Context-Aware Layout to Image Generation with Enhanced Object Appearance

Sen He[1,2*], Wentong Liao[3*], Michael Ying Yang[4], Yongxin Yang[1,2], Yi-Zhe Song[1,2]
Bodo Rosenhahn[3], Tao Xiang[1,2]
[1]CVSSP, University of Surrey, [2]iFlyTek-Surrey Joint Research Centre on Artificial Intelligence
[3]TNT, Leibniz University Hannover, [4]SUG, University of Twente

Abstract

A layout to image (L2I) generation model aims to generate a complicated image containing multiple objects (things) against natural background (stuff), conditioned on a given layout. Built upon the recent advances in generative adversarial networks (GANs), existing L2I models have made great progress. However, a close inspection of their generated images reveals two major limitations: (1) the object-to-object as well as object-to-stuff relations are often broken and (2) each object's appearance is typically distorted lacking the key defining characteristics associated with the object class. We argue that these are caused by the lack of context-aware object and stuff feature encoding in their generators, and location-sensitive appearance representation in their discriminators. To address these limitations, two new modules are proposed in this work. First, a context-aware feature transformation module is introduced in the generator to ensure that the generated feature encoding of either object or stuff is aware of other co-existing objects/stuff in the scene. Second, instead of feeding location-insensitive image features to the discriminator, we use the Gram matrix computed from the feature maps of the generated object images to preserve location-sensitive information, resulting in much enhanced object appearance. Extensive experiments show that the proposed method achieves state-of-the-art performance on the COCO-Thing-Stuff and Visual Genome benchmarks. Code available at: https://github.com/wtliao/layout2img.

1. Introduction

Recent advances in generative adversarial networks (GANs) [11] have made it possible to generate photo-realistic images for a single object, e.g., faces, cars, cats [4, 46, 20, 21]. However, generating complicated images containing multiple objects (things) of different classes against natural backgrounds (stuff) still remains a challenge [18, 3, 31, 30]. This is due to the large appearance variations for objects of different classes, as well as the complicated relations between both object-to-object and object-to-stuff. A generated object needs to be not only realistic on its own, but in harmony with surrounding objects and stuff.

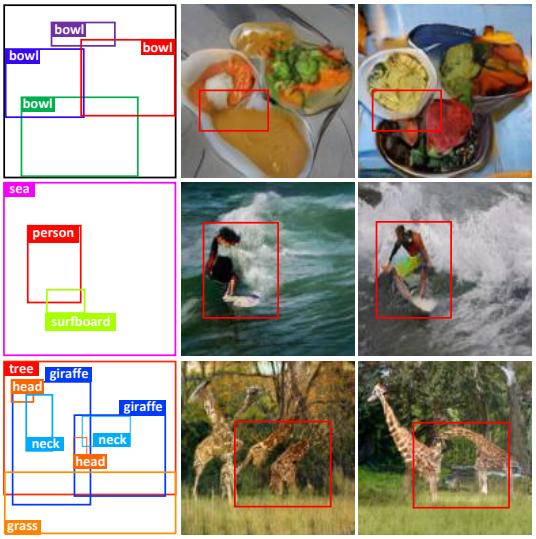

Figure 1. Illustration of the limitations of existing L2I models and how our model overcome them. From left to right: ground truth layout, images generated by the state-of-the-art LostGAN-v2 [39], and by our model with the layout as input. In the middle and right column, regions with key differences in the generation quality between LostGAN-v2 and our model are highlighted in dashed boxes. See text for more details.

Without any conditional input, the *mode collapse* [36, 6] problem is likely to be acute for GANs trained to generate such complicated natural scenes. Consequently, various inputs have been introduced to provide some constraints on the image generation process. These include textual description of image content [31], scene graph representing objects and their relationship [18], and semantic map providing pixel-level annotation [30]. This work focuses on the conditional image generation task using the layout [48, 38, 40] that defines a set of bounding boxes with specified size, location and categories (see Fig. 1). Layout is a user-friendly input format on its own and can also be used as

*Equal contribution

an intermediate input step of other tasks, e.g., scene graph and text to image generation [3, 15].

Since the seminal work [48] in 2019, the very recent layout to image (L2I) generation models [49, 40, 39] have made great progresses, thanks largely to the advances made in GANs [30, 20] as they are the key building blocks. From a distance, the generated images appear to be realistic and adhere to the input layout (see Fig. 1 and more in Fig. 3). However, a closer inspection reveals two major limitations. First, the relations between objects and object-to-stuff are often broken. This is evident from the food example in Fig. 1 (Top-Middle) – the input layout clearly indicates that the four bowls are overlapping with each other. Using the state-of-the-art LostGAN-v2 [39], the occluded regions between objects are poorly generated. Second, each generated object's appearance is typically distorted lacking class-defining characteristics. For instance, the surfing example in Fig. 1 (Middle) and the giraffe example in Fig. 1 (Bottom) show that the object appearance has as if been touched by Picasso – one can still recognize the surfing person or giraffe, but key body parts are clearly misplaced.

We believe these limitations are caused by two major design flaws in existing L2I models in both their GAN generators and discriminators. (1) *Lack of context-aware modeling in the generator*: Existing models generate the feature for the object/stuff in each layout bounding box first, and then feed the generated feature into a generator for image generation. However, the feature generation process for each object/stuff is completely independent of each other, therefore offering no chance for capturing the inter-object and object-to-stuff relations. (2) *Lack of location-sensitive appearance representation in the discriminator*: As in any GAN model, existing L2I models deploy a discriminator that is trained to distinguish the generated whole image and individual object/stuff images from the real ones. Such a discriminator is essentially a CNN binary classifier whereby globally pooled features extracted from the CNN are fed to a real-fake classifier. The discriminator thus cares only about the presence/absence and strength of each semantic feature, rather than where they appear in the generated images. This lack of location-sensitive appearance representation thus contributes to the out-of-place object part problem in Fig. 1 (Middle).

