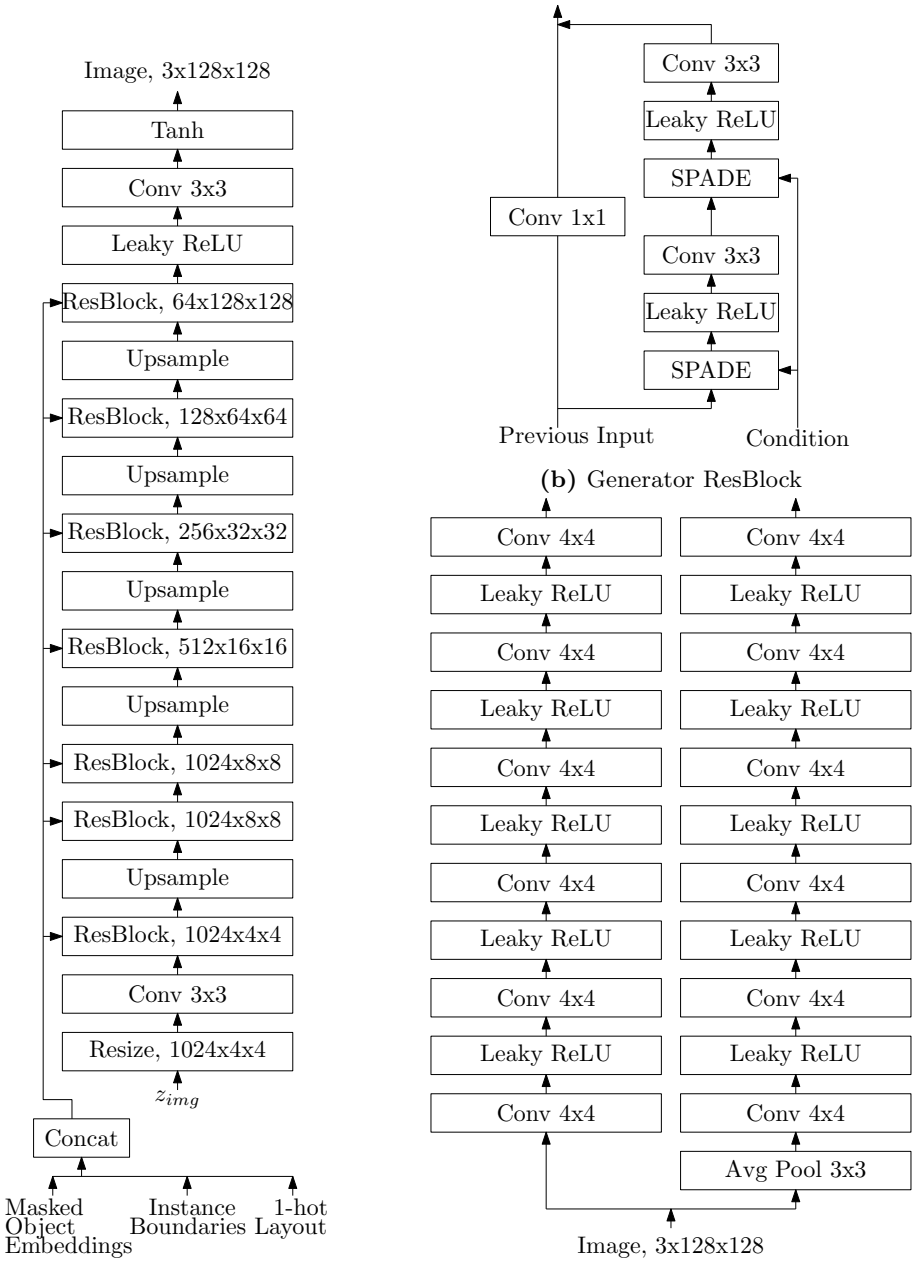
D Implementation and Training Details

Architecture diagrams for all the modules of our model OC-GAN are presented in Figs. 6 and 7. Some additional hyper-parameter details:

- In the SGSM module, images are resized to size 299×299 before being processed by the image encoder.
- In the SGSM module, the common semantic space for graph and image embeddings has a dimension of 256.