In this paper, we provide solutions to overcome both limitations. First, to address the lack of context-aware modeling problem, we propose to introduce a context-aware feature transformation module in the generator of a L2I model. This module updates the generated feature for each object and stuff after each has examined its relations with all other objects/stuff co-existing in the image through self-attention. Second, instead of feeding location-insensitive globally pooled object image features to the discriminator, we use the Gram matrix computed from the feature maps of the generated object images. The feature map Gram matrix captures the inter-feature correlations over the vectorized feature map, and is therefore locations sensitive. Adding it to the input of the real-fake classifier in the discriminator, the generated images preserve both shape and texture characteristics of each object class, resulting in much enhanced object appearance (see Fig. 1 (Right)).

The contributions of this work are as follows: (1) For the first time, we identify two major limitations of existing L2I models for generating complicated multi-object images. (2) Two novel components, namely a context-aware feature transformation module and a location-sensitive object appearance representation are introduced to address these two limitations. (3) The proposed modules can be easily integrated into any existing L2I generation models and improve them significantly. (4) Extensive experiments on both the COCO-Thing-Stuff [25, 5] and Visual Genome [22] datasets show that state-of-the-art performance is achieved using our model. The code and trained models will be released soon.

2. Related Work

Generative Adversarial Networks Generative adversarial networks (GANs) [11], which play a min-max game between a generator and a discriminator, is the mainstream approach used in recent image generation works. However, the training of a GAN is often unstable and known to be prone to the *mode collapse* problem. To address this, techniques like Wasserstein GAN [2] and Unrolled GAN [28] were developed. Meanwhile, noise injection and weight penalizing [1, 34] were used in the discriminator to alleviate the non-convergence problem for further stabilization of the training. To generate high fidelity and resolution images, architectures like Progressive GAN [19] and BigGAN [4] were also proposed.

Conditional Image Generation Conditional image generation, which generates an image based on a given condition (e.g., class label, sentence description, image, semantic mask, sketch, and scene graph) has been studied intensively [29, 31, 17, 51, 30, 7, 3, 9] due to its potential in generating complicated natural images. In general, there are two popular architectures for the conditional image generation. The first one is the encoder-decoder architecture used in *Pix2pix* [17] and *CycleGAN* [51], where the encoder directly takes the conditional input and embeds it to a latent space. The decoder then transfers the embedded representation into the target image. The second popular architecture is the decoder-only architecture used in *StyleGAN* [20] and *GauGAN* [30], where a decoder starts with a random input, and then progressively transforms it to produce the desired output. In this architecture, the conditional input is used to generate part of the parameters in the decoder, e.g., the

affine transformation parameters in the normalization layers [20, 30, 33] or the weight parameters in convolutional kernels [26].

Layout to Image Generation Though the previous work [15] has already touched the concept of layout to image generation (L2I), it is just used as an intermediate step for a different generation task. The first stand-alone solution appeared in [48]. Compared to other conditional inputs such as text and scene graph, layout is a more flexible and richer format. Therefore, more studies followed up by introducing more powerful generator architectures [38, 39], or new settings [23, 27]. Sun *et al*. [38] proposed a new architecture inspired by *StyleGAN* [20], which allows their model to generate higher resolution images with better quality. Li *et al*. [23] introduced a new setting for high resolution street scene generation. Their model retrieves a background from a database based on the given foreground layout. Recently, Ma *et al*. [27] introduced attribute guided layout generation, which is more controllable on the generated objects. As mentioned earlier, all these existing models have two limitations, namely lack of context-aware modeling in their generators, and lack of location-sensitive appearance representation in their discriminators. Both limitations are overcome in this work, resulting in much improved L2I generation performance (see Sec. 5).

Context Modeling Context plays an important role in many discriminative scene analysis tasks [41, 16, 8, 45, 44, 13, 43]. The main idea in context-based analysis is to tie each object instance in the scene with the global context, such that their relationship or interaction can be better understood. However, context has drawn little attention in image generation. One exception is SAGAN [46] which applied self-attention to refine the feature map in the generator for single object image generation. In this work, we introduce context modeling for layout to image generation, a more complicated image generation task with a focus on inter-objects and object-to-stuff relation modeling.

Appearance Representation in CNNs Works on CNN visualization clear show that feature channels, especially those at the top layers of a CNN capture semantically meaningful concepts such as body parts; and the activations of these feature channel at different locations indicate where these concepts are [50]. However, when it comes to object recognition [35] or real-fake discriminator in GAN [11], these feature maps are globally pooled before being fed into a binary classification layer. Location-sensitive information is thus largely lost, and the focus is on the presence/absence of the semantic concepts rather than where. We therefore propose to use the Gram matrix computed on the feature maps to complement the semantics-only appearance representation used in existing discriminators in order to induce location-sensitivity in object image generation. Such a Gram matrix based appearance representation has been used in style transfer [10] for style/texture representation, which seems to suggest that it only captures feature distribution but contains no spatial information. However, as pointed out in [24], this is because the use of entry-wise mean-square distance in [10] removes the location sensitivity in the feature map Gram matrix. In our model, we pass the raw matrix instead of mean-square distance to the discriminator classifier, therefore preserving the location sensitivity.

3. Preliminaries

3.1. Problem Definition

Let $L = \{(y_i, b_i)_{i=1}^N\}$ be a layout with N bounding boxes, where $y_i \in \mathcal{C}$ is the class of the bounding box and $b_i = [x_i, y_i, w_i, h_i]$ is the position and size of the bounding box in the image lattice ($H \times W$). The goal of the layout to image (L2I) generation task is to build a model $\mathcal{G}$, which can generate a realistic photo $I_g \in \mathbb{R}^{3\times H\times W}$, given the coarse information in the layout L.

3.2. Prior Models

Before introducing our proposed method in Sec. 4, we first briefly describe prior L2I models. In all previous models, the first step is always to generate a feature representation for each bounding box based on their classes:

$$\mathbf{p}_i = \phi_0([\mathbf{e}_i, \mathbf{n}_i]), \tag{1}$$

where $\mathbf{p_i} \in \mathbb{R}^{d_l+d_n}$ is the feature representation of the i^{th} bounding box in the layout, ϕ_0 is a linear transformation layer, $\mathbf{e_i} \in \mathbb{R}^{d_l}$ is the label embedding of y_i, and $\mathbf{n_i} \in \mathbb{R}^{d_n}$ is a random noise sampled from a zero-mean unit-variance multivariate Gaussian distribution. The generated feature vector set $\{\mathbf{p}_i\}_{i=1}^N$ is then fed into a generator $\mathcal{G}$ for image generation. Depending on how the generator uses the feature vector set to generate the image, the existing models can be grouped into the following two categories.

L2I Models with Encoder-Decoder Generators These models deploy an encoder-decoder generator [48, 27] which takes the feature vector set as input, and then transfers the feature vector set into a sequence of feature maps. Each feature map is generated by filling the corresponding feature vector into the region in the image lattice based on their bounding box. The generated feature maps are then fed into an encoder, which embeds each feature map into a latent space separately. Those embedded feature maps are merged into a single one through a convolutional LSTM network [37]. Finally, a decoder transforms the combined feature into the target image. Mathematically, the encoder-decoder based method can be formulated as:

$$I_g = \mathbf{D}(\text{cLSTM}(\mathbf{E}(\{\mathcal{F}(\mathbf{p}_i, b_i)\}_{i=1}^N))), \tag{2}$$

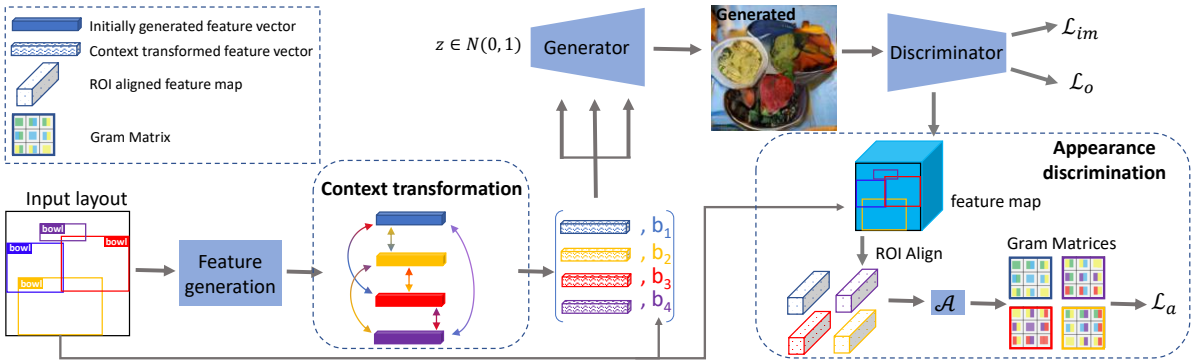

Figure 2. A schematic of our method with a decoder-only generator as in [38, 49]. The feature generation module generates the raw representation for each bounding box based on their class label, the context-aware feature transformation module integrates the global context into the representation of each bounding box. Then the transformed bounding boxes' representation and the box coordinates (b_i) are fed into the generator for image generation. Finally the generated image is compared with real images by a discriminator with three losses, namely image-level and object-level semantic loss ($\mathcal{L}_{im}$ and $\mathcal{L}_o$) and object-level Gram matrix loss ($\mathcal{L}_a$).

where $\mathcal{F}(\cdot,\cdot)$ is a filling operation, **E** is the encoder, cLSTM is the convolutional LSTM network, and **D** is the decoder.

L2I Models with Decoder-Only Generators These models [38, 39, 40] use a decoder-only generator to first generate an auxiliary mask[1] for each bounding box for a fine-grained shape or structure prediction:

$$\mathcal{M}_i = \mathcal{R}_{\mathcal{S}}(\psi(\mathbf{p}_i), b_i), \tag{3}$$

where ψ is a small convolutional neural network, $\psi(\mathbf{p}_i) \in \mathbb{R}^{H\times W}$, and $\mathcal{R}_{\mathcal{S}}(\cdot,\cdot)$ is a resize operator, which resizes each generated mask and fit it to the corresponding region in the image lattice via up/down sampling. Then the decoder receives a zero-mean unit-variance multivariate random noise $\mathbf{n}_0 \in \mathbb{R}^{C_0\times H_0\times W_0}$ as input, and decode it into the target image by modulating the affine transformation in the normalization layer:

$$\hat{f}_l = \text{BatchNorm}(f_l, \varphi_l(\sum_{i=1}^{N} \mathbf{p}_i \otimes \mathcal{M}_{l_i})), \tag{4}$$

where $\hat{f}_l$ and f_l are the feature maps before and after normalization at the l^{th} layer in the decoder, φ_l is a small convolutional block to generate the pixel-wise affine transformation parameters, $\mathcal{M}_{l_i}$ is the resized version of $\mathcal{M}_i$ to match the corresponding feature map's scale, and $\otimes$ is the outer product, by which a vector $\mathbf{p}_i$ and a matrix $\mathcal{M}_{l_i}$ produce a 3D tensor.