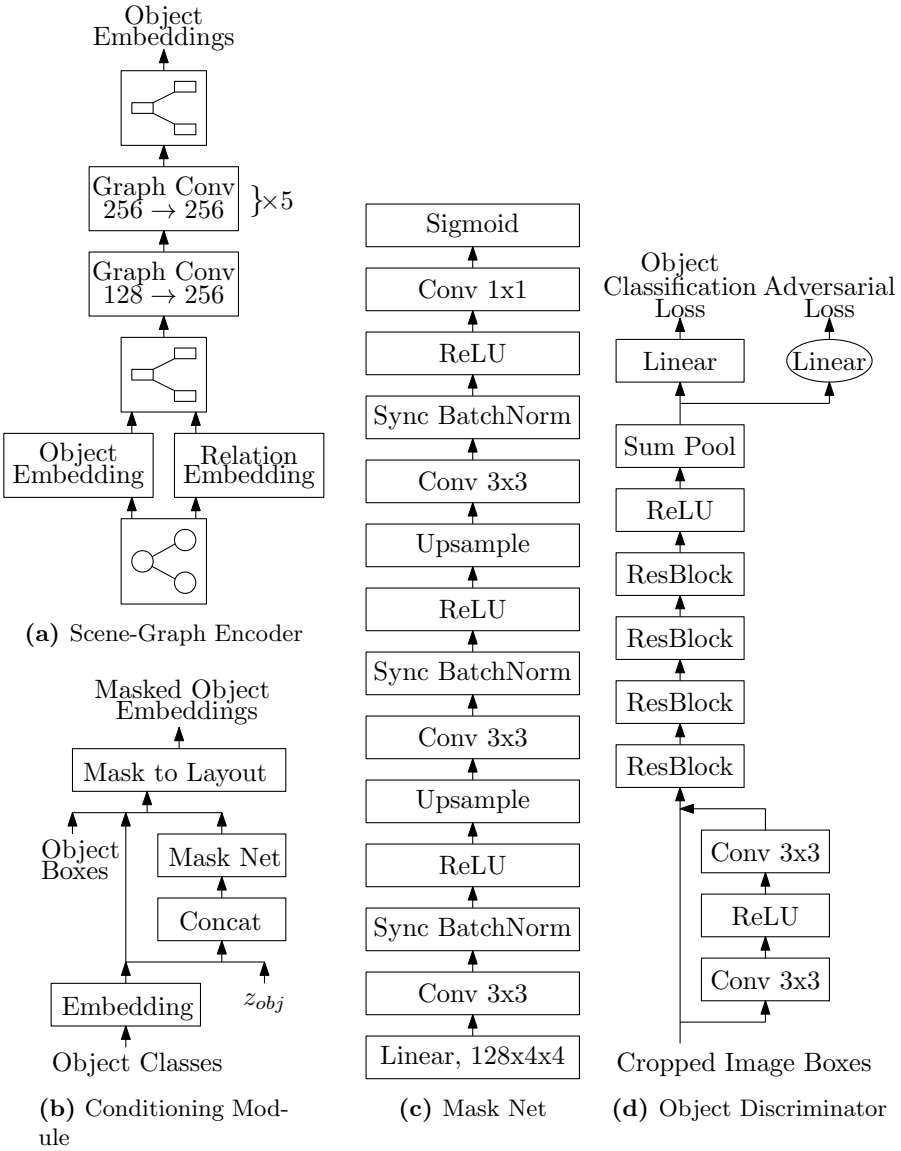
E Additional Qualitative Results

We present additional qualitative 128×128 samples on the COCO-Stuff dataset in Fig. 8 and on the Visual Genome dataset in Fig. 9.



(a) Generator **(b)** Generator ResBlock **(c)** Image Discriminator

Fig. 6: Architecture diagrams for (a) Generator (b) Generator ResBlocks (c) Image Discriminator. All generator inputs are derived from the layout. The Masked Object Embeddings are produced by the Conditioning Module. If input and output dimensions match for the Generator ResBlock, then the shortcut is a skip connection.



(a) Scene-Graph Encoder

(b) Conditioning Module

(c) Mask Net

(d) Object Discriminator

Fig. 7: Architecture diagrams for (a) Scene-Graph Encoder (b) Conditioning Module (c) Mask Net (d) Object Discriminator. The Scene-Graph Encoder takes as input a scene-graph derived from the layout and processes it with a Graph Convolutional Network. The Conditioning Module generates the Masked Object Embeddings, which along with instance boundaries and 1-hot layout, are the conditioning information for the Generator. The Mask Net is a submodule of the Conditioning Module. The Object Discriminator operates on cropped image boxes in an AC-GAN framework, predicting whether the crop is real or generated as well as classifying the object inside the crop.

Fig. 8: 128×128 COCO-Stuff test set images, taken from our method (OC-GAN) and multiple competitive baselines.

Fig. 9: 128 × 128 Visual Genome test set images, taken from our method (OC-GAN) and the LostGAN baseline.