4. The Proposed Method

The main architecture of our proposed method is illustrated in Fig. 2. The proposed context-aware feature transformation module and location-sensitive Gram matrix based object appearance representation are integrated into the generator and discriminator respectively of a decoder-only L2I generation architecture [38, 39, 40]. Similarly they can be easily integrated with those employing an encoder-decoder architecture [48, 27].

[1]The mask is not a strictly binary mask, as it is the output of a layer with sigmoid activation.

4.1. Context-Aware Feature Generation

Let us first look at the feature transformation module. It is clear that the prior models process each bounding box independently (either in the feature generation stage or the mask generation stage in the decoder-only methods), disrespecting the other objects and stuff in the scene. As a result, the generated objects do not appear in harmony with other co-existing objects and stuff in the scene and often appear to be out of place (see Fig. 1 and Fig. 3). To overcome this limitation, we propose a context-aware transformation module, which integrates contextual information into the feature representation of each bounding box by allowing each feature to cross-examine all other features via self-attention [42]. Concretely, the contextualized representation of each bounding box is computed as:

$$\mathbf{p}_i^c = \sum_{j=1}^{N} w_{i,j}\mathbf{p}_j\mathbf{W}_v, \tag{5}$$

$$w_{i,j} = \frac{\exp(\alpha_{i,j})}{\sum_{k=1}^{N}\exp(\alpha_{i,k})}, \tag{6}$$

$$\alpha_{i,j} = (\mathbf{p}_i\mathbf{W}_q)(\mathbf{p}_j\mathbf{W}_k)^T, \tag{7}$$

where $\mathbf{W}_q, \mathbf{W}_k$ and $\mathbf{W}_v \in \mathbb{R}^{(d_l+d_n)\times(d_l+d_n)}$ are linear transformation layers. With the transformation, the contextualized representation of each bounding box not only has its own information, but also the global context in the layout. It is thus able to avoid the poor occlusion region generation problem shown in Fig. 1 (Top-Middle). Note that this module can be used for feature map filling in the encoder-decoder based methods, as well as the mask generation and the feature modulation steps in the decoder-only methods. The contextualized feature representation is then fed into the generator for image generation (see Fig. 2).

4.2. Location-Sensitive Appearance Representation

To address the issue of lacking location-sensitive appearance representation in the discriminators of existing L2I models, we introduce a feature map Gram matrix based appearance representation. In existing models' discriminators, the input image I_{im} is first processed by a convolutional neural network ψ_D, and represented as $f_{im} \in \mathbb{R}^{C\times H_D\times W_D}$:

$$f_{im} = \psi_D(I_{im}). \tag{8}$$

Existing L2I models then apply two losses in the discriminator to train the whole model: an image-level loss $\mathcal{L}_{im}$ according to the globally pooled feature of f_{im}, and an object-level conditional loss $\mathcal{L}_o$ based on the ROI pooled [32] feature of each object in the image, concatenated with its corresponding class information. These losses are designed to boost the realism of the generated image and the objects in the image respectively. However, using pooled feature as appearance representation means that both losses are location-insensitive, i.e., they only care about the presence/absence and strength of each learned semantic feature; much less about where the corresponding visual concept appear in the image.

To address this problem, we propose to introduce an additional appearance discriminator loss, which directly penalizes the spatial misalignment of each semantic feature between the generated and real images. Concretely, we use object feature maps' Gram matrix [10] as a new appearance representation and feed it to the discriminator classification layer. Formally, we define the appearance of a generated object in the image as:

$$\mathcal{A}_i = \mathbf{s}_i\mathbf{s}_i^T/d_s, \tag{9}$$

where $d_s = C$ is the channel dimension of the feature map, $\mathbf{s}_i \in \mathbb{R}^{C\times(H_D\times W_D)}$ is the spatial dimension vectorized feature representation of the i^{th} generated object in the image, computed as:

$$\mathbf{s}_i = \mathcal{R}_{\mathcal{A}}(f_{im}, b_i), \tag{10}$$

where $\mathcal{R}_{\mathcal{A}}(\cdot,\cdot)$ is the ROI align operator [12]. For simplicity, the vectorization operation is omitted here. The new appearance loss is then defined as:

$$\begin{aligned}\mathcal{L}_a(\mathcal{G},\mathcal{D}) = &\mathbb{E}_{\mathcal{A}^r\sim p^r_{data}(\mathcal{A}^r)}[log(\mathcal{D}(\mathcal{A}^r|y)] \\ &+ \mathbb{E}_{\mathcal{A}^g\sim p^g_{data}(\mathcal{A}^g)}[1 - log(\mathcal{D}(\mathcal{A}^g|y)],\end{aligned} \tag{11}$$

where $\mathcal{A}^r$ and $\mathcal{A}^g$ are the Gram matrices of object feature maps in real and generated images respectively, y is their corresponding class label. More specifically, for the i^{th} object in an image, its appearance loss is computed as:

$$\mathcal{D}(\mathcal{A}_i|y) = \frac{1}{C}\sum_{j=1}^{C}[\mathcal{A}_{i,j}, \mathcal{E}(y_i)]\mathcal{W}_A, \tag{12}$$

where $\mathcal{E}(y_i) \in \mathbb{R}^k$ is the label embedding, and $\mathcal{W}_A \in \mathbb{R}^{(C+K)\times 1}$ is a linear layer. The Gram matrix here captures the correlation between different feature channels and is clearly location-sensitive: each entry only assumes a large value when the corresponding two features are both present and activated at the same location. This loss is thus complementary to the two conventional losses ($\mathcal{L}_{im}$ and $\mathcal{L}_o$) which emphasize the presence of the semantics only.

4.3. Training Objectives

The final model is trained with the proposed appearance loss, together with image and object level losses [48, 38]:

$$\mathcal{G}^* = \arg\min_{\mathcal{D}}\max_{\mathcal{G}}\ \mathcal{L}_a(\mathcal{G},\mathcal{D}) + \lambda_{im}\mathcal{L}_{im}(\mathcal{G},\mathcal{D}) + \lambda_o\mathcal{L}_o(\mathcal{G},\mathcal{D}), \tag{13}$$

where λ_{im} and λ_o are the loss weight hyperparameters, and $\mathcal{L}_{im}$ and $\mathcal{L}_o$ are computed as:

$$\begin{aligned}\mathcal{L}_{im}(\mathcal{G},\mathcal{D}) = &\mathbb{E}_{I^r_{im}\sim p^r_{data}(I^r_{im})}[\log(\mathcal{D}(I^r_{im})] \\ &+ \mathbb{E}_{I^g_{im}\sim p^g_{data}(I^g_{im})}[1 - \log(\mathcal{D}(I^g_{im})], \\ \mathcal{L}_o(\mathcal{G},\mathcal{D}) = &\mathbb{E}_{O^r\sim p^r_{data}(O^r)}[\log(\mathcal{D}(O^r|y)] \\ &+ \mathbb{E}_{O^g\sim p^g_{data}(O^g)}[1 - \log(\mathcal{D}(O^g|y)],\end{aligned} \tag{14}$$

where I^r_{im} and I^g_{im} are real and generated images respectively, and O^r and O^g are objects in the real and generated images.

5. Experiments

Datasets Two widely used benchmarks, COCO-Thing-Stuff [25, 5] and Visual Genome [22] are used in our experiments. COCO-Thing-Stuff includes bounding box annotations of the 91 *stuff* classes in [5] and the 80 *thing/object* classes in [25]. Following [48, 38], only images with 3 to 8 bounding boxes are used in our experiments. Visual Genome is originally built for complex scene understanding. The annotations in Visual Genome contain bounding boxes, object attributes, relationships, region descriptions, and segmentation. As per standard in L2I generation, we only use the bounding boxes annotation in our experiments, and each layout contains 3 to 30 bounding boxes. We follow the splits in prior works [48, 38] on both datasets to train and test our model.

Implementation Details Our model is implemented with PyTorch. To show the general applicability of our proposed method, and for fair comparison with prior works, we adopt both encoder-decoder and decoder-only generators in the two instantiations of our method (termed Ours-ED and Ours-D respectively). The encoder-decoder generator has the same architecture as used in [48], and the decoder-only generator shares the same architecture as used in [38]. Following [48, 38], the resolution of generated images is 64×64 for the encoder-decoder generator and 128×128 for the decoder-only generator. The learning rate is set to

Table 1. Comparative results on COCO-Thing-Stuff and Visual Genome. E-D means encoder-decoder based generator, D means decoder-only based generator. † means improved decoder-only generator.

Methods	Resolution	Generator	Inception Score ↑		FID ↓		Diversity Score ↑	
			COCO	VG	COCO	VG	COCO	VG
Real images	64 × 64	-	16.3 ± 0.4	13.9 ± 0.5	-	-	-	-
Real images	128 × 128	-	22.3 ± 0.5	20.5 ± 1.5	-	-	-	-
pix2pix [17]	64 × 64	E-D	3.5 ± 0.1	2.7 ± 0.02	121.97	142.86	0	0
Layout2im [48]	64 × 64	E-D	9.1 ± 0.1	8.1 ± 0.1	38.14	40.07	0.15 ± 0.06	0.17 ± 0.09
Ours-ED	64 × 64	E-D	**10.27 ± 0.25**	**8.53 ± 0.13**	**31.32**	**33.91**	**0.39 ± 0.09**	**0.4±0.09**
Grid2Im [3]	128 × 128	E-D	11.22 ± 0.15	-	63.44	-	0.28 ± 0.11	-
LostGAN-v1 [38]	128 × 128	D	13.8 ± 0.4	11.1 ± 0.6	29.65	29.36	0.40 ± 0.09	0.43 ± 0.09
LostGAN-v2 [49]	128 × 128	D†	14.21 ± 0.4	10.71 ± 0.76	24.76	29.00	**0.55 ± 0.09**	0.53 ± 0.09
OC-GAN [40]	128 × 128	D	14.0 ± 0.2	11.9 ± 0.5	36.04	28.91	-	-
AG-Layout2im [27]	128 × 128	E-D	-	8.5 ± 0.1	-	39.12	-	0.15 ± 0.09
Ours-D	128 × 128	D	**15.62 ± 0.05**	**12.69 ± 0.45**	**22.32**	**21.78**	**0.55 ± 0.09**	**0.54 ± 0.09**

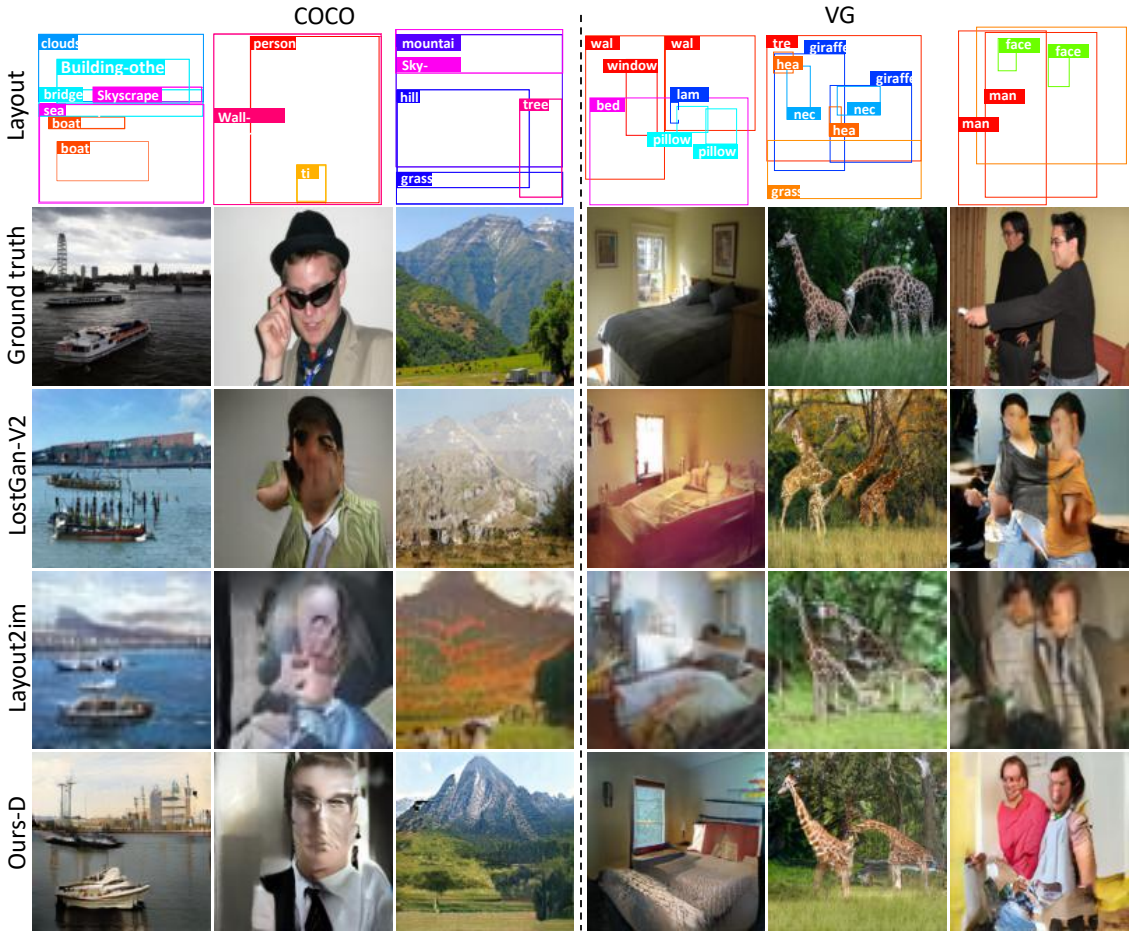

Figure 3. Qualitative results comparing Ours-D against two representative baselines Layout2im [49] and LostGAN-v2 [39].

$1e^{-4}$ for both generator and discriminator in all the experiments. We train our model for 200 epochs. The loss weight hyperparameters λ_{im} and λ_o are set to 0.1 and 1, respectively.

Evaluation Metrics We evaluate our model both automatically and manually. In automatic evaluation, we adopt three widely used metrics, namely Inception Score [36], Fréchet Inception Distance (FID) [14] and Diversity Score [47]. Inception Score evaluates the quality of the generated images. FID computes the statistical distance between the generated images and the real images. Diversity Score compares the difference between the generated image and the real image from the same layout. Following prior evalu-

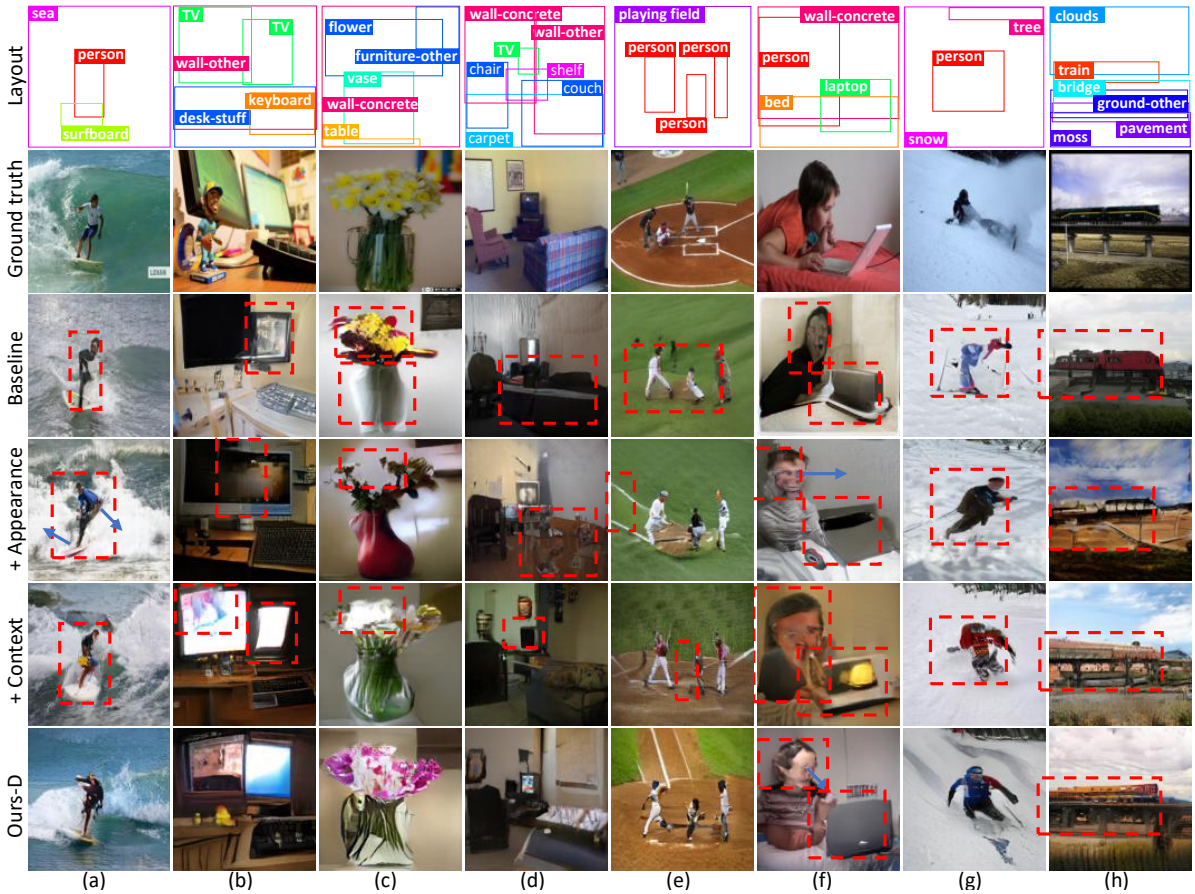

Figure 4. Qualitative ablative experimental results. Regions with clear generation quality differences are highlighted using red dashed boxes for close examination.

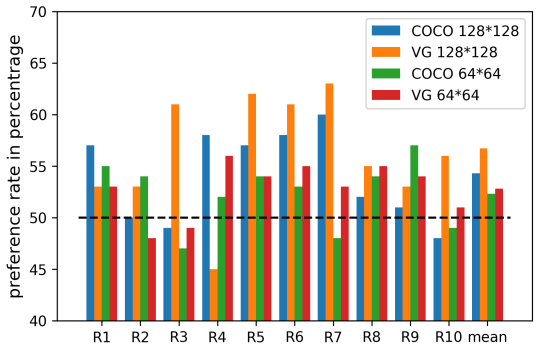

Figure 5. The preference rate of our model. A bar higher than dark dashed horizontal line indicates that our model is judged to be better than the compared baseline by the AMT workers.

ation protocol [3], for each layout, we generate five images in COCO-Thing-Stuff and one image in Visual Genome. In manual evaluation, we run perceptual studies on Amazon Mechanical Turk (AMT) to compare the quality of the generated images from different models. Ten participants engaged in the evaluation. Each participant was given 100 randomly sampled layouts from the testing dataset as well as the corresponding generated images from different models. All participants were asked to vote for their preferred image according to the image's quality and the matching degree to the paired layout. We compute the preference rate of each model from all participants. Due to the difference in generated image's resolution and for fair comparison, we compare our encoder-decoder generator based instantiation (Ours-ED) with the state-of-the-art encoder-decoder generator based baseline Layout2im [48] and decoder-only instantiation (Ours-D) with the state-of-the-art decoder-only generator based baseline LostGAN-v2 [39]. In both comparisons, the generated images are of the same resolution.

Main Results We compare our method with existing L2I models [48, 38, 39, 40, 27], the pix2pix model [17] which takes the input feature maps constructed from layout as implemented in [48], and the Grid2Im model [3] which receives scene graph as input. The following observations can be made on the quantitative results shown in Table 1. **(1)** Our method outperforms all compared methods on all benchmarks with both architectures and under all three automatic evaluations metrics, particularly for Inception Score and FID. **(2)** The more recent L2I methods take a decoder-only generator. Taking the same architecture but with the

Table 2. Ablation study on COCO-Thing-Stuff dataset.

baseline [38]	context	appearance	**Inception Score**	**FID**
✓			13.8 ± 0.4	29.65
✓	✓		14.97 ± 0.27	24.05
✓		✓	15.28 ± 0.24	**21.73**
✓	✓	✓	$\mathbf{15.62 \pm 0.05}$	22.32

two new components, our method (Ours-D) achieves new state-of-the-art. Fig 5 shows detailed statistics in the human evaluation on AMT. Among all 40 evaluation sets, our model won 32 sets. The preference rate is clearly higher at the higher resolution with more complex images (i.e., 128×128, VG dataset). Some qualitative results are shown in Fig. 3. It is evident from these examples that the images generated using our method are much more context-aware, i.e., different objects co-exist in harmony with each other and the background. Importantly, each generated object has sharper texture, clearer shape boundary with respect to background inside the object bounding box, and overall much more spatially-coherent than those generated by existing L2I models.

Ablation Study In this experiment, we adopt LostGAN-v1 [38] as our baseline and evaluate the effects of introducing our context transformation module and location-sensitive appearance representation. The quantitative results are shown in Table 2. We can see that both our context-aware feature transformation module and new appearance representation improve the baseline significantly on their own and when combined give a further boost. Some qualitative results are shown in Fig. 4. It is clear that the model trained with our appearance representation can generate objects with much better appearance both in terms of shape and texture (TV in Fig. 4(b) and person in Fig. 4(a)(f)(g)). Context transformation also plays an important role: the generated occluded regions become more natural (Fig. 4(b)(f)); each object's pose is also more in-tune with its surrounding objects and background, e.g. the surfing person's body pose is more physically plausible in Fig. 4(a); so is the person's head pose in the presence of the laptop in Fig. 4(f).

How Our Context Transformation Module Works In the decoder-only generator, a mask is generated using the representation of each bounding box to predict the fine-grained shape or structure of the object in each bounding box (Eq. 3). Without the context information in the feature representation, the generated masks would interfere with each other. This could result in irregular or incomplete object shape particularly in the occluded regions, which would further affect the feature modulation defined in Eq. 4. We investigate this effect by adding more bounding boxes into a layout, and visualizing the predicted masks as well as the generated images. The visualization results in Fig. 6 show clearly that the context-aware feature transformation

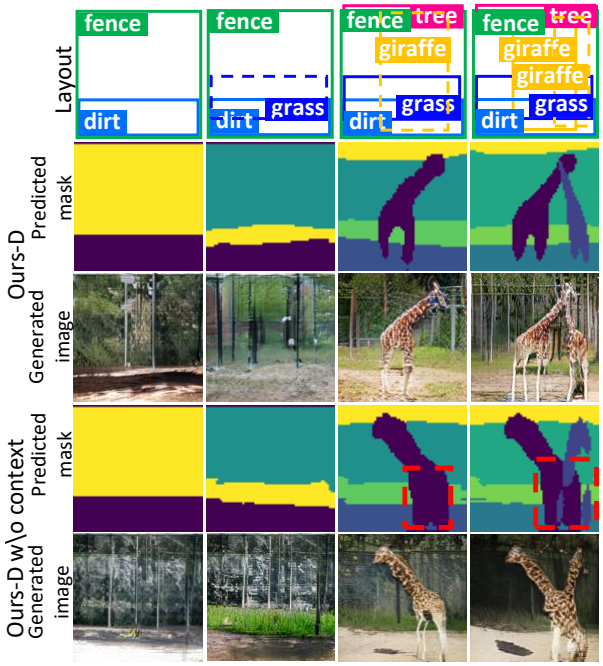

Figure 6. Qualitative examples about the contribution of context transformation in the complex scene generation. From left to right, at each time, we add one more bounding box into the previous layout, visualizing the predicted masks as well as the generated image by a model with our context transformation (Ours-D), and the same model without context transformation. Regions to pay more attention to are highlighted in dashed boxes.

module reduced the negative inter-object appearance interference in a complex scene when occlusion exists, yielding better appearance for the generated objects.

6. Conclusion

In this work, we proposed a novel context feature transformation module and a location-sensitive appearance representation to improve existing layout to image (L2I) generation models. In particular, they are designed to address existing models' limitations on lacking context-aware modeling in their generator and spatially sensitive appearance representation in their discriminator. Extensive experiments demonstrate the effectiveness of our method, yielding new state-of-the-art on two benchmarks.

Acknowledgment

This work was supported by the Center for Digital Innovations (ZDIN), Federal Ministry of Education and Research (BMBF), Germany under the project LeibnizKILabor(grant no.01DD20003) and the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122).

References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017. 2

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. In *ICML*, 2017. 2

[3] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *CVPR*, 2019. 1, 2, 6, 7

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 1, 2

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018. 2, 5

[6] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. In *ICLR*, 2017. 1

[7] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: deep generation of face images from sketches. *TOG*, 39(4):72–88, 2020. 2

[8] Zhe Chen, Shaoli Huang, and Dacheng Tao. Context refinement for object detection. In *ECCV*, 2018. 3

[9] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *CVPR*, 2020. 2

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. In *NeurIPS*, 2015. 3, 5

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1, 2, 3

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 5

[13] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *NeurIPS*, 2019. 3

[14] 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. In *NeurIPS*, 2017. 6

[15] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *CVPR*, 2018. 2, 3

[16] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018. 3

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 6, 7

[18] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 1

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 2

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2, 3

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 2, 5

[23] Yandong Li, Yu Cheng, Zhe Gan, Licheng Yu, Liqiang Wang, and Jingjing Liu. Bachgan: High-resolution image synthesis from salient object layout. In *CVPR*, 2020. 3

[24] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *IJCAI*, 2017. 3

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 5

[26] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *NeurIPS*, 2019. 3

[27] Ke Ma, Bo Zhao, and Leonid Sigal. Attribute-guided image generation from layout. In *BMVC*, 2020. 3, 4, 6, 7

[28] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017. 2

[29] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 1, 2, 3

[31] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 1, 2

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 5

[33] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020. 3

[34] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *NeurIPS*, 2017. 2

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 3

[36] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 1, 6

[37] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm

network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, 2015. 3

[38] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *ICCV*, 2019. 1, 3, 4, 5, 6, 7, 8

[39] Wei Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *arXiv preprint arXiv:2003.11571*, 2020. 1, 2, 3, 4, 6, 7

[40] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. *arXiv preprint arXiv:2003.07449*, 2020. 1, 2, 4, 6, 7

[41] Antonio Torralba, Kevin P Murphy, William T Freeman, Mark A Rubin, et al. Context-based vision system for place and object recognition. In *ICCV*, 2003. 3

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[43] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan L Yuille. Robust object detection under occlusion with context-aware compositionalnets. In *CVPR*, 2020. 3

[44] Hang Xu, Chenhan Jiang, Xiaodan Liang, and Zhenguo Li. Spatial-aware graph relation network for large-scale object detection. In *CVPR*, 2019. 3

[45] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018. 3

[46] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. 1, 3

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[48] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7

[49] Bo Zhao, Weidong Yin, Lili Meng, and Leonid Sigal. Layout2image: Image generation from layout. *IJCV*, 128(10):2418–2435, 2020. 2, 4, 6

[50] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2014. 3

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2